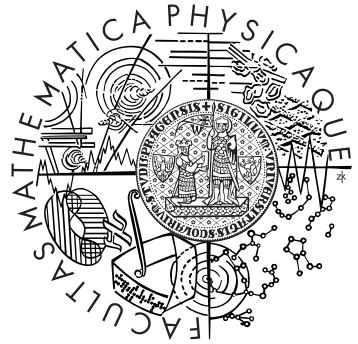
English-to-Czech MT: Large Data and Beyond

Ondřej Bojar

Habilitation Thesis



Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Prague, 2017

Contents

Chapter 1

Introduction

This habilitation thesis consists of 12 publications authored or co-authored by Ondřej Bojar. The publications were selected and organized to highlight the author's contribution to the state of the art in machine translation (MT), particularly translation into morphologically rich languages like Czech.

The thesis is structured as follows. Chapter 2 serves as a very brief overview of the task of machine translation, highlighting the core problems that have to be tackled and setting the context for the author's contributions detailed in the rest of this text.

Chapter 3 starts with a quick summary of the author's efforts devoted to the collection and preparation of training data. What may seem a somewhat boring product is nevertheless a valuable resource for many researchers and a critical component necessary to achieve the state of the art in translation quality, as discussed in the following chapters.

Chapter 4 covers the first of the three main contributions of the author: **improving grammaticality, and particularly morphological coherence**, in phrase-based machine translation. While large data are essential for attaining good performance in machine translation, it is not conceivable to collect corpora large enough to cover all possible word forms and provide sufficiently dense statistics about their usage in all possible contexts. Targeting languages with highly productive morphological systems such as Czech thus requires some form of explicit handling of morphology and this chapter summarizes the author's research in this area.

Chapter 5 is focused on the second main contribution, namely **employing deeper linguistic information** to improve translation quality. While statistical methods have had a great success in machine translation, the nature of the handled subject, natural text, belongs to the field of linguistics, and it is therefore interesting to examine to what extent can statistical approaches to MT benefit from linguistic knowledge. The chapter explains the problems faced when trying to organize the statistical models along the linguistic structure of the sentence and describes the author's proposed method that circumvents these problems. The resulting system Chimera outperformed all other MT systems participating in the English-to-Czech news translation task in the years 2013–2015, including Google Translate and other commercial and on-line systems. The setup of Chimera is naturally not limited to translating news text, and adapted

versions of the system served in applied EU projects (QTLeap, HimL) as well as in commercial collaboration of the author's department with IBM.

Finally, evaluation is critical in all applied sciences and evaluating machine translation is particularly intriguing. Chapter 6 is devoted to the third main area of the author's contributions, namely to methods of **manual and automatic MT evaluation**, explaining why MT evaluation is a difficult discipline, revealing the reasons of low performance of an established automatic evaluation measure and proposing modifications to improve the correlation with human judgement.

The last Chapter 7 summarizes the author's service to the community through his contribution to the organization of shared tasks related to machine translation.

The thesis is concluded in Chapter 8. Key papers (co-)authored by Ondřej Bojar and cited throughout the text are reprinted in Appendix A.

Chapter 2

Problems and Solutions in Machine Translation

The goal of machine translation is to translate text from one natural language to another. Machine translation is sometimes dubbed as the "king discipline" of computational linguistics, because translation easily entails almost all aspect of natural language and its meaning: from meaning ambiguity and the relation between the form of an expression and its function in the communication to complex rules of grammatical correctness.

Despite the complexity of language phenomena involved, machine translation has been very successfully tackled by **statistical methods** even in their relatively simple form.

In statistical machine translation, an approach prevalent since 1990s (Brown *et al.*, 1990, 1993; Berger *et al.*, 1994), we search for the most likely target sentence $\hat{e}_1^{\hat{I}}$ (a sequence of target words $\hat{e}_1, \ldots, \hat{e}_{\hat{I}}$) given the source sentence f_1^J:

$$\hat{e}_1^{\hat{I}} = \underset{I, e_1^I}{\operatorname{argmax}}\, p(e_1^I | f_1^J) \tag{2.1}$$

The parameters of the probabilistic distribution $p(e_1^I | f_1^J)$ are estimated automatically from parallel corpora (texts translated previously by humans), subject to various simplifying assumptions.

One of these assumptions, still mainly followed today and reflected also in Eq. 2.1, is that sentences are translated individually, ignoring any contextual information beyond sentence boundaries.

Another critical assumption is that the sentence can be decomposed into a small finite number of translation units which are then translated more or less independently of each other. This assumption has been removed only very recently through the adoption of deep learning methods (neural machine translation, see Section 2.2.3 below). Since the nature of neural MT is also statistical, we will use the qualifier "classical" statistical methods to denote approaches that rely on the decomposition into separate translation units. We will however follow the common usage of abbreviations and use SMT to denote *classical* statistical MT only.

In SMT, additional model components are used to compensate for the independence assumption of translation units and ensure overall coherence of the sentence.

The first step in the classical SMT derivation is to use the Bayes' law and decompose the probability into two components, the **translation model** $p(f_J^I|e_1^I)$ and the **language model** $p(e_1^I)$:

$$p(e_1^I|f_1^J) = \frac{p(f_J^I|e_1^I)p(e_1^I)}{p(f_1^J)} \tag{2.2}$$

Bayes' law reverses the conditional probability in the translation model, but this does not pose any problem: translational equivalence is usually understood as bidirectional and the reversed probability is going to be estimated from the same type of data, parallel texts, anyway.

Furthermore, the denominator is constant in the maximization, so under argmax, we can write:

$$\hat{e}_1^{\hat{I}} = \underset{I,e_1^I}{\operatorname{argmax}}\, p(e_1^I|f_1^J) = \underset{I,e_1^I}{\operatorname{argmax}}\, p(f_J^I|e_1^I)p(e_1^I) \tag{2.3}$$

Eq. 2.3 is called the **noisy channel model** (Brown *et al.*, 1990). Since Och and Ney (2002), the common formal device used in SMT is the more flexible **log-linear model**: The conditional probability of e_1^I being the translation of f_1^J is modelled as a combination of independent feature functions $h_1(\cdot,\cdot),\ldots,h_M(\cdot,\cdot)$ describing the relation of the source and target sentences:

$$p(e_1^I|f_1^J) = \frac{\exp(\sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J))}{\sum_{e'^{I'}_1} \exp(\sum_{m=1}^M \lambda_m h_m(e'^{I'}_1, f_1^J))} \tag{2.4}$$

Similarly to the noisy channel model (which is in fact a special case of the log-linear model), the denominator in Eq. 2.4 depends on the source sentence f_1^J only and does not affect the selection of the maximum, and neither does the exponential, giving us a simplified formula:

$$\hat{e}_1^{\hat{I}} = \underset{I,e_1^I}{\operatorname{argmax}}\, p(e_1^I|f_1^J) = \underset{I,e_1^I}{\operatorname{argmax}} \sum_{m=1}^M \lambda_m h_m(e_1^I, f_1^J) \tag{2.5}$$

The assumption of translation units is formally reflected by defining a joint segmentation s_1^K of the source sentence and the target candidate into K translation units. The majority of features $h_m(\cdot,\cdot)$ are required to decompose along the segmentation, i.e., to take the form:

$$h_m(e_1^I, f_1^J, s_1^K) = \sum_{k=1}^K \tilde{h}_m(\tilde{e}_k, \tilde{f}_k) \tag{2.6}$$

where $\tilde{f}_k$ represents the source side of the translation unit and $\tilde{e}_k$ represents its target side given the segmentation s_1^K.

Feature functions that decompose along this joint segmentation are called **local** and other feature functions are called **non-local**. To distinguish them, we can divide the sum over model components into two parts: M_L local and M_N non-local features:

$$\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) = \sum_{m_L=1}^{M_L} \lambda_{m_L} \sum_{k=1}^{K} \tilde{h}_{m_L}(\tilde{e}_k, \tilde{f}_k) + \sum_{m_N=1}^{M_N} \lambda_{m_N} h_{m_N}(e_1^I, f_1^J) \tag{2.7}$$

Ideally, the segmentation s_1^K should be treated as a hidden parameter and summed over in the maximization in Eq. 2.1. This would be too complicated and too expensive, so it practice, we search for the best derivation, i.e., the pair of segmentation $\hat{s}_1^{\hat{K}}$ and translation $\hat{e}_1^{\hat{I}}$:

$$\begin{aligned}
\hat{e}_1^{\hat{I}}, \hat{s}_1^{\hat{K}} &= \operatorname*{argmax}_{I, e_1^I, K, s_1^K} p(e_1^I | f_1^J) \\
&= \operatorname*{argmax}_{I, e_1^I, K, s_1^K} \sum_{m_L=1}^{M_L} \lambda_{m_L} \sum_{k=1}^{K} \tilde{h}_{m_L}(\tilde{e}_k, \tilde{f}_k) + \sum_{m_N=1}^{M_N} \lambda_{m_N} h_{m_N}(e_1^I, f_1^J) \\
&= \operatorname*{argmax}_{I, e_1^I, K, s_1^K} \sum_{k=1}^{K} \sum_{m_L=1}^{M_L} \lambda_{m_L} \tilde{h}_{m_L}(\tilde{e}_k, \tilde{f}_k) + \sum_{m_N=1}^{M_N} \lambda_{m_N} h_{m_N}(e_1^I, f_1^J)
\end{aligned} \tag{2.8}$$

The component weights λ_m are most commonly optimized with respect to the final translation quality measure. Traditionally, this process is called "tuning" or "model optimization".

2.1 Problems of Machine Translation

Machine translation is a challenging task for several reasons. Adopting the classical statistical MT strategy, we have to choose adequate translation units first and be able to effectively gather them from training data. Then, SMT has to consider a very large search space of possible outputs. And finally, identifying which possible outputs are good and which are bad is difficult.

Defining Translation Units As mentioned above, individual sentences of natural languages are rather complex and up until very recently, they were always decomposed into some smaller units, translating each of these units more or less independently. The various definitions of the units gave rise to word-based (Brown *et al.*, 1990, 1993), phrase-based (PBMT, Koehn *et al.*, 2003) or various

arts of syntax-based (Yamada and Knight, 2001; Zollmann and Venugopal, 2006; Chiang, 2010; Bojar and Hajič, 2008) statistical machine translation.

The choice of a translation unit affects the difficulty in obtaining the "translation dictionary" of these units and the difficulty in decomposing sentences into these units and putting them back together to form the translated sentence.

Shallow units like individual word forms or short sequences of word forms ("phrases" in phrase-based MT, see Section 4.1) are easier to obtain but we very often risk producing a grammatically incorrect output when combining them. Linguistically more adequate units, e.g., some deep-syntactic nodes or treelets, rely on tools for sentence analysis and generation and suffer from their errors.

Larger units (e.g., longer phrases in phrase-based MT) can cover the necessary linguistic dependencies within a single unit, thereby preventing errors at unit combination, but they are obviously much harder to observe in sufficient numbers.

More coarse-grained units such as base forms (lemmas) of words are less prone to data sparsity issues but they imply some information loss which can easily cause a harm to the meaning of the sentence and they are again harder to use correctly.

Managing Huge Search Space As shown already by Knight (1999), picking the right word order and covering source multi-word translation units with entries from translation dictionary are two sub-tasks that render machine translation NP-complete.

When we work with two languages, we can treat target language words as the repertoire of possible "meanings" of source words. It is easy to notice the ambiguity of expressions and its multiplicative effect whenever more occur in a sentence in striking examples like *The plant is next to the bank.* (The *plant* can be a flower or a factory, the *bank* can be a financial institution or a river bank.)

In practice, the number of options to choose from is actually much higher for two main reasons: (1) the input can be often segmented into translation units in many possible ways, and (2) automatically extracted "translation dictionaries" offer many more possible translations (as observed in the translated data) than one would expect. Bojar (2015)[1] reviews how various problems of MT get worse due to morphological richness of languages, including this type of ambiguity: i.e., the translation system has to choose not only the right word but also its morphological form to indicate its relationship to other words in the sentence (e.g., agreement) or to refine its meaning (e.g., plural).

Assessing Translation Quality Given the large space of possible translations, we would need a reliable method for distinguishing good and bad translations. This enterprise is called "machine translation evaluation" (if a reference

[1] (Bojar, 2015) is reprinted as Appendix A.1 on page 66.

translation is available) or "quality estimation" (if we do not have the reference translation) and it is as old and as complex as MT itself.

Not very surprisingly, small changes in the sentence can drastically change its meaning (e.g., reversing the negation). At the same time, a very different wording can convey the same meaning as the original but we are usually given just one reference translation.

2.2 Complementary Solutions

The history of SMT, see Bojar (2012) or Koehn (2009) for a summary, has seen many complementary methods addressing various aspects of the core problems outlined above. Here we highlight those related our contributions as detailed in the subsequent chapters.

2.2.1 Using Large Data

The success of statistical MT relies on the access to large training data. In fact, some of the problems of MT outlined above lose in their severity as the training data grow. With very large data, we can afford using larger translation units (e.g., longer and longer phrases in phrase-based translation) when covering the input and the phrase-independence assumption will have fewer occasions to do any harm. In the ideal case (which indeed does happen in small and repetitive domains), the whole sentences will be available for reuse.

Precisely for that reason, we have put considerable efforts into collecting large Czech-English parallel data, see Chapter 3.

2.2.2 Adding Linguistic Information

Common approaches to SMT often lack sufficient generalization power and violate many linguistic constraints. For instance, pure phrase-based MT can only produce forms of words as seen in the training data and it has no means to capture the overall sentence structure.

It is therefore interesting to add linguistic knowledge explicitly to the model. In our work, we followed the layered formal description of sentences in natural language defined by the Functional Generative Description (Sgall *et al.*, 1986). We tried to benefit from both relatively shallow morphological layer (information relevant for each token in the linear sequence of words in the sentence) as well as from the syntactic analysis of the sentence.

We were successful in utilizing the token-level information, see Chapter 4 for more details. Our attempts to employ the subsequent layers of linguistic description (shallow and deep syntax) were less successful, mainly because they implicitly *strengthened* the unjustified independence assumption of individual translation units. The deep-syntactic approach to MT was so far best exploited

in the transfer-based system TectoMT (Popel and Žabokrtský, 2010) and despite the system did not perform very well on its own, we managed to incorporate TectoMT to the standard phrase-based system in a way that set the new state of the art in English-to-Czech translation, see Chapter 5.

2.2.3 Removing Independence Assumptions

Our work on the core of machine translation has been carried out in the confines of classical statistical MT that deals with individual translation units. We contributed to attempts at removing this assumption through the supervision of Aleš Tamchyna's PhD studies (2012–2017), where Aleš developed a discriminative model to select translation of phrases considering the whole source-side and a small target-side context (Tamchyna, 2017; Tamchyna *et al.*, 2016a; Huck *et al.*, 2017); more details are provided in Section 4.3.3.

A breakthrough in machine translation quality was achieved recently through deep learning, giving rise to neural machine translation, NMT (Sutskever *et al.*, 2014; Bahdanau *et al.*, 2014).

NMT replaces the log-linear model with a model directly predicting target words, one at a time, conditioned on the whole source sentence f_1^J:

$$\begin{aligned} p(e_1^I|f_1^J) &= p(e_1, e_2, \dots e_I|f_1^J) \\ &= p(e_1|f_1^J) \cdot p(e_2|e_1, f_1^J) \cdot p(e_3|e_2, e_1, f_1^J) \dots \\ &= \prod_{i=1}^{I} p(e_i|e_{i-1}, \dots e_1, f_1^J) \end{aligned} \tag{2.9}$$

The similarity of NMT to a standard language model should be highlighted. A language model (see Section 4.1 below) predicts the next word based on the previous words: $p(e_1^I) = \prod_{i=1}^{I} p(e_i|e_1, \dots e_{i-1})$. NMT adds f_1^J to the antecedent.

While the main steering force in PBMT is the translation model and the language model "only" caters for target coherence, the main steering force in NMT is the language model and the source "only" conditions the word choices. The exact consequences of this major shift are yet to be explored but NMT generally performs much better in fluency of translation and somewhat better in adequacy.

2.2.4 Better Evaluation

The outputs of machine translation are evaluated manually and automatically for a number of reasons. From the end users' point of view, we need to be able to select the overall best performing MT system. System developers need to be able to reliably check progress or, with the help of automatic evaluation methods, automatically optimize model parameters.

If our metrics[2] of MT quality do not reflect well the problems in output, we cannot expect any improvements. At the same time, *understanding* what is a good and what is a bad translation is an essential component of machine translation as a field of study.

Our contributions to both the technical and scientific aspects of MT evaluation are summarized in Chapter 6.

[2]The term "metric" traditionally used in the field of MT evaluation does not imply the properties of a metric in the mathematical sense.

Chapter 3

Large Data

The collection and preparation of training data may seem a rather mundane task from the scientific point of view. It is nevertheless undisputably the key prerequisite for statistical methods in NLP in general and MT in particular. We also take the stance that a high-quality training dataset attracts attention to the task and languages concerned. We believe that our long-term work on a large Czech-English parallel corpus CzEng described in this chapter has thus not only allowed our own research in English-to-Czech MT but also considerably contributed to the overall focus on this language pair and its adoption as an interesting research problem. MT into Czech is thus examined to a much deeper extent than what would correspond for example to the number of speakers of Czech or the amount of money spent on NLP research by national funding agencies.

Our main contribution in data collection and preparation is the series of releases of CzEng, summarized in Table 3.1. Every release, aside from including additional training data was devoted to a particular topic.

Three CzEng releases deserve a special remark. The version 0.9 (Bojar and Žabokrtský, 2009) was the first major upgrade when we processed both of the sides of the corpus with the Treex NLP processing platform (Popel and Žabokrtský, 2010; in 2009, the platform was still called TectoMT). CzEng 0.9 with its 8.0 million sentences posed a significant technical challenge to the toolkit. Up until then, Treex has been used in various NLP tasks, but processing time and stability across a wide range of data conditions were never the main focus of its development. CzEng 0.9 served as a very thorough test case and allowed to identify many corner cases and minor bugs in the toolkit. Since there was no time available for any major code rewrites, the goal was achieved through data parallelization and automatic collection of failures. We then processed the bugs from the most frequent to the less common ones.

The second major step in CzEng development was achieved in the version 1.0 (Bojar *et al.*, 2012b).[1] In that release, we not only almost doubled the corpus size again, provided the automatic processing (improved in various aspects) but we also carefully filtered the corpus to avoid low-quality sentence pairs. In CzEng 1.0 for the first time, we exploited the other side of the corpus to enhance the automatic annotation even monolingually. Specifically, the comparison of the

[1] (Bojar *et al.*, 2012b) is reprinted as Appendix A.2 on page 91.

Ver.	Size	Main Focus	Details in
0.5	0.9M	Sentence alignment, common format	Bojar and Žabokrtský (2006)
0.7	1.0M	Used in WMT06 and WMT07	Bojar *et al.* (2008)
0.9	8.0M	Automatic annotation up to t-layer	Bojar and Žabokrtský (2009)
–	–	Sentence-level filtering	Bojar *et al.* (2010b)
1.0	15.0M	Improving monolingual annotation through parallel data	Bojar *et al.* (2012b)
1.6	62.5M	Processing tools dockered	Bojar *et al.* (2016b)

Table 3.1: Summary of CzEng release versions. Size is reported in millions of sentence pairs.

Czech and English automatic annotation allowed us to (1) improve sentence segmentation by adding dedicated training data and new focus patterns to our trainiable tokenizer (Maršík and Bojar, 2012) and (2) spot and fix several errors in the rules construcing "formemes" (Žabokrtský *et al.*, 2008) due to unexpected formeme mismatches in the aligned sentences.

Finally, the most recent release, CzEng 1.6 (Bojar *et al.*, 2016b) benefited from our supervision of Jakub Kúdela's master thesis and publication (Kúdela *et al.*, 2017): 1.84 billion of web pages of the July 2015 Common Crawl were scanned for parallel Czech-English texts through sentence embeddings and locality-sensitive hashing. The goal was to again extend the CzEng parallel data, but as we described in Kúdela *et al.* (2017), Common Crawl was too "sparse". From each website, Common Crawl usually gets only a handful of pages. We thus could not rely directly on Common Crawl data dump and re-crawled the list of websites with parallel content for CzEng 1.6.

Besides MT, CzEng has been used in research on coreference resolution (Novák *et al.*, 2013), automatic valency frame selection (Dušek *et al.*, 2014), in the development of a valency lexicon (Fučíková *et al.*, 2016), a subjectivity lexicon (Veselovská, 2015), a lexical network (Ševčíková *et al.*, 2016), word-level (Kocmi and Bojar, 2016) and sentence-level (Wieting *et al.*, 2017) embeddings or a spoken corpus of Czech dialects (Michlíková, 2013) and in semi-automatic linking between corpora and lexicons (Bejček, 2015).

Chapter 4

Handling Morphology in Phrase-Based MT

The common topic that threads through this thesis is the difficulty of targetting Czech with its rich morphology. Morphological correctness was undoubtedly the most apparent issue of the PBMT-based systems.

Table 4.1 motivates this research by illustrating the availability of morphological variants of the Czech word *čéška (knee cap)* in plural in training corpora of 50K to 50M sentences. The word is not very frequent, but we are lucky to see it in the nominative case (line 1) already in 50K training sentences. Other morphological variants are seen as we use larger corpora. In 50M sentences, we finally see all morphological variants of the word, although the vocative case (line 5) was actually still not seen and we know the form only thanks to its homonymy with the nominative.

case	surface form	50K	500K	5M	50M
1	čéšky	•	•	•	•
2	čéšek	–	•	•	•
3	čéškám	–	–	•	•
4	čéšky	○	○	•	•
5	čéšky	○	○	○	○
6	čéškách	–	•	•	•
7	čéškami	–	–	–	•

Table 4.1: The seven Czech cases of the word *čéška* (knee cap) in plural as seen in 50K/500K/5M/50M sentences. "•" indicates the word was seen in the particular case, "○" indicates that the surface form was seen but in a different case. Reproduced from Huck *et al.* (2017).

In order to correctly use words in a morphologically rich language, the SMT system has to have the capacity to produce them given the English source in the first place (i.e., to see them in a parallel corpus) and also to select the form that fits the given context. As indicated by the example in Table 4.1, *some* morphological variant of a word may be seen in a relatively small number of sentence pairs, but we can't expect to see all forms.

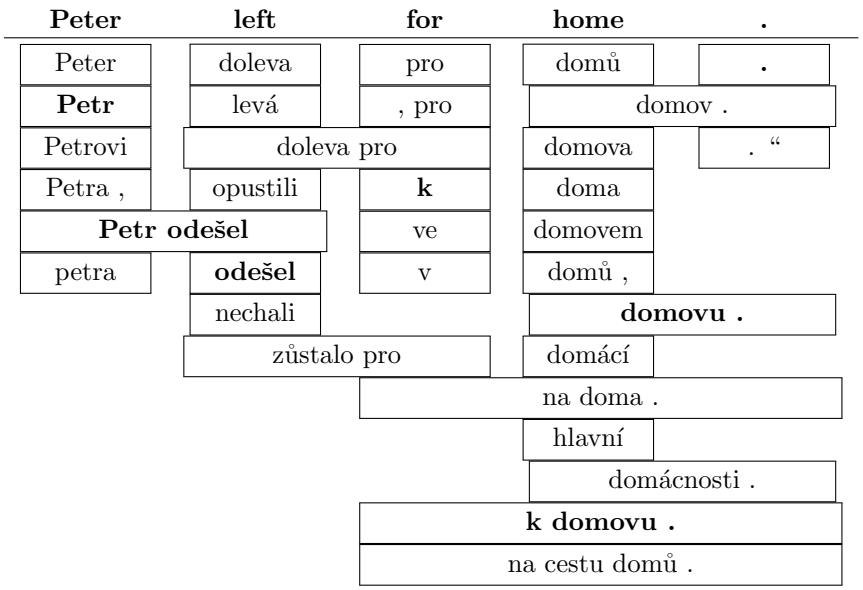


Figure 4.1: Translation options considered by PBMT when translating the sentence “*Peter left for home.*” from English into Czech. Options with a higher translation probability are listed higher, bold indicates options that could be used to construct an acceptable, although not very good translation. Figure simplified from Bojar (2012).

In this chapter, we describe our contributions to producing correct text in morphologically rich languages. We start with a very brief summary of the underlying framework of phrase-based MT (Section 4.1), then focus on improved modelling in situations when the needed target word forms are generally available in the training data (Section 4.2) and conclude by our contributions to producing word forms which were not observed in the parallel or even in the monolingual data (Section 4.3).

4.1 Overview of Phrase-Based MT

Phrase-Based MT (PBMT, Koehn *et al.*, 2003) is one of several classical statistical approaches to MT. Thanks to the availability of open-source implementation of a strong PBMT system Moses (Koehn *et al.*, 2007), phrase-based MT has become the industry standard and remained so until about 2016.

PBMT assumes that the input sentence can be decomposed into *contiguous* sequences of words called “phrases” and each of the phrases can be translated more or less independently. Figure 4.1 illustrates such a decomposition and possible translation units (called **translation options** in PBMT) for the English sentence *Peter left for home.*

The output sentence is constructed left-to-right, selecting phrase translations

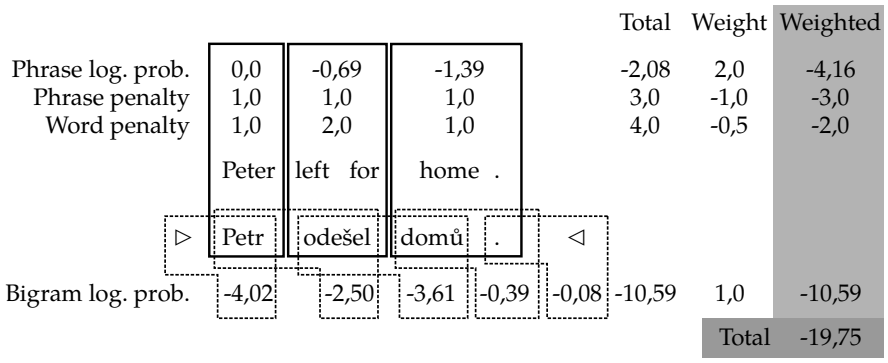


Figure 4.2: Local and non-local features scoring one candidate translation. The solid rectangles indicate individual translation options. Any information available in each of the rectangle can be used for local features. Non-local features cross translation options boundaries and as an example, we illustrate the use of a bigram LM (dotted). The scores are added up for each feature and finally weighted by the weights of the log-linear model. Reproduced from Bojar (2012).

from the source sentence in any order (subject to reordering costs).

Formally, PBMT is implemented as a log-linear model described in Chapter 2, where the key local features are:

- phrase translation probabilities (several variants are used simultaneously, see Koehn, 2009),
- phrase count; its weight is called phrase penalty and moderates if translations are rather literal (word for word) or not,
- word count; its weight is called word penalty and controls the output length.

The only non-local feature is the language model, so any coherence of the selected candidate (e.g., short- or long-range agreements) is to be ensured by the language model. Unfortunately, the language models that were most often used were n-gram LMs. This helped tractability (it was sufficient to keep the previous $n-1$ output tokens in the search state to allow LM evaluation)[1] but it has a serious detrimental effect on overall sentence grammaticality.

Figure 4.2 illustrates one candidate translation, as constructed from the English source using three translation options.

The parameters of these model components are generally estimated using maximum likelihood estimates, usually subject to some form of smoothing or

[1]LMs of unlimited history became possible thanks to deep learning (Bengio *et al.*, 2003) and they indeed brought an improvement to PBMT (Schwenk, 2007) but they never became widely used because the computational costs were too high before the computation was moved to GPUs (Schwenk *et al.*, 2012).

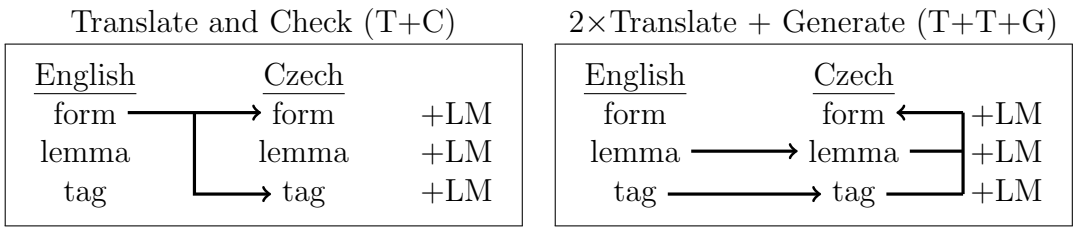


Figure 4.3: Two basic factored translation setups.

interpolation (Chen and Goodman, 1996; Foster *et al.*, 2006) from parallel data (phrase translations) and target-side monolingual data (language models).

As mentioned in Section 2.1, the number of possible translations of a given sentence is exponential to the sentence length, so the space is explored in an approximate search, e.g., **beam search**. Many candidate *partial* translations are considered simultaneously, the more promising ones are further expanded by attaching translation options covering so far untranslated words while the less promising ones are discarded.

In its pure form, PBMT treats word forms as opaque symbols. This is a great advantage for language independence of the method but it comes at the cost of severe data sparsity for morphologically rich languages: the model needs to see all possible forms of all possible translations of a word to have the capacity to produce them. And it should also see each of them in a large number of contexts to be able to select the correct one.

4.2 Factored Setups for Improving Morphological Choices

The implementation of PBMT in the Moses translation system introduced **factors** (Koehn and Hoang, 2007). In short, factors provide additional information for each input and/or output token, and thereby allow to introduce new score components and also to generate output factors based on additional data, not just the parallel corpus.

In Bojar (2007),[2] we thoroughly examined the utility of factored PBMT for targetting Czech.

If we limit ourselves to factors bearing morphological information,[3] two setups immediately come to mind, as illustrated in Figure 4.3 and explored in Bojar (2007):

- T+C (Translate and Check) translates the source word forms into target

[2](Bojar, 2007) is reprinted as Appendix A.3 on page 99.

[3]Other options are obviously possible and helpful, see e.g., Avramidis and Koehn (2008), Birch *et al.* (2007), or Niehues and Waibel (2010).

word forms, as baseline PBMT would do, but it also produces target-side morphological tags. This sequence of tags can be then scored with a dedicated language model which operates on a much smaller vocabulary (morphological tags) and therefore can be effectively trained for a much higher n-gram size (e.g., 7 or 10-grams).

- T+T+G (2×Translate and Generate) translates lemmas and morphological tags *independently* and generates the target word form from the lemma and morphological tag; again, multiple language models are used. This setup is linguistically appealing, it correctly strips morphological variance of words from their lexical values. Figure 4.4 explains the benefit from independent learning of translation of lemmas and translation of morphological tags: evidence can be assembled from different sentences, the co-occurrence counts are generally higher and probability estimates more reliable.

In later studies, we wanted to build upon these setups. The T+C setup works very well, as we demonstrated in Bojar (2007) but it is difficult to improve it further, see Section 4.2.1. The T+T+G setup brings serious complications, as described in Section 4.2.2. We proposed several techniques to circumvent the issues, see Section 4.3.

4.2.1 Automatic Exploration of Configurations Infeasible

The content of factors as well as the exact sequence in which they are used on the source side and constructed on the target side is fully configurable. The space of possible configurations is thus very large, especially if we consider also the various meta-parameters such as n-gram size or type of smoothing of each of the language models, and their effectiveness also depends on the amounts of available training data.

In a series of experiments, we largely explored this space of possible configurations:

- In Bojar and Tamchyna (2013),[4] we developed Eman, an experiment manager. Eman, populated with "seed" scripts relevant for machine translation (or any other field of study), allows to manually explore large numbers of configurations, automatically reusing common model parts and rebuilding only what is necessary.

  Eman has been used in the development of almost all our MT experiments and when building shared task systems as well as commercially applied MT systems. While Eman was designed for research and flexibility in experimenting, it also serves as the backbone of a fully automated batch translation online service that we run for IBM to translate into Czech, Hungarian, Arabic and experimentally also into Japanese.

[4] (Bojar and Tamchyna, 2013) is reprinted as Appendix A.4 on page 107.

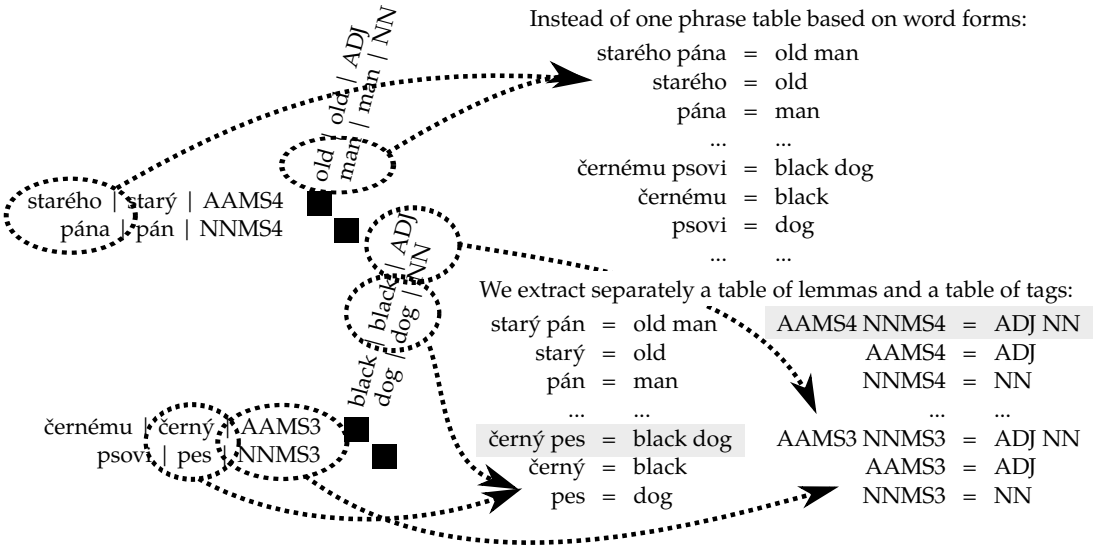


Figure 4.4: Linguistically motivated extraction of factored phrases from a parallel corpus. The corpus, consisting of just two "sentence" pairs: *(viděl jsem) starého pána = (I saw an) old man* and *(dej to tomu) černému psovi = (give it to the) black dog*, does not allow to directly learn the phrase *black dog = černého psa* (the translation of *black dog* into Czech accusative case). In the factored setup T+T+G, this translation is licensed by the combination of the separate lemma (*černý pes*) and tag (*AAMS4 NNMS4*) translations, each of which comes from a different training sentence pair. Reproduced from Bojar (2012).

- In Bojar *et al.* (2012a), we introduced a simple taxonomy for the more common factored setups and further examined which setups work best in various data conditions.

- In Tamchyna and Bojar (2013), Eman served as the underlying engine in an attempt to explore the space of possible PBMT configurations *fully automatically*. While we were able to find a small number of setups that improved the baseline, the main result of that work is negative:

  - The space of possible factored configurations is too large to be explored automatically, i.a. there are exponentially many setups given a number of source-side factors.
  - Evaluating each configuration is computationally demanding (e.g., a few days of computing time with large training data).
  - The automatic evaluation metric (BLEU in that case, see Chapter 6 for more details) is not sufficiently discerning and reliable, many setups receive too similar scores.

– Model optimization is non-deterministic and fragile; several optimization runs of the same setup often differ in their performance more than possible alternative setups.

Across all examined setups, we confirmed that a significant improvement can be expected from essentially only T+C, i.e., a setup that improves target-side morphological coherence by employing an additional language model over morphological tags. This setup does not allow the MT system to produce any word forms that were not seen in both the parallel and monolingual training data, but it improves the probability estimates of word form sequences.

4.2.2 Morphological Explosion on the Fly

The T+T+G setup illustrated in Figure 4.4 unfortunately works only with extremely small datasets (at generally low levels of overall performance). As soon as the parallel corpus becomes reasonably big, T+T+G introduces a loss of essential details and more importantly leads to an explosion of the search space: too many possible word forms have to be generated and scored. Consider our setup where all combinations of lemmas and tags have to be produced and evaluated. For instance for the Czech word *stát* (one of the possible translations of the English word *state*, both the verb and the noun), this amounts to 347 possible Czech word forms (or 182 word forms when dialects and archaic forms are excluded) according to Hajič (2004).

Containing this explosion proved impossible given the design of factored translation models. The models are said to be **synchronous**, i.e., translation options have to be fully generated (all target factors filled) before the main search starts. While we can prune this space by dropping less promising translation options, the scores available at this early stage are only *local*, they cannot consider the context of surrounding words because it will be (gradually) built only later in the main search. At the same time, many morphological features express the relation of words to the context. Dropping some "unlikely" case variations of a noun before the verb is known will inevitably fail because it is the verb that requests a particular case.

In the following section and also later in Chapter 5, we present techniques that avoid these problems.

4.3 Producing Unseen Word Forms

Table 4.1 motivated the need to generate Czech word forms on the fly but in Section 4.2.2, we explained that simply allowing to generate word forms from combinations of lemmas and tags doesn't work.

In this section, we summarize three methods we proposed as possible solutions: two-step translation, reverse self-training and an integrated discriminative model.

Src	after a sharp drop		
Mid	po+6	ASA1.prudký	NSA-.pokles
Gloss	*after+loc*	*adj+sg...sharp*	*noun+sg...drop*
Out	po	prudkém	poklesu

Figure 4.5: An illustration of two-step translation: translating from English to lemmatized Czech (Mid) and only then inflecting.

4.3.1 Two-Step Translation

In Bojar and Kos (2010),[5] we presented the idea of two-step translation to avoid the explosion of variants of words and the difficulties of pruning them before the surrounding context is available. In **two-step translation**, the search is divided into two consecutive phases, see Figure 4.5 for an illustration:

1. **Reordering and lexical choices.** The input sentence is translated into an intermediate "language" that disregards morphological attributes implied solely by the target language. The desired number of tokens, their positions and meaning-bearing morphological features (e.g., plural for nouns or negation) are preserved.[6]

2. **Morphological choices.** The intermediate representation is inflected, preserving the number and order of tokens.

The benefit from phasing the search into two independent steps is that the inflection in Step 2 have full access to the context of surrounding words. Generating all forms is acceptable because they can be effectively pruned without risking serious search errors suffered by T+T+G (Section 4.2.2).

Technically, we realized both steps as factored PBMT setups. Step 1 was trained on parallel data, with standard limits on reordering and target side simplified to lemmas and a hand-picked subset of morphological features.

Step 2 was a monotone word-for-word "translation": the translation model (a phrase table with all phrases limited the length of one token) mapped each simplified Czech word to all possible regular word forms and the standard language model ensured selecting coherent combinations. Since Step 2 was mapping between simplified Czech and regular Czech, we could train it on (large) Czech-only texts.

Compared to the T+C baseline (Section 4.2), our results in Bojar and Kos (2010) were mixed: the two-step translation improved over the baseline in small data setting but not in large data setting.

[5] (Bojar and Kos, 2010) is reprinted as Appendix A.5 on page 127.

[6] Prior work of Minkov *et al.* (2007), Toutanova *et al.* (2008), or Fraser (2009) disregarded all morphological information and also targeted other languages.

	Source English		Target Czech
Para 126k	a cat chased...	=	**kočka** honila...
			kočka honit... (lem.)
	I saw a cat	=	viděl jsem **kočku**
			vidět být kočka (lem.)
Mono 2M	?		četl jsem o **kočce**
			číst být o kočka (lem.)
			Use reverse translation
	I read about a cat	←	backed-off by lemmas.

⇒ A new phrase learned: "about a cat" = "o **kočce**".

Figure 4.6: The key idea of reverse self-training: The English word *cat* is present in the parallel corpus but its Czech counterparts do not cover all morphological cases of the word, the locative *kočce* is missing. Translating (based on lemmas) a sentence with this particular form from the monolingual data adds this form in its correct context to the translation model.

We continued in exploring two-step setups with our PhD student in Bojar *et al.* (2012a) and Jawaid and Bojar (2014) with no significant gains. The area was also subsequently studied by others, most recently Burlot *et al.* (2016) who explored several other technical realizations of step 2, generally confirming smaller gains as parallel training data grow. At about 1M parallel sentences, there is little or no benefit from the separation.

4.3.2 Reverse Self-Training

In Bojar and Tamchyna (2011a) and further in Bojar and Tamchyna (2011b),[7] we realized that the decision capacity about word forms lies ultimately in the language model. If word form combinations (such as an agreeing pair of an adjective and a noun) are known to the language model, it will promote them. And conversely, any unknowns will force the system to fall back to denser statistics, e.g., to shorter n-grams or (if linguistically-informed models are available) to lemmas or tag sequences. Any cleverness in offering word forms in the translation model is not going to provide any improvement if the language model cannot support the proposed sequence. In other words, it is the intersection of the translation and language model capabilities that is capping the performance of the system.

Assuming that we have trained the LMs of the system the best way we could (used all possible data, used LMs over different linguistic factors), we must ensure that the translation model is not adding further limitations and that it is offering translation candidates that the LM can effectively evaluate. Any *further* candidates, coming for example from a morphological generator, are not going

[7] (Bojar and Tamchyna, 2011b) is reprinted as Appendix A.6 on page 135.

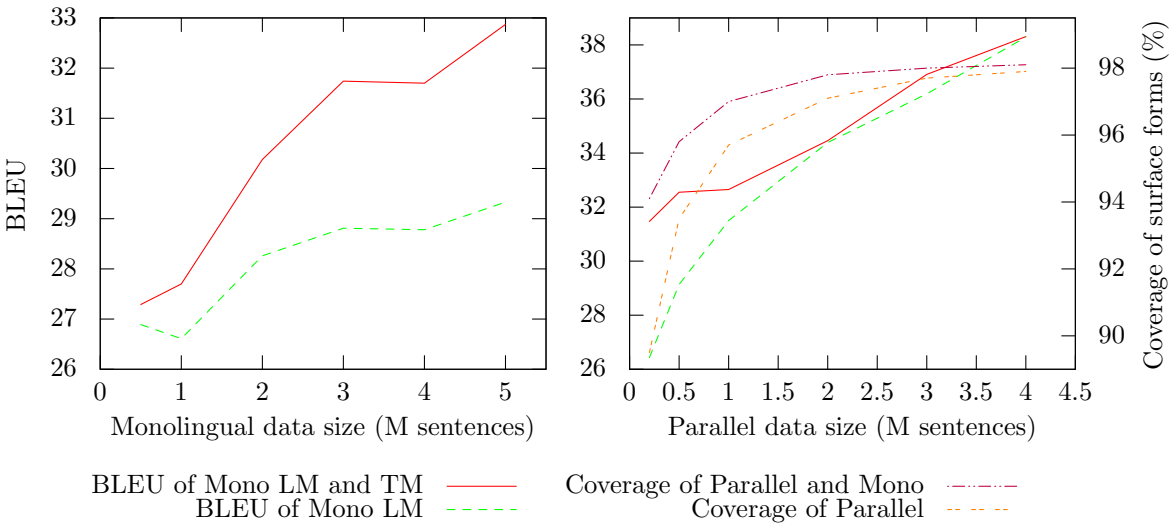


Figure 4.7: Improvements in BLEU score thanks to reverse self-training when adding monolingual data to fixed parallel data (500k sentences, left plot) and when increasing parallel data size with fixed monolingual (5M sentences, right plot). Reproduced from Bojar and Tamchyna (2011b).

to be used anyway because they are not known to the LM (and LM will thus score them lower than other options).

We thus proposed **reverse self-training** as a technique that ensures that the TM is *as capable as* the LM in producing word forms. Given that the LM is trained on generally much larger training data (monolingual texts), we must somehow incorporate these texts into the training of the TM.

The key idea is to use back-translation to translate the target-side monolingual data to the source language and use this synthetic parallel corpus to train the forward system. Back-translation was used previously by Bertoldi and Federico (2009) and became extremely popular recently in neural MT (Sennrich *et al.*, 2016) but one aspect remains unique to our setup.

As illustrated in Figure 4.6 on the preceding page, we back-off the back-translation system to translate from lemmas if the exact word form is not known. If the original source language (English, in our setup) is morphologically less rich, the translation from lemmas will not cause any harm. The forward system will then see a good English sentence or phrase translated to a perfect Czech phrase, containing a word form never seen in the small parallel data. The forward system thus gets the chance to learn a new form of a known word in its correct context.

Figure 4.7 shows the benefits of reverse self-training for English-to-Czech translation. It is well known that increasing LM size is always beneficial (Brants *et al.*, 2007), see the "BLEU of Mono LM" curve in the left plot. Our technique allows to exploit the given monolingual data much better, see the curve "BLEU

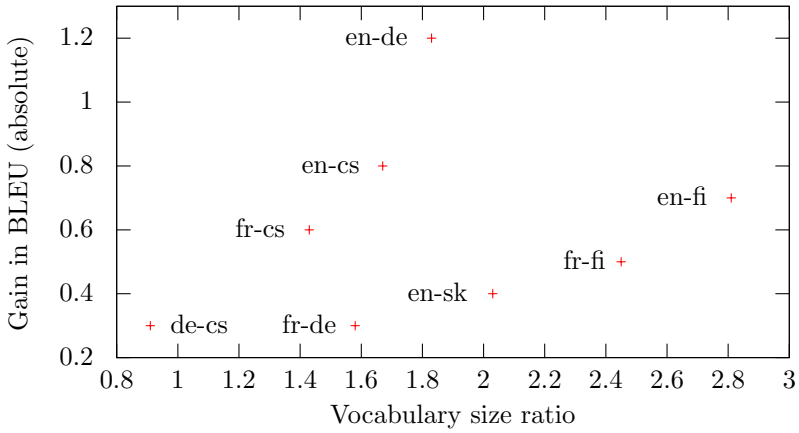


Figure 4.8: Reverse self-training for more language pairs. Reproduced from Bojar and Tamchyna (2011b).

of Mono LM and TM" in the left plot. In the right plot, we can see that the gains diminish as the parallel data grow. The benefit from reverse self-training started at 4 BLEU points but becomes negligible from about 2M of parallel sentences.

Figure 4.8 documents the effectiveness of the method for several language pairs, in relation to their morphological richness. All the underlying experiments used 94–128k parallel sentences and 662–896k monolingual sentences. "Vocabulary size ratio" indicates how many more distinct word forms the target language had in the parallel corpus compared to the source. The extreme is English-Finnish with 2.8× more Finnish forms. The tendency is clear: the richer the target language is compared to the source, the larger the gain. If both languages are rich, such as German-Czech, the benefit is not necessarily big.

4.3.3 Unseen and Discriminatively Trained

As we know from the previous section, having a parallel corpus of 2M sentences for languages like Czech may already be sufficient but arguably, many language pairs suffer from lack of resources much more. Examining methods for particularly low-resource settings is thus interesting.

In the situation when the necessary (target) word forms are not available even in the monolingual data, we have to rely on morphological analyzers and generators, and their dictionaries. Since the dictionaries (naturally) do not provide frequencies or probabilities of the forms in their contexts, we have to rely on a different scoring mechanism.

One option would be to use standard language models in the factored setup (Section 4.2), trained over sequences of morphological tags and (separately) over lemmas. The best form would be selected based on a weighted combination of these scores.

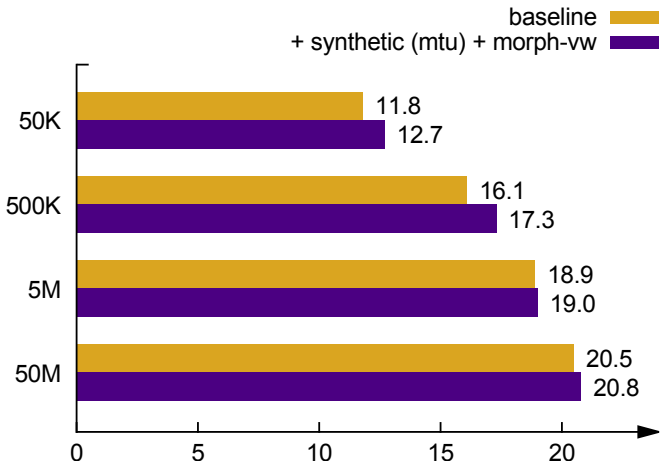


Figure 4.9: The improvement in BLEU thanks to including automatically generated word form variations of translation options ("synthetic (mtu)") and scoring them with the discriminative model ("morph-vw"). Reproduced from Huck *et al.* (2017).

Aleš Tamchyna's thesis examined more fine-grained models, namely **discriminative models** (Tamchyna, 2017). The discriminative model is trained outside of the translation system and allows to include many more features, including fully lexicalized ones (e.g., indicators checking for the presence of individual word forms or lemmas). One of the advantages is that it has the power of learning **valency frames**, that is the requirements of verbs for a particular preposition or case of their arguments.

The integration of such a rich model into the PBMT search is technically challenging because the model is evaluated before pruning for a very large number of translation options. Tamchyna *et al.* (2016a) had to come up with a sequence of optimization tricks to avoid any duplicated calculation. The benefit of this optimization was that the discriminative model could use also a limited context of the *target side*, i.e., the previous word or two of the current partial hypothesis.

In Huck *et al.* (2017), the discriminative model was trained *excluding* the exact word forms and relying only on individual morphological features and the lemma. This allowed to reliably score even word forms generated by the morphological generator; in the case of Czech, Morphodita (Straková *et al.*, 2014) was used. The method is effective especially with corpus sizes of 50k and 500k sentences, small gains are however observed also at 5M and 50M sentence pairs.

Chapter 5

Benefiting from Deep Syntax in MT

The methods and experiments described so far were limited to using relatively shallow linguistic information: lemmatization, tagging, and morphological generation.

In this chapter, we summarize one of our key contributions of this thesis, namely the incorporation of deep-syntactic knowledge to phrase-based MT. We note that we explored this topic already in our PhD thesis (Bojar, 2008), but the approach taken then was not successful.

As we documented for dependency trees used for translation between English and Czech in Bojar and Hajič (2008) and further in Bojar and Týnovský (2009) and as Chiang (2010) described independently for constituency trees for translation from Chinese or Arabic into English, a statistical transfer-based system where the minimum translation units are linguistically-adequate treelets has a considerably harder situation than phrase-based MT or its extension, hierarchical phrase-based translation (Chiang, 2005).

In Section 5.1, we briefly review the problem. Our technique that allows to circumvent it is summarized in Section 5.2, the underlying reasons of its effectiveness are further explained in Section 5.3 and empirical results are provided in Section 5.4.

5.1 Brief Summary of Difficulties with Tree-Based Transfer

In our PhD thesis, we attempted to improve the grammaticality of MT by implementing a transfer-based MT system. Such systems first analyze the input sentence into a formal representation reflecting its syntax and/or semantics, then convert this representation to a corresponding formal representation for the target language and finally generate the plain text in the target language.

The fact that the target string is produced from a formal representation would ideally guarantee that the output will be grammatical and the separation of source linguistic analysis and target generation potentially reduces the need for (large) bilingual training data, benefiting from the generalizations that can be observed monolingually or provided in the form of dictionaries.

In practice, the transfer-based approach fails to surpass shallow methods like phrase-based MT on average, due to especially the following issues (Bojar and Hajič, 2008):

- **Cumulation of errors** when preparing the source and target formal representations of the parallel data. In our case, a tagger was followed by a surface-syntactic parser and then a deep-syntactic parser. If any of them made an error (or if the sentence in the training data was not exactly grammatical, according to the rules embodied in the particular tool or matching the training data behind the tool), the resulting structure contained an error. Shallow methods, on the other hand, suffer only from errors genuinely present in the training data.

- **Mismatching structures** between the source formal representation, the target representation and their alignments prevent extraction of translation counterparts. As outlined above, classical SMT assumes that both source and target can be decomposed into some units, corresponding to one another. If the units follow the syntactic structure of the sentence, as was our case, the decomposition must conform to the structures of both source and target. The underlying grammar formalisms and parsers for the two languages were however built independently and arbitrary decisions as well as natural divergence between languages (Dorr, 1994; Šindlerová *et al.*, 2014) render the sub-structures not matching exactly. Commonly, one accepts only matching sub-structures into the automatically collected "translation dictionary". This means a considerable data loss in comparison to PBMT, where only the word alignment is constraining which pairs of substrings are learned from the data.

- **Increased data sparseness** due to fine-grained details of the deep analysis. As described in Bojar and Týnovský (2009), the core of our approach was a formalism for tree-to-tree transfer (synchronous tree substitution grammars, Eisner, 2003), which assumed operating on trees with atomic nodes. In practice, the nodes of the deep syntactic representation had many attributes, and their values were indeed necessary in order to be able to generate the target sentence correctly. If one combined all the attributes into an atomic unit, the vocabulary size of these units was actually larger than the vocabulary of word forms because the deep representation made finer distinctions. The factorization of translating lemmas and morphological tags separately as discussed for PBMT in Section 4.2 was therefore *necessary*, risking a combinatorial explosion during the translation.

Carefully constructed systems, such as TectoMT (Popel and Žabokrtský, 2010), can to some extent circumvent these shortcomings. For instance, TectoMT still builds upon the assumption that the source and target representations are isomorphic, reducing the transfer to the search for the best labelling of

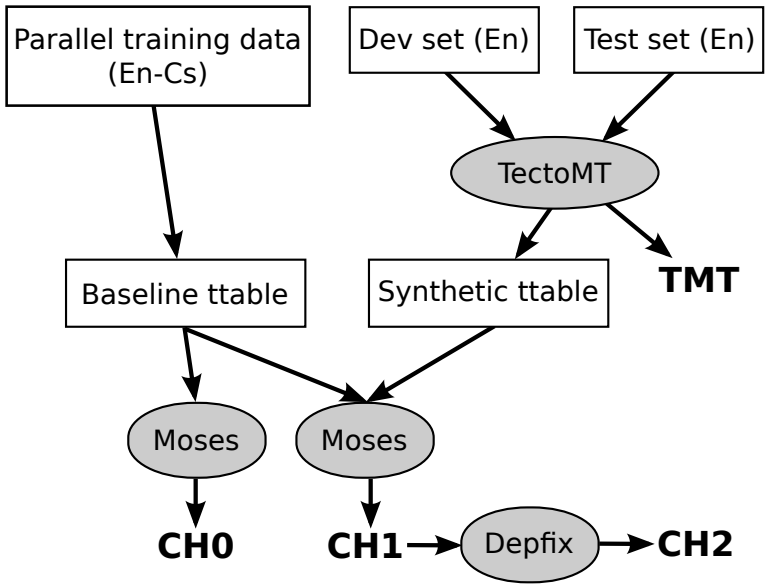


Figure 5.1: Setup of Chimera. Reproduced from Tamchyna and Bojar (2015).

the source-side structure with target-side lemmas and morpho-syntactic labels, so-called "formemes" (Žabokrtský *et al.*, 2008; Dušek *et al.*, 2012). We aimed at a more general data-driven method that would be easier to reuse for other languages, but failed.

While the approach of TectoMT is linguistically appealing and in many cases, it indeed produced grammatically better output than PBMT, it never surpassed PBMT on unconstrained input on average.

5.2 Chimera: Deep-Syntactic and PBMT Systems Combined

In Bojar *et al.* (2013c)[1] and subsequent publications (Tamchyna *et al.*, 2014; Bojar and Tamchyna, 2015; Tamchyna *et al.*, 2016b; Bojar *et al.*, 2017d), we proposed and tested a method that combines the benefits of TectoMT and PBMT. The resulting system was called "Chimera", in reference to the three-headed mythical creature; the third "head" was Depfix (Rosa *et al.*, 2012).

Figure 5.1 schematically illustrates the design of the system combination: the central component is Moses trained on large parallel data and with the best-performing setup (the T+C factored system) as described in Section 4.2. This setup alone is denoted CH0 in the following.

The transfer-based system TectoMT is included in a rather simple but surprisingly effective fashion: TectoMT translates the source side of both the test

[1](Bojar *et al.*, 2013c) is reprinted as Appendix A.7 on page 143.

I	saw	two	green	striped	cats	.
já	pila	dva	zelený	pruhovaný	kočky	.
	pily	dvě	zelená	pruhovaná	**kočky**	
	...	**dvě**	zelené	pruhované	koček	
	viděl	dvou	**zelené**	**pruhované**	kočkám	
	viděla	dvěma	zelení	pruhovaní	kočkách	
	...	dvěmi	zeleného	pruhovaného	kočkami	
	viděl jsem		zelených	pruhovaných		
	viděl jsem		zelenými	pruhovanými		
	viděla jsem	**dvě zelené**		**pruhované kočky**		
		dvě zelené pruhované kočky				

Figure 5.2: Translation options available to CH1: the majority of options come from the corpus and some combination of them hopefully leads to a good translation, underlined. TectoMT provides synthetic options (in bold) that easily match longer sequences of input.

and the development set, leading to a synthetic parallel corpus. The corpus (of a size corresponding to the development and test set, i.e., a few thousand sentence pairs at most) is then processed in the standard "PBMT way": automatic word alignment followed by phrase extraction. We obtain a standard phrase table (the "syntetic ttable" in Figure 5.1) and provide it to Moses, in addition to its standard corpus-based table. Moses has thus the chance to use phrases constructed by TectoMT. Finally, the output is processed by Depfix.

For clarity, we denote the stages of this system TMT (TectoMT alone), CH0 (Moses alone), CH1 (Moses with TectoMT) and CH2 (the full combination). In this chapter, we focus only on the first two components and their interaction.

5.3 Analysis of the Combination

In Tamchyna and Bojar (2015), we carefully analyzed the behavior of the combined system. Technically, the two phrase tables simply provide translation options (as discussed in Section 4.1) to a common pool and the standard search is free to select any of them. Each of the phrase tables comes with its separate phrase penalty, so the model weights can influence whether translation options from one of the tables should be used more often on average.

The nature of the phrases from the CH0 and TMT phrase tables is however rather different. The CH0 table was extracted from a large parallel corpus and, depending of the repetitiveness of the domain and its match with the test data, the source sentence cannot be generally covered with very long phrases, simply because the exact wording is not likely to be seen in the training data.

The TMT table, on the other hand, was created from the source sentences and

all	different?	reachable?	score diff
3003	2665	1741	1601 (<)
			140 (>)
		924	(unreachable)
	338	(identical)	

Table 5.1: Forced decoding—an attempt of CH0 to reach the test set translations produced by CH1. Reproduced from Tamchyna and Bojar (2015).

therefore matches exactly the current source. Much longer phrases can be thus used, as illustrated in Figure 5.2. The CH0 phrase table may have contained all the necessary forms, but they were generally collected from separate sentences. The options by TectoMT may often contain identical words (thus slightly increasing the issue of spurious ambiguity), but it provides them in a longer sequence. The gradual expansion of hypotheses has thus the chance to "jump over" all the combinatorial explosion when searching for a matching combination of word forms.

The language model is applied as usual, giving the combined system the capacity to reject strange parts of the translation that TectoMT may have produced.

Following our discussion on local and non-local features and conflicting structures, our method relieved the language model from being the only source of horizontal coherence of the sentence. Phrases from TectoMT reflect grammatical relations between words locally, within the phrase. The deep-syntactic analysis in TectoMT was useful for producing such phrases but this different structuring along the deep-syntactic tree does not interfere with the simple phrase segmentation of PBMT, thanks to our combination method.

Table 5.1 on the current page documents that TectoMT provided also words not available to CH0. We ran CH0 in the so-called "forced" or "constraint" mode (Schwartz, 2008), checking if it can produce translations created by CH1, i.e., the model with access to TectoMT translations. Out of the 3003 sentences in the WMT14 news test set, CH0 and CH1 produced identical output in 338 cases. In about a third (924) of the remaining sentences, CH0 could not reach the output of CH1, which means that TectoMT either provided a word form never seen in the parallel training data (52M sentences in this experiment), or not seen enough to survive the necessary technical thresholds that disqualify infrequent translations (up to 100 options are considered from each phrase table for each source span).

Figure 5.3 illustrates the complementary benefits of CH0 and TMT, and the ability of CH1 to select the better of each of them. While CH0 makes better lexical choices esp. at the beginning of the sentence when translating the expression *living zone*, it suffers from bad morphological choices at the end of the sentence. The combined system CH1 produces a perfect output for this sentence snippet.

Src	the living zone with the dining room and kitchen section in the household of the young couple .
Ref	obývací zóna s jídelní a kuchyňskou částí v domácnosti mladého páru . *living zone with dining and kitchen section in household* $young_{gen}$ $couple_{gen}$.
CH0	obývací zóna s jídelnou a kuchyní **v sekci** domácnosti **mladý pár** . *living zone with dining_room and kitchen* **in section** $household_{gen}$ $\mathbf{young}_{nom}$ $\mathbf{couple}_{nom}$.
TMT	**živá** zóna **pokoje** s jídelnou a s kuchyňským oddílem v domácnosti mladého páru . **alive** *zone* $\mathbf{room}_{gen}$ *with dining_room and with kitchen section in household* $young_{gen}$ $couple_{gen}$.
CH1	obývací prostor s jídelnou a kuchyní v domácnosti mladého páru . *living space with dining_room and kitchen in household* $young_{gen}$ $couple_{gen}$.

Figure 5.3: Example of translations of Moses (CH0) and TectoMT alone and their phrase-based combination CH1. Errors are in bold, glosses are in italics. Reproduced from Tamchyna and Bojar (2015).

5.4 Empirical Results

We used Chimera in five years of WMT evaluation campaigns, as documented in Table 5.2. During the years 2013–2015, it scored best and it surpassed Google MT significantly in the years 2013–2016.

The table also documents the transition towards neural MT. The first NMT system to join English-to-Czech task was MONTREAL (Jean *et al.*, 2015) and it ended up third or fourth in manual evaluation in 2015. In 2016 and 2017, NMT has proved its superiority.

In Sudarikov *et al.* (2017), we experimented with neural MT but our purely neural approach did not perform well due to various reasons, including the shortage of computing resources (large-memory GPU cards). We nevertheless strongly benefited from NMT outputs by integrating them to our submission in the style of Chimera, adding them in a separate phrase table. Chimera without NMT reached BLEU of 18.3 and NMT allowed an increase to 20.5.

Table 5.2 is sorted by BLEU but it should be noted that this automatic score does not always match human judgements. The most striking difference is seen in WMT17 where our combination including NMT surpassed Google NMT setup in both BLEU and TER but considerably lost in manual scoring. We see this as an indication that humans demand *overall* sentence coherence. This can be achieved by NMT thanks to its avoidance of the assumption of translation units. PBMT, even if provided with well-formed long phrases (from TectoMT or NMT), lacks the capacity to ensure this coherence, and BLEU lacks the capacity to evaluate long-range phenomena.

The disparity between manual and automatic evaluation methods leads naturally to the last large topic in our work, MT evaluation, as described in the next chapter.

	System	BLEU	TER	Manual
WMT13	CH2	20.0	**0.693**	**0.664**
	CH1	**20.1**	0.696	0.637
	CH0	19.5	0.713	–
	Google Translate	18.9	0.720	0.618
	CU-TECTOMT	14.7	0.741	0.455
WMT14	CH2	21.1	0.670	**0.371**
	UEDIN-UNCONSTR.	**21.6**	**0.667**	**0.356**
	CH1	20.9	0.674	**0.333**
	Google Translate	20.2	0.687	0.169
	CU-TECTOMT	15.2	0.716	-0.175
WMT15	CH2	**18.8**	**0.715**	**0.686**
	CH1	18.7	0.717	–
	NMT: MONTREAL	18.3	0.719	0.467
	CH0	17.6	0.730	–
	Google Translate	16.4	0.750	0.515
	CU-TECTOMT	13.4	0.763	0.209
WMT16	NMT: UEDIN-NMT	**26.3**	**0.639**	**0.59**
	CH2	21.7	0.677	0.30
	Google Translate	23.2	0.678	0.19
	CU-TECTOMT	15.2	0.730	-0.03
WMT17	NMT: UEDIN-NMT	**22.8**	**0.667**	**0.308**
	CH2 incl. NMT	20.5	0.696	*0.050*
	NMT: Google Translate	20.1	0.703	0.240
	CH2	18.3	0.719	–

Table 5.2: Automatic scores (BLEU and TER) and results of manual ranking (where available) in WMT13–WMT17. The top other system and Google Translate reported for reference. Bold indicates the best system in each metric, or more systems, if the difference between their manual scores was not sufficiently large for statistical significance.

Chapter 6

Precise MT Evaluation

This chapter summarizes our contributions to the understanding of how to distinguish between good and bad translations.

As mentioned above, MT evaluation serves several purposes and each of them requires a slightly different approach:

- For day-to-day progress check, we need fast and reproducible methods that reflect well overall translation quality *as well as* the problems we want to focus on. Standard automatic evaluation methods may easily neglect our current research target (e.g., translating pronouns or preserving negation), because it is not exhibited on a large portion of the output. Custom targeted methods, on the other hand, can easily overfit, i.e., provide a good score for the aspect their evaluate while ignoring an overall decrease in translation quality.

- For automatic training (model optimization), similarly fast and reproducible methods are necessary. In addition to this, they need to be sufficiently discerning even for very similar candidates (e.g., members of an n-best list). Most importantly, the methods need to be able to rule out poor candidates because otherwise, the optimization could converge to a bad optimum.

- For the selection of the best MT system from a set of fixed possible systems, we have to ask what is the planned use of the MT system: will someone post-edit the translations, or will they be automatically indexed for full-text search, or will someone read them (with or without access or mild understanding of the source)? Each of these uses can lead to a different choice.

  In contrast to the previous situations, most of the compared candidates will be already relatively good machine translations but they can differ considerably on the surface. Methods that work well for selecting the best candidate from an n-best list can fail when the hypotheses become less similar.

In general, both manual and automatic MT output evaluation methods are used. The main benefit of automatic methods is their reproducibility and low

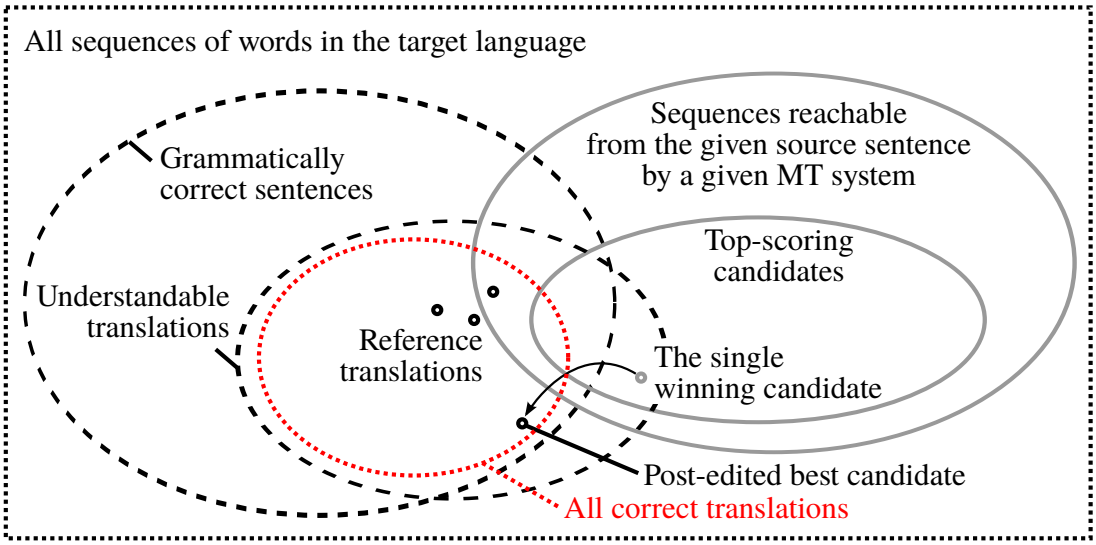


Figure 6.1: Space of possible translations. Reproduced from Bojar *et al.* (2013a).

cost, but they are obviously confined by their inherent assumptions and therefore often overestimate the quality of MT systems based on similar assumptions. Manual evaluation methods are expensive and the main problem is that they are never exactly reproducible because the annotator is affected by the sentences he or she has already evaluated. Reproducibility in manual evaluation can be improved by using large samples with many annotators, however it further increases the cost.

We have contributed to both manual and automatic methods of MT evaluation. In Section 6.1, we explain why MT evaluation is so difficult in general. In Section 6.2, we evaluate the importance of using more references. In Section 6.3, we add a complementary style of manual annotation and notice that PBMT tends to "swallow" words. Finally, Section 6.4 documents that BLEU scores are even less reliable when they are low, and explains why this is the case.

Furthermore, we have contributed to the development of methods of manual MT evaluation that operate along a structured representation of the meaning of the sentence, see Section 6.5.

As a meta-evaluation, automatic MT metrics are evaluated in terms of correlation with human judgements in annual evaluation campaigns, see Section 7.2.

6.1 Why Is MT Evaluation Difficult

It may not be obvious why evaluating MT is so difficult. We contributed to its understanding in Bojar *et al.* (2013a).[1]

[1] (Bojar *et al.*, 2013a) is reprinted as Appendix A.9 on page 165.

A ačkoli ho lze považovat za politického veterána, radní Březina reagoval obdobně.
Ač ho můžeme prohlásit za politického veterána, reakce radního Karla Březiny byla velmi obdobná.
A i přestože je politický matador, radní Karel Březina odpověděl podobně.
A přestože je to politický veterán, velmi obdobná byla i reakce radního K. Březiny.
A radní K. Březina odpověděl obdobně, jakkoli je politický veterán.
A třebaže ho můžeme považovat za politického veterána, reakce Karla Březiny byla velmi podobná.
Byť ho lze označit za politického veterána, Karel Březina reagoval podobně.
Byť ho můžeme prohlásit za politického veterána, byla i odpověď K. Březiny velmi podobná.
K. Březina, i když ho lze prohlásit za politického veterána, odpověděl velmi obdobně.
Odpověď Karla Březiny byla podobná, navzdory tomu, že je politickým veteránem.
Radní Březina odpověděl velmi obdobně, navzdory tomu, že ho lze prohlásit za politického veterána.
Radní Karel Březina, navzdory tomu, že ho můžeme označit za politického veterána, reagoval podobně.
Reakce K. Březiny, třebaže je politický veterán, byla velmi obdobná.
Velmi obdobná byla i odpověď Karla Březiny, ačkoli ho lze prohlásit za politického veterána.

Figure 6.2: Random sample from 71k possible translations of the English sentence: *And even though he is a political veteran, the Councilor Karel Brezina responded similarly.* Reproduced from Bojar (2012).

Given a fixed input sentence, it is easy to see that there are extremely many possible *erroneous* translations. We can start from any correct translation and modify it by introducing typing errors, altering morphological properties of words (e.g., the number or negation), reordering words or inserting or deleting words. The vast majority of these modifications will damage the translation—and a good MT system should avoid all these errors.

Starting from the other end, considering the set of *all correct translations* is not that straightforward. The situation can be schematically illustrated as in Figure 6.1 on the facing page.

In Bojar *et al.* (2013a), we attempted to quantify the number of *correct* possible translations from English into Czech. Inspired by the work of Dreyer and Marcu (2012), we designed a framework fit for morphologically rich languages and asked several annotators to provide as many good translations of a sentence as possible.

The results, in line with what Dreyer and Marcu (2012) observed for English, are rather interesting. An English sentence of 14 words can easily have 70 *thousands* of correct translations, as illustrated in Figure 6.2.

Each annotator in this exercise was instructed to spend up to two hours per sentence, using our tool to generate and validate sentences semi-automatically. The least prolific annotator provided this sentence with 350 possible translations, the second one created 3192 translations. And the most prolific one reached 67936 translations. Among these, only 8 translations were suggested by all three

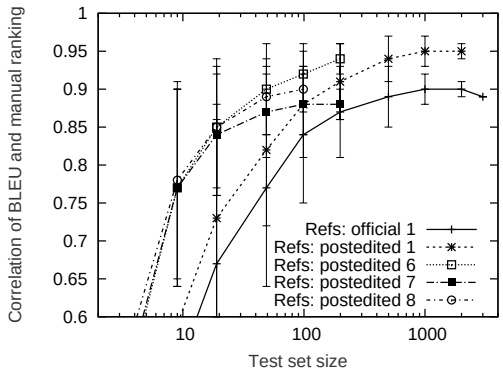


Figure 6.3: Correlation of BLEU and human judgements for varying type and number of reference translations. Reproduced from Bojar *et al.* (2013b).

annotators and only 172 translations were suggested by two of the three annotators. The space of possible translations is thus probably much larger.

The translations are not always 100% literal and they obviously differ in many more or less important aspects, such as register or style, information structure etc. If used in a coherent text and not as isolated sentences, many of these translations may not be acceptable at all, but for the current level of MT quality, all are equally good.

When designing automatic methods of MT evaluation, we thus have to keep in mind that the candidate translation produced by an MT system can be correct but superficially very distant from a given reference translation, or that it can be superficially very similar to the reference translation but suffer from serious errors.

6.2 More and/or Post-Edited References

The most widespread automatic MT evaluation method, BLEU (Papineni *et al.*, 2002), works by validating short fragments (1 to 4-grams) of the candidate translation against a provided reference translation. BLEU has been designed with the assumption that four independent human reference translations will be available, to allow for at least some variance in the MT output. However, BLEU is actually most often used with only one reference.

In Bojar *et al.* (2013b), we extended the manually-collected data of WMT13 with a substantial number of post-edited sentences. Through that experiment, we confirmed that BLEU becomes much more reliable with more references, but also found out that the *nature* of reference translations affects the correlation of BLEU and human judgements. The correlations are generally higher if the reference translations were created by post-editing MT outputs, i.e., if they are (very likely) more similar to the candidate translations.

	Google	Moses-Bojar	PC Translator	TectoMT	Total
Automatic: BLEU	13.59	**14.24**	9.42	7.29	–
Manual: Sentence ranking	0.66	0.61	**0.67**	0.48	–
Manual: Error flags	**2319**	2354	2536	*2895*	10104
Error flags details:					
Words with bad meaning	617	**587**	800	999	3003
Auxiliary word missing	84	111	96	138	429
Content word missing	72	*199*	**42**	108	421
Word form incorrect	783	735	762	713	2993
Superfluous word	381	313	353	394	1441
Non-translated word	51	53	56	97	257
Total serious errors	**1988**	1998	2109	2449	8544
Bad local word order	117	100	157	155	529
Punctuation error	115	117	150	192	574
...	...	...	...	...	...
Tokenization error	7	12	10	6	35

Table 6.1: A comparison of two types of manual evaluation (Sentence ranking and Flagging of errors) and BLEU scores for four English-to-Czech MT systems from WMT09. Noteworthy best results highlighted in bold, noteworthy worst results in italics. Adapted from Bojar (2011).

Figure 6.3 documents the situation. The generally lowest performance is obtained in the standard conditions with 1 "official" reference translation. The error bars reflect the variance due to random subsampling from the full 3k sentences and get narrower as larger and larger portion of the test set is used. With 2k or 3k sentences in the test set, the Spearman's rank correlation coefficient ρ reaches levels of 0.9. Using a single reference created by post-editing randomly selected systems from the set of evaluated systems works clearly better, reaching correlation of 0.95.

We also see from Figure 6.3 that the size of the test set and the number of references can somewhat compensate for each other. Specifically, the common practice of WMT shared translation tasks is to have about 3000 sentences with a single reference translation. A comparable correlation of BLEU and human judgements could be also achieved with just 100–200 sentences and 6–7 reference translations.

6.3 Error Annotations Help to Explain Bad Correlation for BLEU

In Bojar (2011)[2], we experimented with two techniques of detailed error analysis. One was based on semi-automatic interpreting of post edits of candidate

[2](Bojar, 2011) is reprinted as Appendix A.8 on page 151.

translations and another relied on manual flagging of errors using some error classification. Here is an example of the error flagging:

Source	Sarkozy meets angry fishermen.
Reference	Sarkozy jde vstříc rozhněvaným rybářům
Moses	Sarkozy se [MISSC: setkává] [MISSA: s] [FORM]rozzlobení rybáři.
TectoMT	Sarkozy [DISAM]splňuje [MISSC: vstříc] naštvané [FORM]rybáře.
Google	Sarkozy [LEX]splňuje [FORM]zlobit [FORM]rybářů.
PC Translator	Sarkozy se setkává [MISSA: s] [FORM]rozhněvané [FORM]rybáře.

In our annotation, we attached flags to individual tokens in MT output (and added tokens for missing words). The example illustrates errors in word form choice ([FORM]), word meaning (source word disambiguation [DISAM] and bad lexical choice [LEX]; the last two are difficult to distinguish and have the highest disagreement rate), as well as missing content ([MISSC]) and auxiliary ([MISSA]) words.

Both post-editing and error flagging led to similar conclusions about the MT systems competing in English-to-Czech translation back then: the traditional commercial system PC Translator was quite bad in lexical choice, TectoMT performed best in picking the right form of the word and phrase-based Google and our Moses were generally good in lexical choice but suffered from errors in morphology.

The flagging of errors also allowed to explain the bad performance of BLEU for this set of systems, see Table 6.1. Our Moses scored best according to BLEU but ended up third in terms of the WMT09 manual sentence ranking. As the detailed error flags reveal, the winning PC Translator made by far the least number of errors in the category of "Content word missing", while our Moses dropped almost five times more content words.

6.4 Low BLEU Scores Unreliable

In the English-to-Czech evaluation campaigns 2009 and 2010, we saw a strikingly low correlation between human judgements about translation quality and BLEU scores, see the left part of Figure 6.4.

While the correlation of BLEU and human judgements for Czech was low, we found in Bojar *et al.* (2010a)[3] a high correlation between the *absolute BLEU scores* and their correlation to human judgements across all language pairs taking part in WMT09, see the right part of Figure 6.4. Put simply, BLEU scores below 20 are not reliable.

Bojar *et al.* (2010a) have also explained the reason for this. The situation is illustrated in Table 6.2 which compares the sets of n-grams in outputs of several MT systems deemed correct according to (1) the presence of the n-gram in the reference translation vs. (2) the absence of manual error flags described above.

[3](Bojar *et al.*, 2010a) is reprinted as Appendix A.10 on page 175.

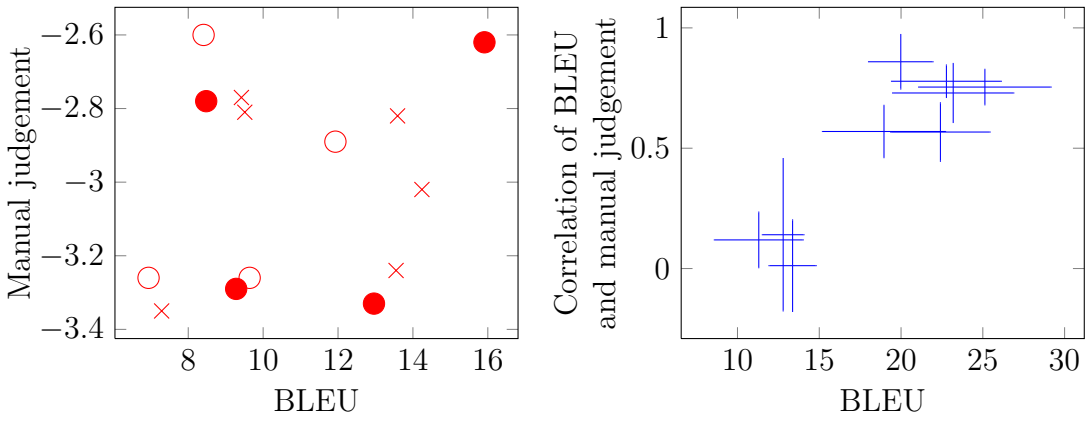


Figure 6.4: Left: Low correlation between BLEU and human judgements. Each point corresponds to one MT system, different point styles indicate a different test conditions. We see no correlation between BLEU and manual judgement. Right: A good correlation between the BLEU scores and their correlation with human judgements, i.e., higher BLEU scores correlate well with humans and lower BLEU scores do not. Each cross corresponds to one language pair, showing the average and standard deviation of BLEU scores and manual judgements across all systems for that language pair. Simplified from Bojar *et al.* (2010a).

Two situations are desirable: when the n-gram does not contain errors and it is confirmed by the reference, and when the n-gram contains errors and the reference does not confirm it. This happens for 59% of unigrams and 56% of bigrams, etc. False positives (n-grams confirmed but containing an error) are luckily rather rare: 6% of unigrams, 2% of bigrams, etc.

The reason for unreliability of BLEU at low scores lies in the fourth case: error-free n-grams that are nevertheless not available in the reference. BLEU does not give any credit to them but the systems can quite differ in the quality of translation in these cases. As seen in Table 6.2, this amounts to more than a third of unigrams, 43% of bigrams etc.

Post-edited references discussed in the previous section are much closer the to candidate translations and don't suffer from this lack of coverage. The unconfirmed n-grams will be only those where the post-editor needed to rephrase the sentence to fix some error or disfluency. Any decrease in BLEU will thus correspond to genuine issues of the candidate translation.

In Bojar *et al.* (2010a), we proposed to increase the coverage of BLEU by matching the candidate with the reference at a coarser level of representation, namely bags of deep-syntactic lemmas (separate for each deep-syntactic part of speech) instead of the common longer n-grams of exact word forms. For English-to-Czech, this increased the correlation in that particular experiment from 0.33 to 0.53.

Confirmed by Ref	Contains Errors	1-grams	2-grams	3-grams	4-grams
Yes	Yes	6,34 %	1,58 %	0,55 %	0,29 %
Yes	No	36,93 %	13,68 %	5,87 %	2,69 %
No	Yes	22,33 %	41,83 %	54,64 %	63,88 %
No	No	**34,40 %**	**42,91 %**	**38,94 %**	**33,14 %**
Total n-grams		35 531	33 891	32 251	30 611

Table 6.2: n-grams as confirmed by the reference and/or by containing or free from errors according to manual error flagging. Lack of coverage of the reference highlighted in bold. Reproduced from Bojar *et al.* (2010a).

In Macháček and Bojar (2011), we further elaborated on that, moving back to the less computationally-demanding shallow but still sufficiently coarse features of words. We also confirmed the applicability of the proposed method in model optimization, performing acceptably in the main manual scoring that rewarded tied results and getting the best score when ties were disfavored (Callison-Burch *et al.*, 2011). See also Section 7.1 for a discussion the manual evaluation method.

6.5 MT Evaluation Focused on Semantics

With the success of neural MT, the focus of MT evaluation has to be changed as well. Multiple studies (Bentivogli *et al.*, 2016a; Bojar *et al.*, 2016a; Castilho *et al.*, 2017b,a) suggest that NMT primarily improves fluency. Adequacy of translations is improved as well, but to a smaller extent. We would therefore expect that, on average, misunderstandings due to MT errors will be less frequent, but at the same time, they will be harder to notice: MT output will be more often seemingly perfect but including a semantic flaw.

For that reason, we have revived our interest in semantic correspondence between the candidate translation and the reference. In Bojar and Wu (2012), we experimented with HMEANT (Lo and Wu, 2011), a manual method of MT evaluation based on aligning the predicate-argument structures of the candidate and the reference. Building upon that, we designed a manual method of MT evaluation that closely follows the semantic structure of the source sentence (and not the reference, thereby avoiding the need to parse the often garbled MT output) in a joint work (Birch *et al.*, 2016).

Chapter 7

Shared Tasks

To reliably measure progress of the field of natural language processing and machine translation in particular, approaches to problems and proposed solutions have to be regularly compared in a rigorous way. Such a comparison is however often difficult to achieve due to many interacting conditions and generally large efforts are needed.

The common practice in NLP resolves this by **shared tasks**: regularly or independently organized events where the organizers specify an exact task description and usually provide training datasets and then collect submissions from participants to evaluate them in a clear and comparable way.

The history of shared tasks related to machine translation has been summarized in Bojar *et al.* (2016c)[1] for WMT (originally Workshop on Statistical Machine Translation which became an ACL-sponsored conference in 2016) and in Bentivogli *et al.* (2016b) for IWSLT, a workshop focused primarily on the translation of spoken language.

Over the years, our contribution to the course of WMT shared task has been twofold: (1) contributing to best practices in MT evaluation, and (2) co-organizing various tasks. We summarize these contributions in the following sections.

7.1 Avoiding Bias in WMT News Translation Task

The main shared task at WMT is translation of news text, see Koehn and Monz (2006) through Bojar *et al.* (2017a). Thanks to our participation in the EU project EuroMatrix[2] and subsequent EU projects within the 6th and 7th Framework Programmes and in H2020, Czech has been included in this task every year since 2007. We also participated in the task with our translation systems of diverse nature.

Up until 2016, the main WMT evaluation measure was derived from annotation screens of up to five systems ranked manually according to the perceived translation quality. The annotators were presented with the source, the reference translation and 5 candidate outputs and they indicated the relative quality of

[1] (Bojar *et al.*, 2016c) is reprinted as Appendix A.12 on page 193.

[2] `http://www.euromatrix.net/`

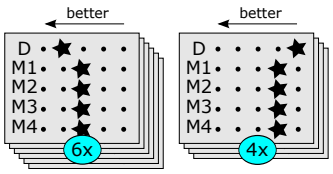


Figure 7.1: Illustration of an artificial collection of manual rankings as used in WMT until 2016. The sample annotation consists of 10 annotation screens in total, in 6 of which the system D wins and in 4 of which it loses. Its four competitors M1...M4 are always on par. Individual annotation screens may be provided by different people.

Interpretation	"≥ Others"	"> Others"	"Ignore Ties"
Formula	$\frac{\text{wins+ties}}{\text{wins+ties+losses}}$	$\frac{\text{wins}}{\text{wins+ties+losses}}$	$\frac{\text{wins}}{\text{wins+losses}}$
Favors	"mainstream"	"distinct"	-
D	$6 \times 4 = 24/40$	**24/40**	24 / 40 = **6/10**
M1	$10 \times 3 + 4 =$ **34/40**	4/40	4/10

Figure 7.2: Various ways of handling ties in WMT ranking. The calculations are based on the sample annotation from Figure 7.1. When ties are rewarded ("≥ Others"), the tying systems M1...M4 "support" each other and each of them thus seems to perform better than D (34/40 over 24/40 wins), unduly favouring similar systems. Penalizing ties ("> Others") promotes distinct systems like D. "Ignore Ties" is a fairer option, for which we advocated in Bojar *et al.* (2011).

these translations; see Figure 7.1 for sample dataset of judgements (the underlying sentences were selected randomly from the test set and were not important when interpreting the evaluation, we thus omit them in the picture). In practice, the exact set of 5 ranked systems differed from screen to screen, sub-sampling five-tuples from all the competing systems.

Observing the performance of our systems in 2010, we noticed that the same collected judgements can be interpreted in subtly different ways, leading to different results. We thus carefully analyzed the discrepancies and reported them in Bojar *et al.* (2011). Here we highlight two of the issues:

Rewarding ties unduly favors similar systems. Figure 7.2 illustrates that depending on the treatment of cases where more systems receive the same rank in an annotation screen, the final ordering of the systems can differ. Specifically, WMT used to rely on a formula that rewards ties ("≥ Others"; "systems ... are ranked based on how frequently they were judged to be better than or equal to any other system", Callison-Burch *et al.*, 2010). This choice can be considered particularly problematic since several system

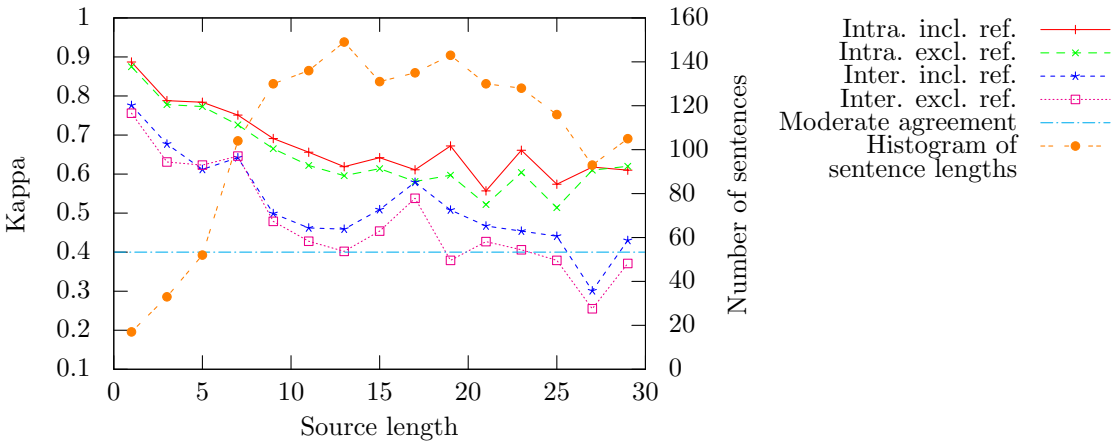


Figure 7.3: Intra- and inter-annotator agreement in terms of the kappa statistic (left axis) of WMT10 evaluation, including of excluding comparisons with reference translations. "Histogram of sentence lengths" (right axis) shows the distribution of sentences in the test set. Adapted from Bojar *et al.* (2011).

submissions were always based on the Moses translation system, where similar translation quality can be expected.

Agreement rates decrease with sentence length. The aggreement rates between different people (inter-) and between annotations of the same person (intra-) have been reported along with the results since Callison-Burch *et al.* (2007), in the form of Cohen's kappa (Bennett *et al.*, 1954). In Bojar *et al.* (2011), we noted that the agreement decreases with sentence length as illustrated in Figure 7.3. Following indicative ranges for the kappa statistic,[3] we see that the inter-annotator agreement when comparing two real systems (as opposed to one system and the reference translation) gets close or below what Landis and Koch (1977) suggest as moderate agreement. Importantly, it turns out that the majority of the evaluated sentences are of this length.

Our discussion sparkled further research and evolution of the method of manual ranking (Lopez, 2012; Koehn, 2012; Hopkins and May, 2013). The current method called "direct assessment" (Graham *et al.*, 2016) simplifies the task by evaluating only one candidate at a time and asking the annotator to provide a score on an effectively continuous *absolute* scale given only the reference translation, not the source. Direct assessment became the official method only in 2017 (Bojar *et al.*, 2017a) so we still anticipate further developments in this area in the coming years.

[3]However, see the discussion in Komagata (2002).

	'07	'08	'09	'10	'11	'12	'13	'14	'15	'16	'17
Participating Teams	-	6	8	14	9	8	12	12	11	9	8
Evaluated Metrics	11	16	38	26	21	12	16	23	46	16	14
Baseline Metrics							5	6	7	7	7
System-level evaluation methods											
Spearman Rank Correlation	●	●	●	●	●	●	●	○			
Pearson Correlation Coefficient							○	●	●	●	●
Segment-level evaluation methods											
Ratio of Concordant Pairs		●	●								
Kendall's τ				❶	❶	❶	❷	❸	❸	❸	◗
Pearson Correlation Coefficient										○	◖
Tuning Task					●				●	●	

● main and ○ secondary score reported for the system-level evaluation.
❶, ❷ and ❸ are slightly different variants regarding ties.

Table 7.1: Summary of metrics and tuning tasks over the years. The vertical bar indicates since when we started co-organizing the task.

7.2 Organizing Shared Tasks

Since 2013, we have been actively involved in the organization of shared tasks of various types:

News Translation Tasks attract the largest number of participants each year. The main goal, translating short news stories, remains unchanged while the underlying set of languages slightly changes every year. The test sets for the task are created anew each year, to provide the participants with genuinely novel text. Huge collective effort is spent on manual evaluation and throughout the years (also due to our analysis presented in Section 7.1 above), the task saw a few modifications to the official method of evaluation.

Our contribution to the organization slightly varied through the years, but every year, we arranged the selection and fixes to the Czech part of the test set (without actually looking at it, to avoid any advantage over other participants in the task), and we organized the evaluation of Czech, relying on a large pool of our Czech colleagues and other annotators.

We were involved in five such campaigns so far (Bojar *et al.*, 2013b, 2014, 2015, 2016a, 2017a).

Metrics Tasks build upon the large pool of manual translation quality judgements collected in the evaluation of News Translation Task and test the performance of automatic metrics against human scoring. Since 2008, two

levels of evaluation are considered: "system-level" (metrics have to predict the quality of a set of sentences) and "segment-level" (metrics have to predict the quality of every sentence).

We were co-organizing five metrics tasks (Macháček and Bojar, 2013; Macháček and Bojar, 2014; Stanojević *et al.*, 2015b; Bojar *et al.*, 2016d, 2017b) and Table 7.1 provides an overview of the full history of the task.

In 2016, we trialled the use of direct assessment as the golden truth in the metrics task and in 2017, it became the official method of news task evaluation, so we switched to it as well. For some language pairs, the direct assessment method did not allow to collect sufficient number of manual judgements and we had to resort to the older style of comparison, as indicated by the symbols ◖ and ◗.

It used to be the case in the past, that successful metrics from one year were never submitted again in the subsequent editions of the task simply because their authors got interested in other topics. To at least partially avoid this loss, we introduced a set of baseline metrics and regularly include them in the task. Accumulating the results over the years (i.e., a varied set of language pairs and evaluated MT systems), we can draw more stable conclusions about the overall performance of these metrics. A first such summary was presented in Bojar *et al.* (2016c).

Tuning Tasks were devoted to the model optimization as mentioned in Section 2: a fixed set of model components for a fixed MT system was provided and task participants had to find the best weight settings. The translations using these settings (run by the task organizers) were then evaluated manually among the News task submissions. The point of the tuning tasks was to assess the applicability of various MT metrics in model optimization and the performance of various model optimization techniques themselves.

After two rounds of the tuning task (Stanojević *et al.*, 2015a; Jawaid *et al.*, 2016), we concluded that the variance among the different submissions in large-data setting (Tuning Task 2016) is small. The results have nevertheless clearly indicated that there was some progress in the optimization algorithms, KBMIRA (Cherry and Foster, 2012) outperforming the prevalent MERT (Och, 2003), but *not* in metrics when used for model optimization: BLEU (Papineni *et al.*, 2002) was still the method that led to the best-performing systems in terms of final manual evaluation.

Neural MT Training Task (Bojar *et al.*, 2017c) is a new type of task we proposed in respose to the shift to neural MT. The performance of neural MT models is affected by several more or less independent aspects: (1) the model structure, (2) the available training data and their pre-processing and (3) the technique used to train the model. In the NMT training task, we fixed (1) and (2), providing task participants with a pre-defined model

in the Neural Monkey toolkit (Helcl and Libovický, 2017), pre-processed training data and some suggestions what would be interesting to evaluate. As with the tuning tasks, participants did not run the translation themselves, they only provided the trained models. We applied the models to the WMT17 news test set and included these outputs in manual evaluation of WMT17.

The results indicate that statistically-significant differences in translation quality can be obtained by different training techniques, and the more successful submissions shared one particular property: they adapted the training corpus to the news domain by subsampling it or by promoting such sentence pairs. Domain adaptation is thus a critical step in the training of neural MT.

Further long-term observations of the news translation task (esp. its manual evaluation) and the metrics task (a summary of the best performing metrics across the years) are provided in Bojar *et al.* (2016c).[4]

[4](Bojar *et al.*, 2016c) is reprinted as Appendix A.12 on page 193.

Chapter 8

Summary

This habilitation thesis summarizes the contributions of Ondřej Bojar in the area of machine translation and machine translation evaluation focused on the translation into morphologically rich languages, mainly from English into Czech.

As documented in the attached publications, the author has:

- created a large automatically-annotated corpus CzEng, allowing a wide audience of researchers to experiment with English-Czech translation and allowing Czech to become a frequent example language in MT research,
- exploited explicit morphological information to improve translation quality into Czech, using several different techniques and different settings: word forms known but less frequent in parallel data, word forms not available in parallel data but covered in monolingual data and word forms not available even in the monolingual data,
- experimented with incorporating deep syntactic processing into machine translation systems, proposing a technique that defined the state of the art for news translation from English to Czech in years 2013–2015,
- contributed to techniques of MT evaluation by analyzing the space of possible translations, difficulties of MT evaluation and issues of the most commonly used MT evaluation method,
- supported the MT community by co-organizing shared task and also significantly contributing to the practices in translation task evaluation.

The original scientific papers detailing these contributions are reproduced in Appendix A, pages 63–200.

Bibliography

Eleftherios Avramidis and Philipp Koehn. Enriching morphologically poor languages for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 763–770, Columbus, Ohio, June 2008. Association for Computational Linguistics. Cited on page 20

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. Cited on page 12

Eduard Bejček. *Automatické propojování lexikografických zdrojů a korpusových dat*. PhD thesis, Charles University, Prague, 2015. Cited on page 16

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3:1137–1155, March 2003. Cited on page 19

E. M. Bennett, R. Alpert, and A. C. Goldstein. Communications through limited questioning. *Public Opinion Quarterly*, 18(3):303–308, 1954. Cited on page 47

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus Phrase-Based Machine Translation Quality: a Case Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 257–267, Austin, Texas, November 2016. Association for Computational Linguistics. Cited on page 44

Luisa Bentivogli, Marcello Federico, Sebastian Stüker, Mauro Cettolo, and Jan Niehues. The IWSLT Evaluation Campaign: Challenges, Achievements, Future Directions. In Ondřej Bojar, Aljoscha Burchardt, Christian Dugast, Marcello Federico, Josef Genabith, Barry Haddow, Jan Hajič, Kim Harris, Philipp Koehn, Matteo Negri, Martin Popel, Georg Rehm, Lucia Specia, Marco Turchi, and Hans Uszkoreit, editors, *Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem*, pages 14–19, Portorož, Slovenia, 2016. [http://www.cracking-the-language-barrier.eu/], LREC. Cited on page 45

Adam L. Berger, Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, John R. Gillett, John D. Lafferty, Robert L. Mercer, Harry Printz, and Luboš Ureš. The candide system for machine translation. In *Proceedings of the workshop on Human Language Technology*, HLT '94, pages 157–162, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics. Cited on page 7

Nicola Bertoldi and Marcello Federico. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece, March 2009. Association for Computational Linguistics. Cited on page 26

Alexandra Birch, Miles Osborne, and Philipp Koehn. CCG Supertags in Factored Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 9–16, Prague, Czech Republic, June 2007. Association for Computational Linguistics. Cited on page 20

Alexandra Birch, Omri Abend, Ondřej Bojar, and Barry Haddow. HUME: Human UCCA-Based Evaluation of Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1264–1274, Austin, Texas, November 2016. Association for Computational Linguistics. peer-reviewed. Cited on page 44

Ondřej Bojar and Jan Hajič. Phrase-Based and Deep Syntactic English-to-Czech Statistical Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 143–146, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Cited on page 10, 29, 30

Ondřej Bojar and Kamil Kos. 2010 Failures in English-Czech Phrase-Based MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 60–66, Uppsala, Sweden, July 2010. Association for Computational Linguistics. Cited on page 24

Ondřej Bojar and Aleš Tamchyna. Forms Wanted: Training SMT on Monolingual Data. Abstract at Machine Translation and Morphologically-Rich Languages. Research Workshop of the Israel Science Foundation University of Haifa, Israel, January 2011. Cited on page 25

Ondřej Bojar and Aleš Tamchyna. Improving Translation Model by Monolingual Data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. Cited on page 25, 26, 27

Ondřej Bojar and Aleš Tamchyna. The Design of Eman, an Experiment Manager. *The Prague Bulletin of Mathematical Linguistics*, 99:39–58, 2013. Cited on page 21

Ondřej Bojar and Aleš Tamchyna. CUNI in WMT15: Chimera Strikes Again. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 79–83, Lisboa, Portugal, September 2015. Association for Computational Linguistics. Cited on page 31

Ondřej Bojar and Miroslav Týnovský. Evaluation of Tree Transfer System. Project Euromatrix - Deliverable 3.4, ÚFAL, Charles University, March 2009. Cited on page 29, 30

Ondřej Bojar and Dekai Wu. Towards a Predicate-Argument Evaluation for MT. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 30–38, Jeju, Republic of Korea, July 2012. Association for Computational Linguistics. Cited on page 44

Ondřej Bojar and Zdeněk Žabokrtský. CzEng: Czech-English Parallel Corpus, Release version 0.5. *Prague Bulletin of Mathematical Linguistics*, 86:59–62, 2006. Cited on page 16

Ondřej Bojar and Zdeněk Žabokrtský. CzEng 0.9: Large Parallel Treebank with Rich Annotation. *Prague Bulletin of Mathematical Linguistics*, 92:63–83, 2009. Cited on page 15, 16

Ondřej Bojar, Miroslav Janíček, Zdeněk Žabokrtský, Pavel Češka, and Peter Beňa. CzEng 0.7: Parallel Corpus with Community-Supplied Translations. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. ELRA. Cited on page 16

Ondřej Bojar, Kamil Kos, and David Mareček. Tackling Sparse Data Issue in Machine Translation Evaluation. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 86–91, Uppsala, Sweden, July 2010. Association for Computational Linguistics. Cited on page 42, 43, 44

Ondřej Bojar, Adam Liška, and Zdeněk Žabokrtský. Evaluating Utility of Data Sources in a Large Parallel Czech-English Corpus CzEng 0.9. In *Proceedings of the Seventh International Language Resources and Evaluation (LREC'10)*, pages 447–452, Valletta, Malta, May 2010. ELRA, European Language Resources Association. Cited on page 16

Ondřej Bojar, Miloš Ercegovčević, Martin Popel, and Omar Zaidan. A Grain of Salt for the WMT Manual Evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. Cited on page 46, 47

Ondřej Bojar, Bushra Jawaid, and Amir Kamran. Probes in a Taxonomy of Factored Phrase-Based Models. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 253–260, Montréal, Canada, June 2012. Association for Computational Linguistics. Cited on page 22, 25

Ondřej Bojar, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. The Joy of Parallelism with CzEng 1.0. In *Proceedings of the Eighth International Language Resources and Evaluation Conference (LREC'12)*, pages 3921–3928, Istanbul, Turkey, May 2012. ELRA,

European Language Resources Association. Cited on page 15, 16

Ondřej Bojar, Matouš Macháček, Aleš Tamchyna, and Daniel Zeman. Scratching the Surface of Possible Translations. In *Proc. of TSD 2013*, Lecture Notes in Artificial Intelligence, Berlin / Heidelberg, 2013. Západočeská univerzita v Plzni, Springer Verlag. Cited on page 38, 39

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. Cited on page 40, 48

Ondřej Bojar, Rudolf Rosa, and Aleš Tamchyna. Chimera – Three Heads for English-to-Czech Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 92–98, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. Cited on page 31

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, MD, USA, 2014. Association for Computational Linguistics. Cited on page 48

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisboa, Portugal, September 2015. Association for Computational Linguistics. Cited on page 48

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 Conference on Machine Translation (WMT16). In Ondřej Bojar et al., editors, *Proceedings of the First Conference on Machine Translation (WMT). Volume 2: Shared Task Papers*, volume 2, pages 131–198, Stroudsburg, PA, USA, 2016. Association for Computational Linguistics, Association for Computational Linguistics. Cited on page 44, 48

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, editors, *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pages 231–238, Cham / Heidelberg / New York / Dordrecht / London, 2016. Masaryk University, Springer International Publishing. Cited on page 16

Ondřej Bojar, Christian Federmann, Barry Haddow, Philipp Koehn, Matt Post, and Lucia Specia. Ten Years of WMT Evaluation Campaigns: Lessons Learnt. In Ondřej Bojar, Aljoscha Burchardt, Christian Dugast, Marcello Federico, Josef Genabith, Barry Haddow, Jan Hajič, Kim Harris, Philipp Koehn, Matteo Negri, Martin Popel, Georg Rehm, Lucia Specia, Marco Turchi, and Hans Uszkoreit, editors, *Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem*, pages 27–34, Portorož, Slovenia, 2016. [http://www.cracking-the-language-barrier.eu/], LREC. Cited on page 45, 49, 50

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. Results of the WMT16 Metrics Shared Task. In Ondřej Bojar and et al ., editors, *Proceedings of the First Conference on Machine Translation (WMT). Volume 2: Shared Task Papers*, volume 2, pages 199–231, Stroudsburg, PA, USA, 2016. Association for Computational Linguistics, Association for Computational Linguistics. Cited on page 49

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. Cited on page 45, 47, 48

Ondřej Bojar, Yvette Graham, and Amir Kamran. Results of the WMT17 Metrics Shared Task. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. Cited on page 49

Ondřej Bojar, Jindřich Helcl, Tom Kocmi, Jindřich Libovický, and Tomáš Musil. Results of the WMT17 Neural MT Training Task. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. Cited on page 49

Ondřej Bojar, Tom Kocmi, David Mareček, Roman Sudarikov, and Dusan Varis. CUNI Submission in WMT17: Chimera Goes Neural. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. Cited on page 31

Ondřej Bojar. English-to-Czech Factored Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 232–239, Prague, Czech Republic, June 2007. Association for Computational Linguistics. Cited on page 20, 21

Ondřej Bojar. *Exploiting Linguistic Data in Machine Translation.* PhD thesis, ÚFAL, MFF UK, Prague, Czech Republic, October 2008. Cited on page 29

Ondřej Bojar. Analyzing Error Types in English-Czech Machine Translation. *Prague Bulletin of Mathematical Linguistics*, 95:63–76, March 2011. Cited on page 41

Ondřej Bojar. *Čeština a strojový překlad (Czech Language and Machine Translation)*, volume 11 of *Studies in Computational and Theoretical Linguistics.* ÚFAL, Praha, Czech Republic, 2012. Cited on page 11, 18, 19, 22, 39

Ondřej Bojar. *Machine translation*, chapter 13, pages 323–347. Oxford Handbooks in Linguistics. Oxford University Press, Oxford, UK, 2015. Cited on page 10

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, 2007. Cited on page 26

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Comput. Linguist.*, 16(2):79–85, June 1990. Cited on page 7, 8, 9

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311, June 1993. Cited on page 7, 9

Franck Burlot, Elena Knyazeva, Thomas Lavergne, and François Yvon. Two-Step MT: Predicting Target Morphology. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT'16, Seattle, USA, 2016. Cited on page 25

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. (Meta-) Evaluation of Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic, June 2007. Association for Computational Linguistics. Cited on page 47

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden, July 2010. Association for Computational Linguistics. Revised August 2010. Cited on page 46

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. Findings of the

2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. Cited on page 44

Sheila Castilho, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley, and Andy Way. Is Neural Machine Translation the New State of the Art? *The Prague Bulletin of Mathematical Linguistics*, 108:109, Jan 2017. Cited on page 44

Sheila Castilho, Joss Moorkens, Federico Gaspari, Rico Sennrich, Vilelmini Sosoni, Yota Georgakopoulou, Pintu Lohar, Andy Way, Antonio Valerio Miceli Barone, and Maria Gialama. A Comparative Quality Evaluation of PBSMT and NMT using Professional Translators. In *Proceedings of Machine Translation Summit XVI*, Nagoya, Japan, 2017. Cited on page 44

Stanley F. Chen and Joshua Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics. Cited on page 20

Colin Cherry and George Foster. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 427–436, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. Cited on page 49

David Chiang. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 263–270, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. Cited on page 29

David Chiang. Learning to translate with source and target syntax. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1443–1452, Uppsala, Sweden, July 2010. Association for Computational Linguistics. Cited on page 10, 29

Bonnie J. Dorr. Machine translation divergences: a formal description and proposed solution. *Comput. Linguist.*, 20(4):597–633, 1994. Cited on page 30

Markus Dreyer and Daniel Marcu. HyTER: Meaning-Equivalent Semantics for Translation Evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 162–171, Montréal, Canada, June 2012. Association for Computational Linguistics. Cited on page 39

Ondřej Dušek, Zdeněk Žabokrtský, Martin Popel, Martin Majliš, Michal Novák, and David Mareček. Formemes in english-czech deep syntactic MT. In *Proceedings of NAACL 2012 Workshop on Machine Translation*, pages 267–274, Montréal, Canada, 2012. Association for Computational Linguistics. Cited on page 31

Ondřej Dušek, Jan Hajic, and Zdenka Uresova. Verbal valency frame detection and selection in czech and english. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 6–11, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics. Cited on page 16

Jason Eisner. Learning Non-Isomorphic Tree Mappings for Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL), Companion Volume*, pages 205–208, Sapporo, July 2003. Cited on page 30

George Foster, Roland Kuhn, and Howard Johnson. Phrasetable Smoothing for Statistical Machine Translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 53–61, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. Cited on page 20

Alexander Fraser. Experiments in morphosyntactic processing for translating to and from german. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 115–119, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. Cited on page 24

Eva Fučíková, Jan Hajič, and Zdeňka Urešová. Enriching a Valency Lexicon by Deverbative Nouns. In Eva Hajičová and Igor Boguslavsky, editors, *Proceedings of the Workshop on Grammar and Lexicon: Interactions and Interfaces (GramLex)*, pages 71–80, -Osaka, Japan, 2016. ICCL, The COLING 2016 Organizing Committee. Cited on page 16

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, FirstView:1–28, 1 2016. Cited on page 47

Jan Hajič. *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Nakladatelství Karolinum, Prague, 2004. Cited on page 23

Jindřich Helcl and Jindřich Libovický. Neural Monkey: An open-source tool for sequence learning. *The Prague Bulletin of Mathematical Linguistics*, 107:5–17, 2017. Cited on page 50

Mark Hopkins and Jonathan May. Models of translation competitions. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1416–1424, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. Cited on page 47

Matthias Huck, Aleš Tamchyna, Ondřej Bojar, and Alexander Fraser. Producing Unseen Morphological Variants in Statistical Machine Translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 369–375, Stroudsburg, PA, USA, 2017. Universitat Politécnica de València, Association for Computational Linguistics. Cited on page 12, 17, 28

Bushra Jawaid and Ondřej Bojar. Two-step machine translation with lattices. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, and Joseph Mariani, editors, *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 682–686, Reykjavík, Iceland, 2014. European Language Resources Association. Cited on page 25

Bushra Jawaid, Amir Kamran, Miloš Stanojević, and Ondřej Bojar. Results of the WMT16 Tuning Shared Task. In Ondřej Bojar and et al ., editors, *Proceedings of the First Conference on Machine Translation (WMT). Volume 2: Shared Task Papers*, volume 2, pages 232–238, Stroudsburg, PA, USA, 2016. Association for Computational Linguistics, Association for Computational Linguistics. Cited on page 49

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. Montreal Neural Machine Translation Systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 130–136, Lisboa, Portugal, September 2015. Association for Computational Linguistics. Cited on page 34

Kevin Knight. Decoding complexity in word-replacement translation models. *Comput. Linguist.*, 25(4):607–615, 1999. Cited on page 10

Tom Kocmi and Ondřej Bojar. SubGram: Extending Skip-gram Word Representation with Substrings. In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, editors, *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pages 182–189, Cham / Heidelberg / New York / Dordrecht / London, 2016. Masaryk University, Springer International Publishing. Cited on page 16

Philipp Koehn and Hieu Hoang. Factored Translation Models. In *Proc. of EMNLP*, 2007. Cited on page 20

Philipp Koehn and Christof Monz. Manual and automatic evaluation of machine translation between european languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121, New York City, June 2006. Association for Computational Linguistics. Cited on page 45

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *HLT/NAACL*, 2003. Cited on page 9, 18

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej

Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. Cited on page 18

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009. Cited on page 11, 19

Philipp Koehn. Simulating Human Judgment in Machine Translation Evaluation Campaigns. In *Proc. of IWSLT*, pages 179–184, 2012. Cited on page 47

Nobo Komagata. Chance agreement and significance of the kappa statistic. http://nobo.komagata.net/pub/Komagata02-Kappa.pdf (as of Aug 2017), 2002. Cited on page 47

Jakub Kúdela, Irena Holubová, and Ondřej Bojar. Extracting parallel paragraphs from common crawl. *The Prague Bulletin of Mathematical Linguistics*, (107):36–59, 2017. Cited on page 16

J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174, 1977. Cited on page 47

Chi-kiu Lo and Dekai Wu. Meant: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 220–229, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. Cited on page 44

Adam Lopez. Putting Human Assessments of Machine Translation Systems in Order. In *Proc. of the Seventh Workshop on Statistical Machine Translation*, pages 1–9, Montréal, Canada, 2012. Cited on page 47

Matouš Macháček and Ondřej Bojar. Results of the WMT14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, Baltimore, MD, USA, 2014. Association for Computational Linguistics. Cited on page 49

Matouš Macháček and Ondřej Bojar. Approximating a Deep-Syntactic Metric for MT Evaluation and Tuning. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 92–98, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. Cited on page 44

Matouš Macháček and Ondřej Bojar. Results of the WMT13 Metrics Shared Task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. Cited on page 49

Jiří Maršík and Ondřej Bojar. TrTok: A Fast and Trainable Tokenizer for Natural Languages. *Prague Bulletin of Mathematical Linguistics*, 98:75–85, September 2012. Cited on page 16

Vendula Michlíková. Výslovnostní rysy češtiny - dialektová analýza. Bachelor Thesis. Charles University, 2013. Cited on page 16

Einat Minkov, Kristina Toutanova, and Hisami Suzuki. Generating Complex Morphology for Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 128–135, Prague, Czech Republic, June 2007. Association for Computational Linguistics. Cited on page 24

Jan Niehues and Alex Waibel. Domain adaptation in statistical machine translation using factored translation models. In *EAMT*, 2010. Cited on page 20

Michal Novák, Anna Nedoluzhko, and Zdeněk Žabokrtský. Translation of "It" in a Deep Syntax Framework. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 51–59, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. Cited on page 16

Franz Josef Och and Hermann Ney. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *ACL*, pages 295–302, 2002. Cited on page 8

Franz Josef Och. Minimum Error Rate Training in Statistical Machine Translation. In *Proc.*

of the Association for Computational Linguistics, Sapporo, Japan, July 6-7 2003. Cited on page 49

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, 2002. Cited on page 40, 49

Martin Popel and Zdeněk Žabokrtský. TectoMT: Modular NLP framework. In Hrafn Loftsson, Eirikur Rögnvaldsson, and Sigrun Helgadottir, editors, *Lecture Notes in Artificial Intelligence, Proceedings of the 7th International Conference on Advances in Natural Language Processing (IceTAL 2010)*, volume 6233 of *Lecture Notes in Computer Science*, pages 293–304, Berlin / Heidelberg, 2010. Iceland Centre for Language Technology (ICLT), Springer. Cited on page 12, 15, 30

Rudolf Rosa, David Mareček, and Ondřej Dušek. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 362–368, Montréal, Canada, June 2012. Association for Computational Linguistics. Cited on page 31

Lane Schwartz. Multi-Source Translation Methods. In *Proc. of AMTA*, 2008. Cited on page 33

Holger Schwenk, Anthony Rousseau, and Mohammed Attik. Large, Pruned or Continuous Space Language Models on a GPU for Statistical Machine Translation. In *Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, WLM '12, pages 11–19, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. Cited on page 19

Holger Schwenk. Continuous Space Language Models. *Comput. Speech Lang.*, 21(3):492–518, July 2007. Cited on page 19

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. Cited on page 26

Magda Ševčíková, Zdeněk Žabokrtský, Jonáš Vidra, and Milan Straka. Lexikální síť derinet: elektronický zdroj pro výzkum derivace v češtině. *Časopis pro moderní filologii*, 98(1):62–76, 2016. Cited on page 16

Petr Sgall, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects*. Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands, 1986. Cited on page 11

Jana Šindlerová, Zdeňka Urešová, and Eva Fučíková. Resources in conflict: A bilingual valency lexicon vs. a bilingual treebank vs. a linguistic theory. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, and Joseph Mariani, editors, *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, pages 2490–2494, Reykjavík, Iceland, 2014. European Language Resources Association. Cited on page 30

Miloš Stanojević, Amir Kamran, and Ondřej Bojar. Results of the WMT15 Tuning Shared Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 274–281, Lisboa, Portugal, September 2015. Association for Computational Linguistics. Cited on page 49

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. Results of the WMT15 Metrics Shared Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273, Lisboa, Portugal, September 2015. Association for Computational Linguistics. Cited on page 49

Jana Straková, Milan Straka, and Jan Hajič. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18,

Baltimore, Maryland, June 2014. Association for Computational Linguistics. Cited on page 28

Roman Sudarikov, David Mareček, Tom Kocmi, Dušan Variš, and Ondřej Bojar. CUNI Submission in WMT17: Chimera Goes Neural. In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. Cited on page 34

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. pages 3104–3112, 2014. Cited on page 12

Aleš Tamchyna and Ondřej Bojar. No Free Lunch in Factored Phrase-Based Machine Translation. In *Proc. of CICLing 2013*, volume 7817 of *LNCS*, pages 210–223, Samos, Greece, 2013. Springer-Verlag. Cited on page 22

Aleš Tamchyna and Ondřej Bojar. What a Transfer-Based System Brings to the Combination with PBMT. In Bogdan Babych, Kurt Eberle, Patrik Lambert, Reinhard Rapp, Rafael Banchs, and Marta Costa-Jussà, editors, *Proceedings of the Fourth Workshop on Hybrid Approaches to Translation (HyTra)*, pages 11–20, Stroudsburg, PA, USA, 2015. Association for Computational Linguistics, Association for Computational Linguistics. Cited on page 31, 32, 33, 34

Aleš Tamchyna, Martin Popel, Rudolf Rosa, and Ondřej Bojar. CUNI in WMT14: Chimera still awaits bellerophon. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 195–200, Baltimore, MD, USA, 2014. Association for Computational Linguistics. Cited on page 31

Aleš Tamchyna, Alexander Fraser, Ondřej Bojar, and Marcin Junczys-Dowmunt. Target-Side Context for Discriminative Models in Statistical Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1704–1714. Association for Computational Linguistics, Association for Computational Linguistics, 2016. Cited on page 12, 28

Aleš Tamchyna, Roman Sudarikov, Ondřej Bojar, and Alexander Fraser. CUNI-LMU Submissions in WMT2016: Chimera Constrained and Beaten. In *Proceedings of the First Conference on Machine Translation*, pages 385–390, Berlin, Germany, August 2016. Association for Computational Linguistics. Cited on page 31

Aleš Tamchyna. *Lexical and Morphological Choices in Machine Translation*. PhD thesis, ÚFAL, MFF UK, Prague, Czech Republic, June 2017. Cited on page 12, 28

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. Applying morphology generation models to machine translation. In *Proceedings of ACL-08: HLT*, pages 514–522, Columbus, Ohio, June 2008. Association for Computational Linguistics. Cited on page 24

Kateřina Veselovská. *On the Linguistic Structure of Emotional Meaning in Czech*. PhD thesis, Charles University, Prague, 2015. Cited on page 16

John Wieting, Jonathan Mallinson, and Kevin Gimpel. Learning Paraphrastic Sentence Embeddings from Back-Translated Bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Copenhagen, Denmark, 2017. Cited on page 16

Kenji Yamada and Kevin Knight. A Syntax-Based Statistical Translation Model. In *ACL '01: Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, pages 523–530, Morristown, NJ, USA, 2001. Association for Computational Linguistics. Cited on page 10

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. TectoMT: Highly Modular Hybrid MT System with Tectogrammatics Used as Transfer Layer. In *Proc. of the ACL Workshop on Statistical Machine Translation*, pages 167–170, Columbus, Ohio, USA, 2008. Cited on page 16, 31

Andreas Zollmann and Ashish Venugopal. Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June 2006. Association for Computational Linguistics. Cited on page 10

Appendix A

Reprints of Key Papers of the Thesis

This appendix contains the full texts of the key articles and papers published in the years 2007–2016 and supporting the research summary outlined in the main content of the thesis.

1. **Ondřej Bojar**. *Machine translation*, chapter 13, pages 323–347. Oxford Handbooks in Linguistics. Oxford University Press, Oxford, UK, 2015.

   Citations: 2 (excluding self-citations)

2. **Ondřej Bojar**, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. The Joy of Parallelism with CzEng 1.0. In *Proceedings of the Eighth International Language Resources and Evaluation Conference (LREC'12)*, pages 3921–3928, Istanbul, Turkey, May 2012. ELRA, European Language Resources Association.

   Citations: 35 (excluding self-citations)
   Estimated contribution of the applicant: 40%. Ondřej Bojar organized the team, assembled the corpus and carried out the automatic processing using data and tools provided by co-authors.

3. **Ondřej Bojar**. English-to-Czech Factored Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 232–239, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

   Citations: 21 (excluding self-citations)

4. **Ondřej Bojar** and Aleš Tamchyna. The Design of Eman, an Experiment Manager. *The Prague Bulletin of Mathematical Linguistics*, 99:39–58, 2013, doi:10.2478/pralin-2013-0003.

   Citations: 4 (excluding self-citations)
   Estimated contribution of the applicant: 90%. Ondřej Bojar is the main author of Eman and also provided most of the paper text.

5. **Ondřej Bojar** and Kamil Kos. 2010 Failures in English-Czech Phrase-Based MT. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 60–66, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

   Citations: 7 (excluding self-citations)

6. **Ondřej Bojar** and Aleš Tamchyna. Improving Translation Model by Monolingual Data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

   Citations: 4 (excluding self-citations)

7. **Ondřej Bojar**, Rudolf Rosa, and Aleš Tamchyna. Chimera – Three Heads for English-to-Czech Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 92–98, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

   Citations: 7 (excluding self-citations)

8. **Ondřej Bojar**. Analyzing Error Types in English-Czech Machine Translation. *The Prague Bulletin of Mathematical Linguistics*, 95:63, 2011, doi:10.2478/v10108-011-0005-2.

   Citations: 10 (excluding self-citations)

9. **Ondřej Bojar**, Matouš Macháček, Aleš Tamchyna, and Daniel Zeman. *Scratching the Surface of Possible Translations*, pages 465–474. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, doi:10.1007/978-3-642-40585-3_59.

   Citations: 5 (excluding self-citations)

10. **Ondřej Bojar**, Kamil Kos, and David Mareček. Tackling Sparse Data Issue in Machine Translation Evaluation. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 86–91, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

    Citations: 6 (excluding self-citations)

11. **Ondřej Bojar**, Miloš Ercegovčević, Martin Popel, and Omar Zaidan. A Grain of Salt for the WMT Manual Evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

    Citations: 27 (excluding self-citations)

12. **Ondřej Bojar**, Christian Federmann, Barry Haddow, Philipp Koehn, Matt Post, and Lucia Specia. Ten Years of WMT Evaluation Campaigns:

Lessons Learnt. In Ondřej Bojar, Aljoscha Burchardt, Christian Dugast, Marcello Federico, Josef Genabith, Barry Haddow, Jan Hajič, Kim Harris, Philipp Koehn, Matteo Negri, Martin Popel, Georg Rehm, Lucia Specia, Marco Turchi, and Hans Uszkoreit, editors, *Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem*, pages 27–34, Portorož, Slovenia, 2016. [http://www.cracking-the-language-barrier.eu/], LREC.

CHAPTER 14

MACHINE TRANSLATION

ONDŘEJ BOJAR

14.1 INTRODUCTION

THE goal of machine translation (MT) is easy to define but hard or impossible to achieve in its entirety: to implement a computer program that takes some text in one natural language (the source language) and produces the equivalent in another natural language (the target language). Although MT is a quintessentially linguistic problem, most current MT systems are fuelled by statistics rather than linguistic rules. Since the statistics are based on what has been observed in a corpus, morphological productivity leads to a problem of data sparsity in that many of the word forms will be unobserved or insufficiently observed even in a very large corpus. The problem is compounded by divergences between source and target languages. Divergences arise when one language has morphological exponence of concepts that are not expressed explicitly in the other language. Number, for example, is difficult to translate from Chinese to English because it is not explicitly marked in Chinese. Another type of divergence arises when meanings are expressed by bound morphemes in one language and by free words in another.

Hopes for an automatic translation system have been around since the era of John von Neumann and Alan Turing, see Hutchins (2005) or the very optimistic IBM press release in 1954.[1] A notable drop in MT research activity followed the ALPAC report (ALPAC 1966; Hutchins 2003), which raised scepticism about the possibility of automatic translation.

Since the ALPAC report, the field has recovered its reputation mainly by addressing two shortcomings. First, it was necessary to establish reasonable expectations about MT output and its uses. Fully automatic translation might still be of low quality, but it might be useful for gisting large amounts of material. Higher quality might be required

[1] <http://www-03.ibm.com/ibm/history/exhibits/701/701_translator.html>.

for the dissemination of technical information such as equipment repair manuals, but this could be achieved if the semantic domain was limited (Reuther 2003; Muegge 2007) or if human post-editors were involved in the process (Koehn and Haddow 2009; Plitt and Masselot 2010; Federico *et al.* 2012). Secondly, if MT were to be commercially viable and fundable as a research area, it would be necessary to measure progress. Having humans judge MT quality proved to be too slow to support rapid research cycles of evaluation and improvement of MT systems. A giant leap forward was facilitated by the introduction and wide acceptance of automatic metrics of MT quality (Papineni *et al.* 2002). Despite all the shortcomings of such evaluation methods, as we will see in Section 14.4.5, including a bias towards linguistically superficial statistical methods, the automatic evaluation methods have facilitated large-scale research in MT.

The Holy Grail of MT is a system that is fully automatic (without human intervention), domain independent (able to translate a variety of text and speech genres), and high quality. This goal remains elusive. However, many research projects nowadays focus a range of applications that already have satisfied users:

- free MT services on the Internet: although the output is usually not perfect, it may be useful for understanding or conveying the gist of a text;
- integration of MT into platforms for computer-assisted translation (CAT), which may include confidence estimation to automatically identify near-perfect segments or mark problematic parts of the output;
- integration of MT into applications such as cross-lingual information retrieval.

14.2 Machine Translation and Rich Morphology

14.2.1 History of MT with Rich Morphology

Lopez (2008) mentions that a large proportion of MT research deals with translation into English, with the side effect of obscuring deficiencies of the current approaches with respect to morphological complexity. To verify this claim, we analysed papers available in the Association for Computational Linguistics (ACL) Anthology.[2] Subject to optical character recognition (OCR) and other conversion errors, our collection as of October 2013 contains 27,015 documents. We selected only those published in 1985 or later that have between 2,000 and 9,000 words to avoid short abstracts or whole workshop proceedings, leaving us with 21,291 papers. About one-quarter of all of these

[2] <http://aclweb.org/anthology/>.

papers mention MT once or more and about 3,500 papers mention MT twice or more with a clear increase in counts since 2007.

When searching for language names in the 3,500 papers presumably on MT, we see that 85 per cent of them mention English and about one-quarter of them contains the phrase *(in)to English*. The second most frequent language is morphologically poor Chinese (mentioned in about a third of the papers). The remaining most studied languages, each appearing in a few hundred MT papers, have differing degrees of morphological complexity (French, German, Czech, Japanese, Spanish, Arabic, Korean, Italian, Dutch, Hindi, and Russian), representing six genetic or typological groups, four of which are Indo-European (Romance, Germanic, Slavic, Indo-Iranian, Japanese/Korean, and Semitic). Many other language families with richer morphology have little coverage, such as Turkic or Uralic (e.g. Turkish is mentioned in about 100 papers, Hungarian in 70 papers and Finnish in 60 papers), or are almost totally absent such as Bantu, Austronesian, Dravidian, and Eskimo-Aleut (used in two parliaments: Canada and Greenland).

14.2.2 Morphological Richness

In the field of MT, morphological richness can be measured by examining the number of different **tokens** in a corpus. We will use the term **token** to denote an occurrence of a word or punctuation mark in a text. A type of program called **tokenizer** separates a text into tokens based on various criteria (depending on whether the language uses spaces between words). Punctuation marks also get separated from adjacent words or other punctuation marks, forming tokens on their own. The term **vocabulary size** refers to the number of distinct token types (referred to as **word forms**) seen in a given text.[3] The word forms *cat* and *cats* are two different vocabulary items for us. For our purposes, we define **morphological richness** only in contrast to another language as the relative vocabulary size compared to the same text in the other language. We will also use the broad term **morphologically rich language** (MRL) to denote a language richer than English.

Consider the book *1984* by George Orwell in the original English and translated to ten other languages. Table 14.1 provides statistics of this book as morphologically annotated in the project MULTEXT-East (Erjavec 2010). Obviously, all the variants of the book tell the same story and the almost identical number of paragraphs and sentences validate the statistics. In contrast, the vocabulary sizes greatly vary across languages, English being one of the least morphologically rich languages and all other ones having 1.82 ± 0.28 times bigger vocabulary on average. The differences in vocabulary size become much less pronounced if we move from distinct word forms to lemmas

[3] Perhaps somewhat unintuitively, the term 'word form' usually does include punctuation marks and punctuation marks are thus regular entries of the 'vocabulary'.

Table 14.1 MULTEXT-East statistics of the book *1984* by decreasing 'Lemmas' count

	Tokens	Vocabulary size in terms of			
	excl. punct	Word forms	Lemmas	Paragraphs	Sents
Slovak	84062	20240	10065	1359	6354
Hungarian	80708	20311	10050	1303	6768
Polish	79772	21051	9451	1401	6666
Czech	79870	19107	9114	1298	6752
Estonian	75431	17836	8724	1266	6478
Bulgarian	86020	16343	8516	1322	6682
Serbian	89829	18082	8392	1293	6677
Slovenian	90792	17861	8303	1288	6689
Romanian	101772	15195	7249	1343	6520
English	104286	**9762**	**7069**	1287	6737
Persian	95812	11308	6597	1266	6604

('base' or citation forms of words): other languages have on average 1.22±0.15 times more distinct lemmas than English.

Morphological richness may be caused by inflection or derivation. However, in MT, and especially in contemporary statistical MT, we are not aware of any issues specific to languages rich by inflection as opposed to languages rich by derivation.

What *is* relevant in MT though, are the differences in morphological richness between the two languages in question. We see four different cases:

- richer language on the source side (such as Czech→English);
- richer language on the target side (such as English→Czech);
- both sides comparably rich and similar in the system of morphological properties as well as syntactic patterns (such as Czech-Slovak), which allows translation by token;
- both sides comparably rich but more distant in the set of explicitly represented morphological features, word structure, or sentence structure (such as Czech-Turkish or German-Finnish), so that translation token by token is no longer linguistically adequate.

Birch *et al.* (2008) and later Koehn *et al.* (2009) find, on a large set of European languages, that the target-side vocabulary size is one of the most important reasons for the failure of MT systems of the phrase-based variety (see Section 14.3.3). Finnish was an extreme representative in the experiment, with vocabulary size (in terms of distinct word forms) more than five times larger than English. On the other hand, the source-side richness was not identified as critical. There are two possible explanations. First, the observation may be a side effect of automatic evaluation methods

(Section 14.4.5), which often require an exact match between the output tokens of the MT system and a human reference translation. With a larger target-side vocabulary, the match is less likely regardless of translation quality. Secondly, MT is actually harder into languages with a larger vocabulary size, because there are more possible outputs to choose from. This choice is further complicated by the fact that less frequent word forms are not adequately observed in a corpus, which negatively affects both the bilingual 'translation dictionaries' (Section 14.4.2) and the target-side language models (Section 14.4.4).

If both languages are comparably rich and match in the set of morphological properties overt on the surface (of matching words), the system can very successfully decompose translation into independent streams of information, the lexical values and the morphological properties, see also Section 14.5.1.

We are not aware of much MT research devoted to rich and morphologically non-matching pairs of languages aside from a few preliminary attempts for Arabic and French (Hasan and Ney 2008) or Italian (Cettolo *et al.* 2011) and the broad experiment of Koehn *et al.* (2009) where, for example, Hungarian–Finnish had very poor results in both directions. Certainly, language pairs excluding English are going to be the next focus of the field and interesting efforts in this respect have already started (Megyesi *et al.* 2010).

On a general note, rich morphology is bound to make the translation harder for simple combinatorics reasons. Having a sentence of n words requires us in principle to consider up to $n!$ word permutations. Modelling morphology explicitly would force us to operate on characters instead of words, considering up to all *character permutations* of sentences. For m characters in a sentence of n words, $m > n$ and thus $m! \gg n!$.

14.3 ANATOMY OF A STATISTICAL MT SYSTEM

Dorr *et al.* (1998) provide an excellent survey of approaches to machine translation (MT), starting with a list of linguistic problems MT has to handle and including a categorization of rule-based systems depending on the deepest level of linguistic analysis performed by the system (morphology, syntax, or semantics). At that time, there was a division of approaches in MT depending on whether the system consisted mainly of rules written by linguists (rule-based systems) or whether the system consisted mainly of statistics about correspondences between languages learnt from a bilingual corpus (corpus-based systems). Early corpus-based systems that focused on probabilities of translations of individual words did not have the capability to work with linguistic structures such as parse trees and predicate–argument structures.

In a more recent survey, Lopez (2008) focuses on approaches to MT that can be called 'statistical' (SMT) regardless of the depth of linguistic analysis. The ten years between the surveys saw a great shift towards data-oriented (including statistical) approaches in general. The rivalry between rule-based and corpus-based systems has

been softened in that corpus-based systems are now able to work with statistics about structures such as parse trees and developers of rule-based systems are more likely to use a large corpus to discover the necessary rules.

The design of any MT system, whether corpus-based or rule-based must designate an approach to the following questions:

- What is the smallest **unit of translation** (e.g. morpheme, word, string of words, syntactic constituent, or dependency treelet, etc.)?
- How is an input sentence **decomposed** into such units (including the selection of some top-scoring subset of all the possible decompositions)?
- How is each of the source units **translated** (including the selection of the subset of preferred translations)?
- How is the output string of words **generated** from the set of target-side units?

Different choices of the unit of translation and the design decisions for the processes of decomposition, translation of the units, etc. are associated with many different approaches to MT.

Translation units usually include morphological information but depending on the system type, the information can be implicit in exact word forms if the system does not analyse them further (e.g. in phrase-based MT, Section 14.3.3), or it can be represented formally using various attributes or node properties and thus detached from the lexical value of the word. Existing systems differ in how far the formalization goes and how easy it is for the system to access the lexical value when handling the morphological attributes or vice versa. Dealing with translation inputs that have not been observed during training is a particular problem in languages with very productive morphology.

As most MT systems are currently corpus-based and statistical, we will briefly review the components of a statistical MT system, focusing particularly on the most common type of statistical MT system: a phrase-based system. For other approaches, we just mention some interesting specifics on handling inflection, see Section 14.5.

14.3.1 Learning from Corpora

Statistical MT systems are learnt from parallel corpora. One side of the parallel corpus consists of sentences in the **source language** and the other consists of sentences in the **target language**. We can assume for now that the source and target language sentences in the corpus line up with each other roughly one-to-one. The parallel corpus is used for learning probabilities, largely by counting things such as how frequently a particular source language word and a particular target language word occur in corresponding sentences; or how frequently a particular target language word follows another particular target language word. During **training**, probabilities are learnt from the parallel corpus using complex algorithms capable of dealing with large amounts of data, and

often requiring much computing time. When the system is used for translation, the probabilities learnt in training are applied to a new sentence in order to find its most probable translation.

Regardless how large the training corpus is and how complex translation units were chosen (words, parts of words or larger units such as sequences of words, or subtrees in a syntactic representation), we cannot expect to see all possible units in the training corpus. Even if all of the source–target pairs of units are observed, they will not be observed in every possible context.

That is, when the system translates a sentence, it is very unlikely that it has seen the precisely same sentence before (unless it is translating a very restricted genre). All practical MT systems therefore include some form of **generalization** in order to be able to assess the translation equivalence of unseen sentence pairs in the same way as humans do. Breaking up sentences into smaller units is the first vital step to such generalization, but as mentioned, the system will encounter even unseen units. In MRLs, such unseen units are more likely and in order to adequately handle them, the system should generalize further.

For statistical systems, the term **smoothing** is used to refer to this generalization capacity: some function is applied to the observed occurrence counts to 'smooth out' the rough edge between a unit never seen and a unit observed once. Unobserved units need some non-zero probability.

14.3.2 MT Pipeline

The process of creating an MT system as well as applying it to unseen texts is sometimes referred to as the 'MT pipeline'. The pipeline originally comes from statistical phrase-based systems (see Section 14.3.3) and it applies well to most data-driven paradigms but at least some of the steps in the pipeline are relevant even for very different MT paradigms such as rule-based MT.

The full pipeline (Figure 14.1) includes the following steps: word alignment (i.e. finding corresponding tokens in sentence-parallel training data), extraction of translation units (i.e. automatic construction of a system-specific 'translation dictionary'), estimation of output fluency (the language model), translation of unseen text, automatic evaluation of MT output, and model optimization (also called tuning; i.e. finding the best internal settings, e.g. weights of individual components of the system).

There are four main flavours of relevant data: target-side monolingual texts for language modelling, large parallel texts for extracting translation units and their equivalents, a small held-out parallel text (development set, devset) for model optimization, and finally the input we want to have translated.

All the textual data undergo a common preprocessing such as tokenization and tagging. While the preprocessing mostly deals with just technical issues, it easily lends itself to various linguistically motivated tricks, see Section 14.5.2.

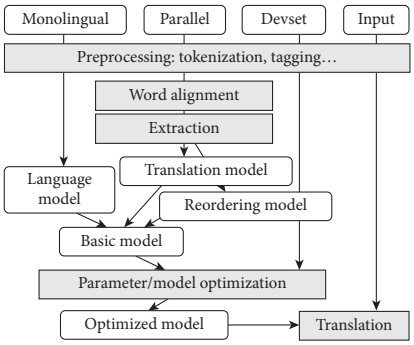


fig. 14.1 Typical components of a phrase-based and many other statistical MT systems.

14.3.3 Phrase-based MT (PBMT)

Statistical phrase-based MT (PBMT, Koehn *et al.* (2003)) superseded word-based statistical models (Brown *et al.* 1988) and has dominated the last decade of MT. One of the reasons is the applicability of the model to any language pair and part of its success also comes from the availability of the complete open-source toolkit Moses (Koehn *et al.* 2007).

The unit of translation in PBMT is a **phrase** or rather a **phrase pair**. The term refers simply to a sequence of tokens, completely disregarding any syntactic structure of the sentence.

Note that in its basic formulation, PBMT operates with word forms. The phrase pairs translating, for example, the word *cat* are thus not related to the phrase pairs for *cats* at all. In Section 14.5.2.1, we describe an attempt to relax this limitation.

Without delving into the underlying statistical models, the essence of phrase-based MT can be illustrated by Figure 14.2. The matrix with solid dots illustrates one sentence pair from a large parallel corpus. Each dot corresponds to one automatically estimated point of **word alignment**. The word alignments serve as the basis for the construction of the critical data resource of PBMT, namely the **phrase table**. The phrase table serves as the 'translation dictionary' for PBMT. It is extracted using just a simple heuristic: all phrases consistent with the word alignment are included. A phrase (pair) is consistent with the alignment if no word is linked to a word outside of the phrase.

When a new sentence is to be translated, all decompositions of the sentence into non-overlapping phrases are considered in the **search** for the best output. Each source phrase is equipped with all its translations as available in the phrase table. The output is gradually constructed from left to right by picking phrase translations for input words that have not yet been covered. Many such partial outputs are constructed and considered in order to select the best output according to a weighted score including the translation probabilities of individual phrases, the 'language model' (a model

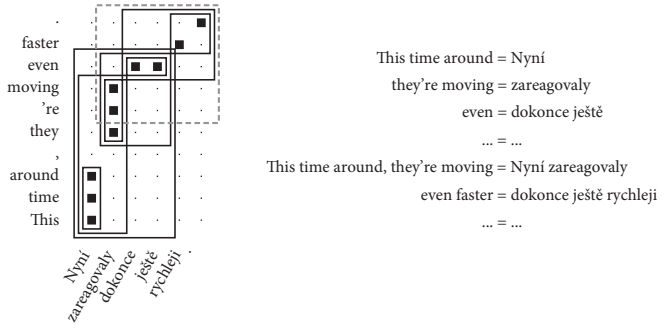


FIG. 14.2 Sample word alignment (solid dots) and some of the phrase pairs that are consistent (solid black rectangles) or inconsistent (the grey dashed rectangle) with the English sentence *This time around, they're moving even faster* and its Czech counterpart *Tentokrát zareagovaly ještě rychleji*

approximating output fluency, see Section 14.4.4), the number of phrases used, the number of words produced and several other criteria. The space of possible translations is extremely large, so less promising partial outputs are pruned (discarded) early. Pruning can lead to so-called 'search errors', that is, situations where a good translation becomes inaccessible because its beginning scored too low and was pruned away. Chang and Collins (2011) show that Moses makes search errors quite frequently but they do not seem to be detrimental to the (automatically estimated) translation quality.

Across almost all language pairs, PBMT is *relatively* successful. Humans may find the output unintelligible or full of errors but it is surprisingly difficult to outperform the PBMT baseline. The reason for this success lies in the ability of reproducing quite long sequences verbatim. In repetitive text domains, this greatly reduces the risk of mishandling known sentences.

14.4 Problems in MT Caused by Rich Morphology

Without adhering to any particular framework, we will describe problems of MT caused by rich morphology in terms of the typical 'MT pipeline' as introduced in Section 14.3.2.

14.4.1 Word Alignment

Given a corpus of parallel sentences, we need to learn the correspondence between smaller units in the source and target languages. The most apparent units to consider

are tokens, although more adequate models aligning morphemes are being studied, see the end of this section.

A very powerful statistical model for word alignment is IBM Model 1 (Brown *et al.* 1993). The model assumes that every word form in the source language (e.g. *cat* in English) has a probability distribution of correct translations in the target language (e.g. *kočka* in Czech), or possibly no counterpart: *cat* is a likely translation of *kočka*, while *dog* is not; the English article *the* frequently has no Czech counterpart. Given a sentence pair and some estimate of the word-to-word translation probabilities, IBM Model 1 finds the alignment of tokens between the source and target that maximizes the overall translation probability. The Expectation-Maximization algorithm (Dempster *et al.* 1977) is used to iteratively improve the estimated word translation probabilities and alignment probabilities starting from uniform distribution. Because this is learnt from a corpus with no human intervention, the word translation probabilities (sometimes called lexical translation probabilities) as well as word alignments frequently contain errors. Fortunately, in large datasets, true translations of words dominate the distributions.

Rich morphology, and inflection in particular, causes a '**sparse data**' problem for the task. If a word in both source and target languages does not undergo any inflection, all co-occurrences of the word and its translation contribute to the same entry in the word-to-word dictionary, making the estimate more reliable. Otherwise, for example in Czech–English translation, the model has to learn independently that the Czech *zelený* (masculine singular, ambiguous between nominative, accusative, and vocative), *zeleného* (masculine singular, ambiguous between genitive and accusative), *zelenou* (feminine singular, ambiguous between accusative and instrumental), etc., are all valid translations of the English *green*. The hardest case are language pairs with both languages rich in inflection: the word-to-word translation lexicon is polluted with up to the full Cartesian product (all combinations) of word forms, for example the German *grün*, *grüner*, *grünem*, etc., vs. the forms of the Czech *zelený*, see Figure 14.3. Note that the respective morphological features need not be relevant for both languages (e.g. there is no strong or weak inflection in Czech) and even if they are, their values can differ across languages. The problem of sparse data simply arises whenever one Czech word form of a given word corresponds to multiple German word forms and vice versa.

The main trouble is not caused by the larger storage requirements of such a polluted dictionary, but rather by the lack of enough training data to support all the combinations with observations.

Given for example two sentence pairs such as:

(1) green colorless ideas sleep = bezbarvé zelené myšlenky spí
(2) I like green pears = mám rád zelené hrušky

the model learns that *zelené* co-occurs with *green* more often than it co-occurs with *pears* (twice vs. once in this tiny corpus). However, a new form of the word makes no use of this information in a new sentence pair:

(3) I sat under a green tree = seděl jsem pod zeleným stromem

German	Czech	Number	Gender	Case	German Inflection
grüner	zelený	SG	M	NOM	strong
grünen	zelený	SG	M	ACC	strong
grünem	zelenému	SG	M	DAT	strong
grünen	zelenému	SG	M	DAT	weak
...	...	...	...	...	...

FIG. 14.3 A snippet of the Cartesian product of German and Czech morphological variants of the word *green*

Without further data, the model is completely uncertain whether *zeleným* means *green*, *tree*, *sat* or any other of the words in the English sentence.

Word alignments for rare word forms are established thanks to the assumption that all Czech words map to an English word. If all other Czech words in the sentence are aligned with a high probability, *zeleným* is likely to align with the remaining *green*. In combinatorics, this is called the 'pigeon-hole' principle: if p pigeons enter h holes in a dovecote and $p > h$, then at least one hole has to be used by at least two pigeons.

The quality of word alignments can be greatly increased by using more advanced models (Brown *et al.* 1993; Vogel *et al.* 1996; Liu *et al.* 2010; Setiawan *et al.* 2010; V. Graça *et al.* 2010). Some models have freely available implementations such as GIZA++ (Och and Ney 2003) or a parallelized version MGIZA (Gao and Vogel 2008). However, all of these models treat words as indivisible units. One model (Fraser and Marcu 2007) treats head words and function words differently, allowing only head words to link across languages. The function words are used only for modelling within each separate language. This model would give a promising approach to morphology if bound morphemes could be separated from word stems and treated as function words.

The sparse data issue can be mitigated by conflating morphological distinctions in one language that are not overt in the other language. A very rough approximation of lemmas can be sufficient (Corston-Oliver and Gamon 2004; Bojar and Prokopová 2006; Hermjakob 2009), such as the reduction of all word forms to just the first four letters. Such a 'lemmatizer' correctly equates tokens like *sleep* and *sleeps* under the label 'slee', leaving irregularities like *slept* unhandled. False positives like collapsing the English words *envious* and *environment* into a single class 'envi' do not cause much harm in the alignment task because the distributional properties of such unrelated words are different and only very few random sentences containing both *envious* and *environment* could confuse the model.

Bodrumlu *et al.* (2009) propose a promising model capable of aligning sub-word segments. The model performs well on a small English–Turkish corpus, despite the built-in limitation to 1–1 alignments. Naradowsky and Toutanova (2011) improve alignment quality using source-side linguistic information and automatically segmenting target-side words to morphemes that best match the alignments. Sub-word

features (the set of all prefixes and suffixes) in the discriminative alignment model by Dyer *et al.* (2011a) also allow rich morphology to be handled. Recently, Eyigöz *et al.* (2013) proposed a two-level alignment model where estimated alignments between words and then between morphemes within words mutually inform each other. Rare words are reported to be aligned more reliably than in other models thanks to their frequent morphemes where the correspondence is stronger.

14.4.2 Extraction of Translation Units

The critical component of most MT systems is some dictionary of translation units. If the dictionary is extracted from data, inflection causes the sparse data problem again, as illustrated in Figure 14.3 with the comparison of Czech and German.

In statistical systems, entries in the translation dictionary have to be equipped with a probability estimated, for example, as the fractional co-occurrence count (maximum likelihood estimate, MLE):

$$(4) \qquad p(\text{kočka}|\text{cat}) = \frac{\text{occ}(\text{kočka}, \text{cat})}{\text{occ}(\text{cat})}$$

If the dataset were large and repetitive enough, the counts would be relatively high numbers and the fractions would be reliable estimates of the probability. In real-world datasets, such as the proceedings of the European Parliament (Koehn 2005), we will be lucky to see *cat* co-occurring (and correctly aligned) with *kočka* just a few times.

The key problem in the extraction of translation units and estimating their probabilities is in interpreting low co-occurrence counts. If we have seen *cat* just twice and aligned to *koťátko* (kitten) just once, should we trust it? If the denominator is reasonably high, we can assume a low numerator to be caused by random alignment errors. A low denominator means trouble: the probability of observed translation equivalents is often overestimated, for example the *cat* being translated as *kitten* in 50 per cent of cases, while there are salient translations that were not observed and thus have the probability of zero. This is another instance of the **sparse data issue** and the necessity to **smooth** the observed counts (Foster *et al.* 2006; Kuhn *et al.* 2010).

14.4.3 Translation of Unseen Texts

Translation of unseen texts is the heart of MT. The need for generalization capacity concerns all text units down to word forms: the systems need to accept unseen forms and generate novel forms of known words as required by context.

Dorr *et al.* (1998) more or less restrict the issues of target-side generation to cases where the information is not overt in the source language. This affects both the lexical

value (e.g. the translation of the German *können* as either *know* or *be able to*) as well as morphological properties (e.g. the tense in Chinese-to-English translation).

To date, it seems that richer morphology poses problems for MT even in cases where the information *is* available, simply because the extent of possible choices is too complex to be correctly and fully captured by linguistic rules and/or well covered in available training data.

Languages with rich inflection suffer a great deal from a lack of generalization at the word level. Specifically, most SMT systems are completely unable to produce novel word forms or handle unseen words aside from copying them verbatim to the output (which is a reasonable default for proper names in languages with the same script). It is very common, even for very large training data, that a particular form of a word is never seen in the training data.

If this '**out-of-vocabulary**' issue happens on the source side, the system can in principle resort to translating, for example, the lemma instead of the form. Risking an error in, for example, morphological number for nouns, the system increases the chance of having seen the translation unit (now the more general lemma) in the training data and thus producing an acceptable translation; see also the study on error perception by Kirchhoff *et al.* (2012). Naturally, the generalization capacity is still in the system: it has merely been shifted from the MT engine to the pre-processing step of morphological analysis and/or tagging.

If the out-of-vocabulary issue happens on the target side (i.e. the lemma of the target word needed is known, but the required word form is not available), typical SMT systems struggle between producing the most likely translation, that is, the word in an inappropriate form (even though the language model will not give it a high score) and choosing a less salient translation that is available in the form needed by the context of closely neighbouring words.

14.4.4 Language Modelling

A very influential component of MT systems is the **language model** (LM) that is used to select fluent sentences among the many possible candidates. Formally, the language model defines the probability of the sentence being treated as a sequence of tokens. The standard method is called an n-gram language model, because it decomposes the sentence into all (overlapping) sequences of n consecutive tokens, always predicting the last token of this n-gram given the previous $n-1$ tokens.

While many other sentence decompositions are known to work better, for example those that follow the dependency structure of the sentence (see Chelba and Jelinek 1998; Fox 2002; Popel and Mareček 2010 and the cited works), they are harder to integrate into PBMT (Schwartz *et al.* 2011).

MRLs make the estimation of n-gram LM parameters considerably harder: unless a given n-gram of words is seen in each considered inflection, the model cannot confirm its correctness. Stating that an unseen n-gram is impossible would be too harsh, so LMs are smoothed using one of many proposed methods (Chen and Goodman 1996).

The standard technique is to consider a shorter *n*-gram. So if a particular sentence cannot be confirmed using triples of tokens, only pairs of neighbouring tokens or even individual tokens are checked. Factored LMs (Bilmes and Kirchhoff 2003) allow a linguistically more motivated smoothing path to be specified, for example to preserve the length of the *n*-gram but consider only, for example, the morphological properties of words ignoring their lexical values or vice versa. So if the sequence *big black cats* was never seen, hopefully the sequence *adj adj noun* was. For morphologically rich languages, the reduced model could ensure, for example, agreement in case even for *n*-grams not seen in the training data, provided that the MT system can actually propose them. The improvement based on this linguistically motivated smoothing in PBMT has so far been unfortunately rather small (Yang and Kirchhoff 2006). Recently, success has been reported in a small data setting with advanced smoothing techniques based on stochastic processes (Okita and Way 2010, 2011) and some of these models are even character-based, allowing them to handle morphemes (Mochihashi *et al.* 2009).

While attempts to model the probability of sentences in a clever way help rather marginally, the brute force of more training data and the standard *n*-gram LMs is successful and hard to surpass (Brants *et al.* 2007).

14.4.5 MT Evaluation

MT evaluation can serve multiple purposes. If MT is used as a component in a larger process, it can be evaluated extrinsically, using a method relevant for that particular application, for example Krings and Koby (2001) and O'Brien (2011) focus on manual post-editing efforts while Parton and McKeown (2010) aim at cross-lingual question answering.

Kirchhoff *et al.* (2012) offer a rather unusual perspective, evaluating the intuitive perception of MT errors. Using Google translate from English to Spanish followed by manual correction of MT errors, Kirchhoff *et al.* find that errors in morphology are very common but perceived as far less serious than less frequent errors in word order. Bojar (2011), however, observes for Czech, that errors in word form (including the negation prefix, reversing the meaning of the sentence) can be difficult to spot if the user is presented only with the system output.

The more common goal in the MT research community is to measure translation quality during system development in order to check progress or even improve the system automatically (Section 14.4.6). The most useful in this respect are *automatic* MT evaluation methods. Automatic evaluations are obviously just an approximation of human preferences and do not always match them but they are much faster, cheaper, and also reproducible because they do not depend on an annotator's subjective criteria and capabilities.

The prominence of MT evaluation as such is highlighted by the series of WMT workshops, documented in the works of Koehn and Monz (2006) through

Source	The earnings on its 10-year **bonds** are 28.45%.
Reference	Výnos na jejích 10letých **dluhopisech** je na 28,45%.
System 1	Příjmy na své desetileté **dluhopisy** jsou 28,45%.
System 2	Příjmy na jeho 10-letých **poutech** jsou 28.45%.
Another Reference	**Zisk z** jejích 10-letých **dluhopisů** je 28,45%.

FIG. 14.4 Exact match of tokens is too rigid for MT evaluation. In this English-to-Czech example, both candidates suffer problems (underlined). *Příjmy* corresponds to a slightly different meaning of *earnings* than that used in the original sentence. System 1 selected a wrong case for the translation of *on its 10-year bonds*, resulting in an ungrammatical but understandable sentence. On the other hand, the lexical choice for *bonds* by System 2 is completely wrong, the word *poutech* means handcuffs. In BLEU and other simple evaluation methods, System 1 and System 2 score almost equally. System 1 has preserved the meaning of *bonds* but due to the different case, the word form is not confirmed by the Reference. As the last line with another human translation illustrates, BLEU would equally penalize a different but correct lexical choice (*zisk* vs. *výnos*) and a different morphological variant (*dluhopisů*), although it is correct and required by the different preposition (*z* vs. *na*).

Callison-Burch *et al.* (2012) and Bojar *et al.* (2013b), which regularly include a shared task on automatic MT evaluation and have led to notable improvements. Starting with Callison-Burch *et al.* (2012), one of the tasks is to automatically predict post-editing effort.

The automatic evaluation is usually based on the (monolingual) alignment between the evaluated MT output (also known as the **hypothesis**) and one or more **reference translations**.

If the target language is rich in inflection, it can happen that the hypothesis and the reference share the content words but, due to a different grammatical relation being chosen, they do not match in form. Simple exact token match as implemented in the most widely used metric BLEU (Papineni *et al.* 2002) is unable to align such forms and penalizes a significant portion of the hypothesis just as much as it would penalize completely garbage words, see Figure 14.4 for an example. Bojar *et al.* (2010) report that about one-third of output tokens of English-to-Czech MT systems are not scored by BLEU (because they are not confirmed by the reference) and still do not contain any error based on manual flagging of errors.

Having more references helps to mitigate the issue, because they are more likely to confirm the particular forms chosen in the hypothesis. Dreyer and Marcu (2012) propose a technique that captures 'all' possible reference translations in a compact data structure and observe that naturally occurring English sentences have billions of meaning-equivalent variations. Bojar *et al.* (2013a) adapt the technique for languages with richer morphology and morphological agreements that would be cumbersome to ensure in the framework by Dreyer and Marcu. The observations are similar: hundreds of thousands of correct Czech translations can be produced for a single input sentence and having access to them significantly improves the correlation of BLEU

with human judgements. However, in their initial stages of development, the techniques are extremely expensive, needing about two hours of manual annotation work per sentence.

The task of monolingual word alignment between the reference and the hypothesis has not been studied as much as the bilingual alignment described in Section 14.4.1 (see an open-source implementation by Yao *et al.* (2013) for some references), but many MT metrics strive to overcome the limitations of exact match. For instance, Tantuğ *et al.* (2008) propose a variant of BLEU that considers morphemes and also uses Wordnet similarity to validate root words.

The most elaborate handling of the monolingual alignment is implemented in the metric called METEOR (Banerjee and Lavie 2005; Denkowski and Lavie 2010), with several stages: if word forms cannot be matched, lemmas or even Wordnet synonyms or automatic phrase paraphrases (Bannard and Callison-Burch 2005) are considered.

Kauchak and Barzilay (2006) transform the problem of alignment for MT evaluation into automatic paraphrasing of the whole reference sentence to better match the output of the system, easing the situation for simple MT metrics. However, neither Kauchak and Barzilay nor Madnani and Dorr (2010) in their survey of paraphrasing methods consider morphologically rich languages.

For agglutinative and fusional languages, the issue may also to some extent be solved using letter-BLEU (Yang *et al.* 2008), a variant of BLEU that is applied on sequences of characters instead of words. Words with matching stems but different affixes would still get at least partial credits.

14.4.6 Model Optimization (Tuning)

Current MT system engines always consist of a number of independent components, each associated with a weight. The weights are set automatically in a process called **model optimization** or **tuning** to achieve best performance on a heldout set of sentences, the 'devset'. From a machine learning point of view, this is the actual training of the system.

The heldout set of sentences is repeatedly translated and the hypotheses compared to the reference translation using an automatic evaluation metric. As above, in languages rich in inflection, a mismatch in the exact word form is more likely, making the automatic evaluation less reliable.

Another issue is that the system should find a balance between a focus on errors in lexical choice and errors in form choice. We are not aware of any study of this balance so far. For the widely used BLEU, this means that both types of errors are considered equally severe, which goes against the findings of Kirchhoff *et al.* (2012). On the other hand, several studies so far indicate that it is surprisingly hard to come up with a metric that would perform significantly better in model optimization than BLEU (Callison-Burch *et al.* 2011; Cer *et al.* 2010).

Madnani (2010) reports improvements in tuning by adding automatic paraphrases to the reference translation, reducing the sparsity issue. Instead of taking an automatic paraphraser, Dyer *et al.* (2011b) exploit the fact that their devset was available in multiple languages and report an improvement in the tuning of a German–English system by adding a 'reference' as produced by a Spanish–English MT system. Tamchyna *et al.* (2012) achieve an improvement when tuning English-to-Czech translation on multiple manually created reference translations and Koehn and Haddow (2012b) report better results simply by taking a larger tuning set which also has the positive effect of smoothing out unjustified mismatches with the reference.

14.5 Explicit Handling of Morphology in MT

In Section 14.4, we described in detail how each and every processing step of the general MT pipeline (Section 14.3.2) struggles with rich morphology. In this section, we survey old and contemporary MT systems that do not always quite follow the MT pipeline and, more importantly, include specific modules or procedures to handle morphology.

14.5.1 Shallow Approaches for Closely Related Languages

Very close languages such as Czech-Slovak or Spanish-Catalan often differ primarily at the morphological level of representation and share syntactic properties. This lends such language pairs to shallow and direct MT approaches. Word order can be almost preserved and sometimes even the morphological systems are very similar: the same morphological features are overt on the surface and their values can be directly mapped to the other language.

An MT system can thus be limited to only performing morphological analysis, including tagging, replacing lemmas and morphological tags in the sequence of tokens, and generating the target-language word form. Depending on the vocabularies of the two languages, the translation of the lemma may need some word-sense disambiguation module, but the morphological properties are often mapped to the target language deterministically. Successful examples of such shallow systems include Apertium (Corbí-Bellot *et al.* 2005)[4] originally developed for the Romance languages of Spain but gradually extended to cover also less related languages, and Česílko (Hric *et al.* 2000; Homola *et al.* 2009).

Figure 14.5 illustrates the shallow translation by Česílko from Czech to Slovak including two errors in named entities. Given the nearly one-to-one mapping of

[4] <http://www.apertium.org/>.

Source	Barack	Obama	dostane	jako	čtvrtý	americký	prezident	Nobelovu	cenu	míru
Gloss	Barack	Obama	will-get	as	the-fourth	American	president	Nobel	prize	of-peace
S. Lemmas	Barack	Ob	dostat	jako-2	čtvrtý	americký	prezident	Nobelův	cena	mír
S. Tags	X@—	NNIP7	VB-S-	Db—	CrMS1	AAMS1	NNMS1	AUFS4	NNFS4	NNIS2
Output	Barack	Ob	dostane	ako	štvrtý	americký	prezident	Nobelův	cenu	mieru
Corrected	Barack	Obama	dostane	ako	štvrtý	americký	prezident	Nobelovu	cenu	mieru

FIG. 14.5 Sample input and output of Česílko, including intermediate Czech lemmas and tags (simplified). There are two errors in the output (underlined) and both are caused by insufficient coverage of the morphological dictionary. The name Obama is mis-interpreted as the instrumental (case 7) of the river Ob while the possessive form of Nobel, *Nobelův* is not covered by the target-side dictionary.

morphological properties, inflection does not pose any unexpected challenge except for unknown words and names in particular.

14.5.2 Morphology in Phrase-based MT

This section reviews a range of modifications of the phrase-based model (Section 14.3.3) that treat the input and output in a more adequate way than just atomic word forms.

14.5.2.1 *Factored phrase-based MT*

Koehn and Hoang (2007) introduce an extension of the phrase-based model aimed at explicit handling of morphology (or other features of languages) called the **factored phrase-based model**. A **phrase** is no longer a sequence of atomic tokens but rather a sequence of vectors, or 'factored tokens'. One of the factors is usually the surface form of the word while other factors can represent any information the author of the system deems relevant.

The configuration specifies the order in which factors are considered and filled, see Figure 14.6 for an example. The whole preparation of the factored output tokens called 'translation options' is performed in an initial phase with no access to neighbouring words. The standard PBMT search follows to pick the best combination of these now richer translation options. It is just the language model that helps to select coherent combinations.

Improving target-side morphological coherence is rather easy in factored PBMT models. It is sufficient to introduce additional language models over a subset of output factors, for example part-of-speech or morphological tags (Bojar 2007; Koehn and Hoang 2007; Koehn *et al.* 2010). This computationally inexpensive benefit, however, concerns primarily the *selection* of word forms as seen in the parallel training data. Constructing unseen word forms is much more difficult, if we want to avoid a

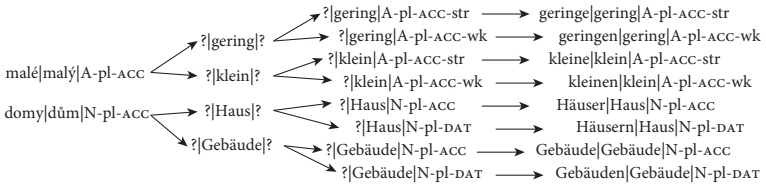


FIG. 14.6 Sample sequence of factored translation of the phrase *small houses* from Czech (*malé domy*, ambiguous in case but marked as accusative in the example) to German (*kleine Häuser* and several other variants). The setup is linguistically motivated: first, Czech lemmas are translated to German lemmas, then Czech morphology is translated to German tags and finally, a German word form is constructed from the lemma and the tag. Later, the best combination of the phrases is selected.

combinatorial explosion of all possible forms; see Section 14.5.2.4 for some promising approaches.

Depending on the language pair, data, and exact configuration, factored setups may perform better or run into new problems such as the combinatorial explosion of translation options compared to PBMT (Bojar *et al.* 2012).

14.5.2.2 *Reducing rich source side*

When the morphologically richer language is on the source side, the translation quality can be increased by stripping the unnecessary details or decomposing complex word forms into separate tokens.

Goldwater and McClosky (2005) extend some of the ideas by Yaser *et al.* (1999) and provide a brief survey of such morphology-stripping methods and evaluate a collection of preprocessing techniques of Czech input when translating to English using a (non-factored) PBMT system. Aside from just reducing Czech word forms to lemmas (optionally equipped with features like number), both Yaser *et al.* and Goldwater and McClosky also introduce pseudo-words, that is, placeholders that will get translated to English auxiliary words.

A minor extension of factored PBMT called 'alternative decoding paths' or 'interpolated back-off' (Birch *et al.* 2007; Bojar and Kos 2010; Koehn and Haddow 2012a) allows us to consider both the original rich forms as well as the reduced variants of tokens, taking whichever is easier to use in the given context. The reduced variant is then usually used only as a fallback for unknown forms. Similar results can be achieved with custom models, for example Nießen and Ney (2001, 2004), who introduce a hierarchical translation lexicon where the source word is searched for using gradually less and less specific morphological constraints, or MT techniques capable of handling ambiguous input like confusion networks (Dyer 2007) or lattices (Wuebker and Ney 2012), see Figure 14.7. Nakov and Ng (2011) suggest one more technique to relax the match between the 'translation dictionary' as extracted from the training corpus and

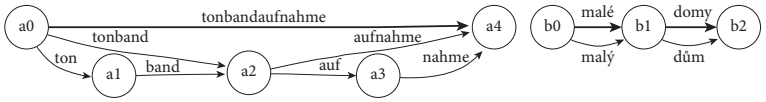


fig. 14.7 A sample lattice from Dyer (2009) encoding several segmentation possibilities of the German word *Tonbandaufnahme* (audio recording) and a sample confusion network encoding lemmatization of the Czech words *malé domy* (small houses). The original input is displayed as thick edges, the alternative paths serve as fall-back options. Confusion networks are a special case of lattices where all paths reconnect after every edge.

the input. Instead of relying on morphological tools for the source language to produce alternative source tokens, they use automatic paraphrases of words or phrases (Bannard and Callison-Burch 2005). The added benefit is that some divergence in lexical choice can also be accommodated.

For agglutinative and compounding languages, the reduction in source-side richness is better achieved by segmenting of complex words. A range of works confirm that such decomposition (or 'decompounding' as used in the MT community) is useful for German (Koehn and Knight 2003; Alfonseca *et al.* 2008; Stymne 2008; Dyer 2009; Hardmeier *et al.* 2010). The work of de Gispert *et al.* (2009) decomposes Arabic and Finnish input in many ways, including the unsupervised morphology of Morfessor (Creutz and Lagus 2007) and automatically selects the decomposition that works best for a particular sentence. Nguyen *et al.* (2010) use a rather different underlying model to achieve the same effect for translating from Arabic and Chinese to English. Virpioja *et al.* (2010) continue the experiments with German and Czech as source languages; the improvements in translation quality are, however, obtained only when the final outputs are constructed by combining hypotheses from multiple segmentation options.

14.5.2.3 *Augmenting poor source side*

Avramidis and Koehn (2008) enrich English tokens with artificial case markers for nouns and accompanying parts of speech (adjectives, articles, and determiners) and artificial person markers for verbs based on the parse tree. The increase in data sparseness of the source side (due to the richness of the morphology) does not cause much harm, because it is in line with the target language properties (Greek and Czech), where such agreement is required. The underlying SMT model is factored phrase-based using alternative decoding paths to prefer the augmented English tokens but resort to plain tokens if necessary.

Yeniterzi and Oflazer (2010) use linguistically motivated rules for pre-processing English when translating to Turkish and gain significant improvements with a small training corpus. They join English tokens to mimic the morphological properties of Turkish, for example concatenating the conjunction *if* with the verb in the clause or appending the preposition to the head noun of a noun phrase. Assuming that the

underlying parser does its job well, the features can come from quite distant auxiliary words, effectively allowing some 'gaps' in phrases and handling, for example 'in their … economic relations' as one translation unit (see also Section 14.5.3). A side effect of this transformation is that the number of input tokens drops by 30 per cent, making the number of tokens on both sides similar. Interestingly, a different set of rules aimed at constituent reordering (as opposed to just attaching English auxiliary words to the corresponding content words) did not bring any improvement.

Again, the underlying model is factored phrase-based. This time, an additional target-side factor with Turkish morphological tags is used to allow for an additional language model. Yeniterzi and Oflazer also use the alternative decoding paths to resort to the original English source token if the augmented one was not seen in the training data.

Ramanathan *et al.* (2009) improve English–Hindi factored translation in a small data setting by reordering English in a pre-processing step to better match the Hindi SOV word order and replacing English word forms with separate streams of lemmas, suffixes, and automatic semantic relations. English lemmas are translated to Hindi lemmas while English suffixes, and semantic relations are mapped to Hindi suffix or case markers. The final generation step combines Hindi lemma and marker streams to the stream of Hindi word forms. No alternative decoding path (e.g. to ignore the English suffix and relation if their combination is not known) is allowed but the model still outperforms the baseline given the very small training data.

14.5.2.4 *Improving generative capacity on the target side*

Some attempts have been made to increase the capacity for generating new forms. These experiments so far focus on language pairs with only the target side morphologically richer.

A rule-based generation component is used by de Gispert *et al.* (2005) to produce unobserved conjugations of Spanish verbs.

Oflazer and El-Kahlout (2007) describe a set of preprocessing techniques for English-to-Turkish translation. Word forms in the target side of the training data are split into morphemes, pseudo-words are added for features such as verb tense (e.g. '+vvn' to indicate passive) and novel word forms are constructed using a separate component from the concatenated stems and pseudowords prior to final scoring. Figure 14.8 illustrates the modified English' and Turkish'. Producing the final output tokens in the agglutinative language would be very difficult for the standard PBMT, while the split Turkish (e.g. '**kat** +hl +ma' that deterministically maps to 'katılma') allows the simple PBMT model to translate 'accession' into the stem 'kat' and add the necessary morphemes based on the English auxiliary words. For example, the verbal noun indicator '+ma' probably often co-occurs with the English definite article of deverbal nouns and the model thus learns to translate 'the' into '+ma'. Note that this can already be seen as a simple variant of the so-called 'analysis–transfer–synthesis' approach, see Section 14.5.4.

Input	the implementation of the accession partnership will be monitored in the framework of the association agreement .
English'	the **implementation**$_1$ of the **accession**$_2$ **partnership**$_3$ will be **monitor**$_4$ +vvn in the **framework**$_5$ of the **association**$_6$ **agreement**$_7$.
Turkish'	**kat**$_2$ +hl +ma **ortaklık**$_3$ +sh +nhn **uygula**$_1$ +hn +ma +sh , **ortaklık**$_6$ **anlaşma**$_7$ +sh **çerçeve**$_5$ +sh +nda **izle**$_4$ +hn +yacak +dhr .
Output	katılma ortaklığının uygulanması , ortaklık anlaşması çerçevesinde izlenecektir .
Gloss	accession partnership's application , partnership agreement in-framework will-be-followed.

FIG. 14.8 Pre-processed English and Turkish (Oflazer and El-Kahlout 2007). For the explanation, content words are boldfaced and the English–Turkish counterparts are co-indexed. A standard PBMT model operates on the modified English' and Turkish' representations (without the boldfacing and subscripts).

Several works (Toutanova *et al.* 2008; Fraser 2009; Bojar and Kos 2010; Fraser *et al.* 2012) use an intermediate 'language' with the target-language word order and lexicon but reduced morphological richness. This artificial language (including the necessary training data) is created by reducing target-side morphological features to a bare minimum. The first step of the translation, performed using a standard phrase-based system, is responsible for most of the transfer and requires parallel data to train. The second step handles the necessary inflection and it can be trained on much larger monolingual data. While successful on small datasets, the benefit of the method diminishes with large datasets. Fraser *et al.* (2012) use separate models for predicting individual morphological features (case, number, gender, and weak or strong inflection) and are the first to show gains even in a large data setting. Related experiments for the hierarchical model are reported by Weller *et al.* (2013), see Section 14.5.3 below. Clifton and Sarkar (2011) apply the two-step approach for English-to-Finnish, combining it with 'segmented translation' as Oflazer and El-Kahlout (2007), that is, operating the PBMT model on artificial tokens that correspond to (unsupervised) morphemes instead of words.

Somewhat related to segmented translation are systems that produce unseen compounds in Germanic languages (Stymne and Cancedda 2011; Fraser *et al.* 2012).

The simplest methods that boast 'more powerful' target-side generation use just additional target-side only texts. Sometimes dubbed **reverse self-training** (Bertoldi and Federico 2009; Bojar and Tamchyna 2011; Lambert *et al.* 2011), the approach uses an auxiliary MT system trained in the reverse direction (from the morphologically richer language) to translate large monolingual data to the source language. This synthetic parallel corpus is used to train an improved MT system. With some back-off in the reverse translation (e.g. if the form is not known, translate using the lemma), this approach learns to generate word forms never seen in the original parallel data.

14.5.3 Hierarchical and Surface-Syntactic MT

Presented as an extension of the phrase-based model, the **hierarchical phrase-based model** (Chiang 2005, 2007) correctly handles the hierarchical structure of sentences. Phrases (in the non-linguistic sense of token sequences) can now contain **gaps** where other phrases fit. In the hierarchical model, this composition is not restricted by the type of the phrase in any way; in a (surface) **syntactic model**, phrases and gaps are labelled with non-terminals that have to match, formally making a **synchronous context-free grammar** (Chiang and Knight 2006). If the non-terminal labels and the recursive structure come from a treebank, we have a truly syntax-based translation. The motivation for such a model from the linguistic point of view is obvious and includes, for example the chance to capture long distance dependencies between words (e.g. agreement in some morphological feature). The non-terminals in hierarchical model can be used to encode not only syntax but any other latent feature (information not overt on the surface of source or target languages), for example Baker *et al.* (2012) use this for better handling of modality and negation.

The additional constraint requiring non-terminals to match has to be introduced with great caution. An option to resort to non-matching phrase has to be allowed in order to preserve the performance of the plain hierarchical model. Otherwise, the rigid syntactic model effectively reduces the available training data by disabling non-matching phrases (Bojar and Hajič 2008; Chiang 2010).

Across language pairs, hierarchical translation has not quite outperformed PBMT. The added complexity and constraints of the structure do not always pay off. Since the search for a robust hierarchical model is still in progress, relatively few people have tried to focus specifically on morphology in this model.

Williams and Koehn (2011) are probably the only ones who attempt to formally capture agreement constraints in the hierarchical model using proper unification instead of simple identity of non-terminals. Weller *et al.* (2013) use the hierarchical model as the basis for their two-step approach (Section 14.5.2.4) and report gains when predicting case for German noun phrases using a range of features up to subcategorization frames.

14.5.4 Systems Following Analysis–Transfer–Synthesis Sequence

Approaches to MT that follow the analysis–transfer–synthesis sequence (Vauquois 1975; Vauquois and Boitet 1985) include separate components for handling morphology. The lemma and the various morphological features of words are separated during the analysis phase. The transfer can thus handle morphological and lexical divergence separately. Finally, a fully-fledged morphological synthesis can produce forms never seen in the training data. The exact specification of the morphological component and the set of observed features is system-dependent.

Unfortunately, not many such systems have made it up to the 'production' state. We are aware of two commercial systems: MT-MSR (Richardson *et al.* 2001), later discontinued, and Lucy Technologies (Wolf *et al.* 2010), originally based on the METAL system (Bennett and Slocum 1985), and a few research systems.

A moderately-sized MT system for the translation from Norwegian to English was achieved in the LOGON project (Bond *et al.* 2005; Oepen *et al.* 2007; Bond *et al.* 2011) using an interesting combination of Norwegian analysis in Lexical Functional Grammar (LFG, Bresnan 2001), transfer in Minimal Recursion Semantics (Copestake *et al.* 1995), and English generation using Head-driven Phrase Structure Grammar (HPSG, Pollard and Sag 1994).

Meaning-Text Theory (MTT, Mel'čuk 1988; Kahane 2003), is exploited in the MT system ETAP-3 (Apresjan *et al.* 2003; Boguslavsky *et al.* 2004).[5] The transfer happens at a normalized syntactic representation (NormS), slightly above surface syntax.

TectoMT (Žabokrtský *et al.* 2008; Dušek *et al.* 2012)[6] is based on the tectogrammatical layer (t-layer) of linguistic representation (Sgall *et al.* 1986) also known from Prague dependency treebanks (Hajič *et al.* 2006; Hajič *et al.* 2012) and translates only from English to Czech. Compared to MTT, the t-layer is less semantic and stops at the level of lemmas and morphosyntactic realizations of relations between words (e.g. capturing preposition and case instead of some deep syntactic or semantic role type). Transfer at the t-layer thus has to handle context-dependent choices in case markers and other morphosyntactic markers. TectoMT uses a statistical model for this (Žabokrtský *et al.* 2010), circumventing the need to manually encode lexical functions or grammatical rules.

Bojar and Hajič (2008) and Bojar and Týnovský (2009) document the additional problems that such a complex processing pipeline creates compared to shallow or direct approaches. This explains to some extent why deep approaches have not surpassed the performance of simpler models yet, despite the obvious improvement in linguistic adequacy. TectoMT remains probably the only such deep system that performs reasonably well in the broad domain of news (Callison-Burch *et al.* 2010).

14.5.5 System Combination and Corrective Approaches

So-called 'system combination' techniques can be used to benefit from the strengths of diverse types of MT systems such as lexical choice of large-data PBMT and better grammar and unseen word forms produced by more syntactic and/or rule-based systems. The seminal work on MT system combination (Matusov *et al.* 2008) follows the idea originally introduced for automatic speech recognition (Fiscus 1997). This one and subsequent variations, for example Heafield and Lavie (2010), are generally based

[5] <http://cl.iitp.ru/etap>.
[6] <http://ufal.mff.cuni.cz/tectomt/>.

on some voting for individual tokens: if many baseline systems produce a particular word form, then it should probably appear in the combined output.

None of the system combination techniques addresses inflection explicitly, all rely on the standard *n*-gram LMs (Section 14.4.4). It is thus again only the size of the dataset that is supposed to bring some guarantee of grammaticality.

Rosa *et al.* (2012) take a different approach and implement Depfix, a rule-based system that fixes (primarily morphological) errors in a baseline MT output given automatic syntactic analyses of both the source and the hypothesis. A complex ensemble of the deep syntactic TectoMT (Section 14.5.4), the phrase-based Moses in a factored setup (Section 14.5.2.1) and Depfix for final correction was the best performing English-to-Czech system in WMT13 shared task (Bojar *et al.* 2013c). Aside from various changes in verb conjugation, the most reliable automatic correction was the re-introduction of lost negation.

14.6 SUMMARY

The inflection and morphological richness of source and/or target languages introduce extra complexity to almost every processing step of machine translation systems, starting from the alignment of words in parallel texts up to MT system evaluation.

In the history of machine translation, the relatively morphologically poor English has long been the most common target language. Specific handling of inflection and rich morphology has thus received proper attention only rather recently. The currently prevailing data-driven approaches still have a long way to go until they adequately capture and apply the necessary morphological generalizations. The former opponents, rule-based vs. statistical methods, have grown very close to each other and we expect even further convergence on this journey to correct generalizations.

It is primarily the underlying material that forms the interest and focus of research. We therefore expect many influential discoveries to arise from the study of machine translation between morphologically rich but divergent languages, a sector where the explorations have barely started.

While the quantity of parallel and monolingual texts suitable for the training of MT systems is growing every minute, fine-grained models of inflection (and word formation) remain a needed component of general-purpose MT systems, because new word forms are constantly being created. Sooner or later such models will be designed and become a part of the standard MT pipeline.

The Joy of Parallelism with CzEng 1.0

**Ondřej Bojar, Zdeněk Žabokrtský,
Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček,
Jiří Maršík, Michal Novák, Martin Popel, Aleš Tamchyna**

Charles University in Prague, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
surname@ufal.mff.cuni.cz *except* {mnovak,odusek}@ufal.mff.cuni.cz, jiri.marsik89@gmail.com

Abstract

CzEng 1.0 is an updated release of our Czech-English parallel corpus, freely available for non-commercial research or educational purposes. In this release, we approximately doubled the corpus size, reaching 15 million sentence pairs (about 200 million tokens per language). More importantly, we carefully filtered the data to reduce the amount of non-matching sentence pairs.
CzEng 1.0 is automatically aligned at the level of sentences as well as words. We provide not only the plain text representation, but also automatic morphological tags, surface syntactic as well as deep syntactic dependency parse trees and automatic co-reference links in both English and Czech.
This paper describes key properties of the released resource including the distribution of text domains, the corpus data formats, and a toolkit to handle the provided rich annotation. We also summarize the procedure of the rich annotation (incl. co-reference resolution) and of the automatic filtering. Finally, we provide some suggestions on exploiting such an automatically annotated sentence-parallel corpus.

Keywords: Czech-English parallel corpus, automatic parallel treebank, training data for machine translation

1. Introduction

We present the new release of a Czech-English parallel corpus with rich automatic annotation, CzEng 1.0.[1]
CzEng 1.0 is a replacement for CzEng 0.9 (Bojar et al., 2010) which was successfully used in various NLP experiments including the machine translation evaluation campaigns of 2010 and 2011 (Callison-Burch et al., 2010; Callison-Burch et al., 2011).[2] Both the old and the new release are freely available for research purposes; restricted versions of CzEng 0.9 have also their commercial applications. With 8 million parallel sentences, CzEng 0.9 moved Czech out of the "low resource" rank of languages. While we did not primarily focus on increasing the overall size of the corpus, CzEng 1.0 nevertheless doubled the size of parallel Czech-English data available for research. More details are available in Section 2.
In CzEng 1.0, our main aim was to improve the quality of the resource. We focused on:

- User access to the rich annotation (Section 3.),
- Improved rich annotation, including automatic co-reference (Section 4.),
- Filtering of the sentence pairs to increase the precision of the corpus (Section 5.).

We believe this large and richly annotated resource will be of interest not only to the machine translation community but also to many other NLP researchers. Our first examples utilizing the parallelism (aside from the obvious applications in machine translation) are given in Section 6.

[1] http://ufal.mff.cuni.cz/czeng/
[2] http://www.statmt.org/wmt10, http://www.statmt.org/wmt11

2. Core CzEng 1.0 Properties

This section is devoted to basic statistics of the released resource, data sectioning and file formats.

2.1. CzEng 1.0 Data Sizes

Table 1 lists the total number of parallel sentences and Czech and English surface tokens per source. Please note that the number of tokens includes punctuation marks and other symbols.
In Table 1, we also list the number of nodes in the deep syntactic layer of representation (see Section 4.), which roughly correspond to content words in the sentences. We can see that English uses about 12% more surface tokens than Czech. The numbers of deep nodes in Czech and English are much closer. The higher number of deep nodes observed for Czech can be attributed to the fact that the procedure of adding artificial nodes for dropped pronouns and similar phenomena is more elaborated in our annotation pipeline than the similar procedure for English.

2.2. CzEng 1.0 Data Structure

CzEng 1.0 is shuffled at the level of "blocks", sequences of not more than 15 consecutive sentences from one source. The original documents thus cannot be reconstructed but some information about cross-sentence phenomena is preserved. Specifically, CzEng includes Czech and English grammatical and textual co-reference links that do span sentence boundaries (see Section 4.2.).
Each "block" comes from one of the text domains (EU Legislation, etc., see Table 1) and the domain is indicated in the sentence ID.
Individual text "blocks", shuffled, are combined to numbered files; each file holds about 200 sentence pairs.

Source Domain	Parallel Sentences	Surface Tokens ("Words+Punct.") Czech	Surface Tokens ("Words+Punct.") English	Deep Nodes ("Content Words") Czech	Deep Nodes ("Content Words") English
Fiction	4,335 k	57,177 k	64,264 k	41,142 k	38,690 k
EU Legislation	3,993 k	78,022 k	87,489 k	56,446 k	52,718 k
Movie Subtitles	3,077 k	19,572 k	23,354 k	14,615 k	14,918 k
Parallel Web Pages	1,884 k	30,892 k	35,455 k	23,141 k	22,057 k
Technical Documentation	1,613 k	16,015 k	16,836 k	11,942 k	11,207 k
News	201 k	4,280 k	4,737 k	3,208 k	2,963 k
Project Navajo	33 k	484 k	557 k	363 k	344 k
Total	15,136 k	206,442 k	232,691 k	150,857 k	142,897 k

Table 1: Sources in CzEng 1.0, including data sizes in thousands.

The files are further organized into 100 similarly-sized sections, the last two of which are designated for development and testing purposes: `00train`, ..., `97train`, `98dtest`, `99etest`. Users of CzEng 1.0 are kindly asked to avoid training on these last 2% of the data.

2.3. CzEng 1.0 File Formats

CzEng 1.0 is available in three data formats: rich Treex XML format, "export format", and parallel plain text.

2.3.1. Treex Format

The primary data format of CzEng 1.0 is the Treex XML, a successor to the TectoMT TMT format used in CzEng 0.9. Treex XML can be processed using the Treex platform or manually browsed in the TrEd tree editor, see Section 3. for details. Users are encouraged to use the Treex toolkit and access the data programmatically using Treex API rather than directly parsing the XML.

2.3.2. Export Format

To facilitate the access to most of the automatic rich annotation of CzEng 1.0 without any XML hassle, we provide the data also in a simple "factored" line-oriented export format. Note that e.g. named entities or co-reference links are not available in the export format at all.
An example and the meaning of all the tab-delimited columns of the export format is given in Table 5 at the end of the paper.

2.3.3. Plaintext Format

The plaintext format is very simple, consisting of just four tab-delimited columns: sentence pair ID, filter score, Czech sentence, and English sentence.
The plain text preserves the original tokenization (i.e. no tokenization) of the source data.

2.4. Brief Summary of the Automatic Annotation

The processing pipeline of CzEng 1.0 was in essence very similar to the the pipeline used in CzEng 0.9, although we replaced some of the tools with their updated versions.

1. The original texts were segmented into sentences using TrTok, see Section 6.1. (preserving the original tokenization).

2. Sentence alignment was obtained using Hunalign (Varga et al., 2005), where we tokenized, lowercased and chopped each token to at most 4 characters to reduce the sparseness of esp. Czech vocabulary. Hunalign was run on each document pair separately and without any shared translation dictionary.

3. All sentences were morphologically tagged and lemmatized with the tools available in the Treex platform (the Morce tagger (Spoustová et al., 2007) and a rule-based lemmatizer for English).

4. We applied GIZA++[3] (Och and Ney, 2000) to obtain alignment between surface tokens. To reduce the data sparseness, GIZA++ was run on Czech and English lemmas, not fully inflected word forms. We aligned all the data in one large process, which needed about 2 days of CPU time to finish. As common in statistical machine translation, GIZA++ was applied in both translation directions and the two unidirectional alignments were symmetrized. We provide outputs of several symmetrization techniques.

5. The word alignment was loaded into the Treex format and all subsequent steps of analysis were carried out within the Treex framework. MST parser (McDonald et al., 2005) was used for surface syntax dependency parsing.

2.4.1. A Note on Node Alignment

Besides the word alignment, CzEng 1.0 is provided with automatic alignment on the tectogrammatical layer as well. Unlike in CzEng 0.9, where the tectogrammatical alignment was created by the trainable *TAlign* tool (Mareček, 2009), the alignment links in CzEng 1.0 are simply projected from GIZA++ intersection word alignment to the corresponding tectogrammatical trees. The number of links produced by this simple projection is higher, which causes higher recall but lower precision.

3. Treex Framework for CzEng 1.0

As mentioned above, all the automatic annotation of CzEng 1.0 was carried out using the Treex multi-purpose NLP framework (Popel and Žabokrtský, 2010).[4] The core modules of the framework are freely available and can be in-

[3] `http://http://code.google.com/p/giza-pp/`
[4] `http://ufal.mff.cuni.cz/treex`

```
# Convert treex.gz to CoNLL format
treex Write::CoNLLX language=en to=f00001en.conll \
      Write::CoNLLX language=cs to=f00001cs.conll \
      -- data.treex-format/00train/f00001.treex.gz

# See the most frequent translations
treex -Lcs Util::Eval tnode='my ($en)=$tnode->get_aligned_nodes_of_type("int");
                             say $tnode->t_lemma . "\t" . $en->t_lemma if $en' \
      -- data.treex-format/00train/f0000?.treex.gz \
 | sort | uniq -c | sort -rn | head -n 20
# prints:
#    593 a           and
#    291 #PersPron   #PersPron
#    222 být         be

# Open a file in the TrEd editor (via a wrapper to support Treex file format)
ttred data.treex-format/00train/f00001.treex.gz
```

Figure 1: Examples of using the Treex command-line interface.

stalled from CPAN.[5] There are a number of NLP tools integrated in Treex, such as morphological taggers, lemmatizers, named entity recognizers, dependency parsers, constituency parsers, and various kinds of dictionaries.

For users of CzEng 1.0, the Treex platform offers a versatile API, a more appropriate way of accessing the Treex XML files than generic XML parsers can offer. Aside from custom export procedures, one can use ready-made *writers* available in Treex. Figure 1 shows how to convert the surface dependency trees to CoNLLX format or emit the most frequent pairs of tectogrammatical lemmas.

The Treex platform also provides a simple wrapper for TrEd,[6] a tree editor which can read Treex XML using a designated plug-in module. TrEd offers the best option for manual inspection of CzEng data.

Figure 2 shows a sample sentence pair (English and Czech) annotated on both analytical (surface syntax, *a-tree*) and tectogrammatical (deep syntax, *t-tree*) layers. The morphological annotation is stored together with the analytical annotation.

4. Rich Annotation

CzEng 1.0 is automatically annotated in the same theoretical framework as the Prague Dependency Treebank (PDT) 2.0 (Hajič, 2004). Many small updates of various annotation steps have happened since CzEng 0.9. Here we focus on the two more complex ones at the deep syntactic layer (also called *tectogrammatical* or *t-layer*): formemes (Section 4.1.) and automatic co-reference (Section 4.2.).

4.1. Formemes

In addition to the PDT 2.0 annotation style attributes, each node at the t-layer is assigned a *formeme* (Ptáček and Žabokrtský, 2006; Žabokrtský et al., 2008) describing its morphosyntactic form, including e.g. prepositions, subordinate conjunctions, or morphological case. The set of possible formemes contains values such as:

- `n:subj`—an English noun in subject position,
- `v:to+inf`—an English infinitive clause with the particle *to*,
- `adj:attr`—attributive adjectives in both languages, or
- `n:k+3`—a Czech noun in dative (third) case with the preposition *k*.

Figure 3 gives an example of other formemes in a sentence. The values are filled in using rule-based modules operating on both t-trees and the corresponding a-trees.

The formeme annotation had already been present in the previous versions of CzEng and had been successfully employed in structural MT (Žabokrtský et al., 2008) and Natural Language Generation (Ptáček and Žabokrtský, 2006) tasks. We use a version improved (mostly on the Czech side) to depict various linguistic phenomena more accurately and to maintain a greater consistency across the two languages (see Section 6.2. for a cross-lingual evaluation). Our modifications involve e.g. treating nominal usages of adjectives as nouns, distinguishing nominal and adjectival numerals, marking case in Czech adjectival complements of verbs, or allowing prepositions with most English verb forms, plus several fixes for erroneous marking of the previous versions.

4.2. Co-Reference Links

In one of the last stages of automatic annotation, the co-reference resolution is performed on both language parts of the corpus. The range of co-reference types annotated in CzEng corresponds to the types present in PDT 2.0 and on the English side of PCEDT 2.0. Namely, it captures the so-called grammatical co-reference and pronominal textual co-reference.

[5] `http://search.cpan.org/search?query=treex`

[6] `http://ufal.mff.cuni.cz/tred/`

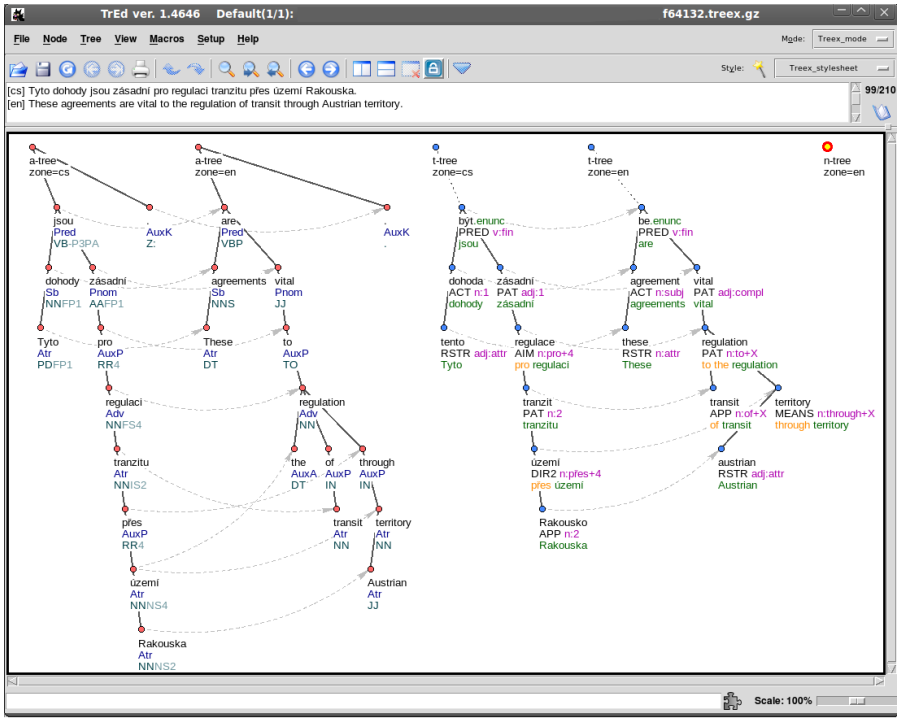


Figure 2: Visualization of one sentence pair in TrEd (Tree Editor). Czech a-tree, English a-tree, Czech t-tree, and English t-tree are presented (left to right). Other attributes which are not shown (e.g. grammatemes) can be inspected after clicking the nodes.

There	*is*	*no*	*asbestos*	*in*	*our*	*products*	*now*	.	”
	be	no	asbestos		#PersPron	product	now		
	v:fin	**n:attr**	**n:obj**		**n:poss**	**n:in+X**	**adv**		

Figure 3: An example sentence with tectogrammatical lemmas and formemes

Grammatical co-reference comprises several subtypes of relations, which mainly differ in the nature of referring expressions (e.g. relative pronoun, reflexive pronoun). However, all of them have in common that they appear as a consequence of language-dependent grammatical rules. This fact allows us to resolve them with a relatively high success rate, using the rule-based system proposed by Nguy (2006). For instance, given a relative pronoun that introduces a relative clause, the parent of the clause head is marked as an antecedent of the pronoun.

On the other hand, the arguments of textual co-reference are not realized by grammatical means alone, but also via context (Mikulová et al., 2006), which makes the resolution far more difficult. To indentify textual co-reference relations with a personal pronoun as the referring expression, we incorporated the perceptron ranking system of Nguy et al. (2009). On the Czech side, we employed the original feature set and trained the system on the PDT data. We used the English side of PCEDT to train the English system, for which we had to limit and modify several features to comply with a somewhat different annotation style.

Table 2 shows the values of pairwise precision, recall and F-score of co-reference resolution on the evaluation part of PDT and PCEDT for Czech and English, respectively. On Czech gold standard trees, the scores are close to those reported by Nguy et al. (2009). Since CzEng annotation is completely automatic, it is necessary to measure the success rate on automatically analyzed trees, so that we can reliably assess the quality of co-reference annotation in CzEng. Unfortunately, one can observe a substantial drop for automatic trees. The reason is twofold.

First, Czech is a pro-drop language, thus the pronouns must be reconstructed on the tectogrammatical layer. Nonetheless, the number of personal pronouns reconstructed incor-

	Gold Standard Features			Automatic Features			Oracle Gender and Number		
Language	P	R	F	P	R	F	P	R	F
Czech	77.06	77.58	77.32	55.23	46.14	50.28	65.70	54.89	59.81
English	45.52	58.69	51.27	44.53	57.32	50.12	–	–	–

Table 2: Results of the co-reference resolution evaluation. The precision, recall and F-score were measured on both languages using the features coming either from the gold standard or the automatic annotation. In the last three columns, the features were automatic except for the manual gender and number.

rectly or not at all accounts for 25% of all pronouns elided on the surface layer (and 15% of all personal pronouns).
Second, gender and number of some pronouns cannot be disambiguated without the knowledge of co-reference links. At the same time, gender and number information is one of the most valuable features in our co-reference resolver. While all attributes are disambiguated in manually annotated trees, they are left ambiguous in automatically analyzed data, which certainly decreases the quality of co-reference resolution. This claim is confirmed by our oracle experiment: when we replaced the automatic gender and number with the manually assigned values, the F-score improved by almost 10% absolute (see the last three columns of Table 2).
As regards the co-reference resolution in English, the difference between its quality using manual and automatic trees is not as dramatic as in Czech. This further confirms the above-mentioned reasons for the success rate drop in Czech since both of the issues (pro-drop recovery and gender and number disambiguation) are marginal in English.
We would like to emphasize that the presented experiments on co-reference resolution are to our knowledge the first for Czech using no gold standard features and one of a few for English employing the deep syntactic layer.

5. Filtering Sentence Pairs

The amount of data included in CzEng along with the varying reliability of its sources (such as volunteer-submitted movie subtitles) demand an automatic method for recognizing and filtering out bad sentence pairs.
Simple filters have been used in previous editions of CzEng. Details about their evaluation and suggestions for improvements can be found in Bojar et al. (2010). We extend the previous work by adding several new filters and introducing a robust method for their combination.
Filtering features for CzEng 1.0 exploit all layers of automatic annotation and include:

- indication of Czech and English sentences' identity,
- lengths of sentences and the words contained in them,
- no Czech (English) word on the Czech (English) side,
- various checks for remains of meta-information, such as HTML tags or file paths,
- use of a translation dictionary to determine the coverage of English words by the Czech side,
- score of symmetrized automatic word alignment obtained by GIZA++,

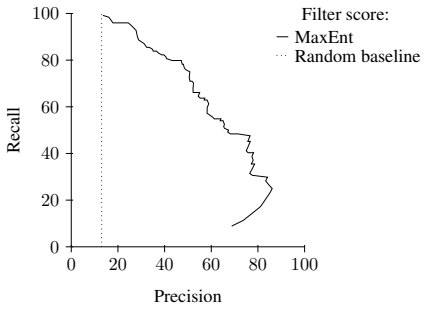


Figure 4: Precision and recall of CzEng filters.

- matching part-of-speech tags,
- matching grammatical number, verb tense or presence of comparative/superlative modifiers.

Wherever possible, we try to model the feature as a ratio or score and empirically find interval bounds for its quantization.
The features are combined to form a single score using a classifier trained to distinguish between correct and wrong sentence pairs. We evaluated the performance of decision trees, naive Bayes classifier, and maximum entropy classifier. We found the maximum entropy classifier to be best suited for this setting. Figure 4 shows the trade-off between precision and recall for all threshold settings. Note that the random baseline stays at roughly 13% regardless of the threshold—our evaluation data consists of 1000 manually annotated sentence pairs, out of which 124 were marked as wrong.

5.1. Precision-Size Trade-off for CzEng Users

Since our filter combination is still not reliable, we include all sentences that pass the threshold of 0.3 in CzEng 1.0, favoring precision of the filtration over recall. We also provide the score assigned by our filters to each sentence pair so that users can create a cleaner, more strictly filtered subset of CzEng 1.0.
Moreover, 2330 input documents containing 60% or more sentences with scores below the threshold were discarded entirely.

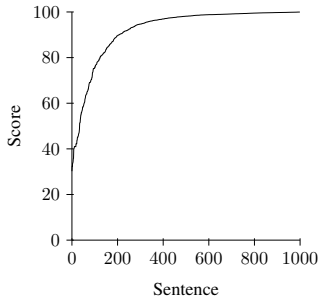


Figure 5: Distribution of sentence filter scores in a random 1000-sentence sample.

Corpus and Domain		Sents	BLEU	Vocab. [k] En	Cs
CzEng 0.9	all	1M	14.77±0.12	187	360
CzEng 1.0			**15.23±0.18**	221	396
CzEng 0.9	news	100k	**14.34±0.05**	53	125
CzEng 1.0			14.01±0.13	47	113

Table 3: Results of MT evaluation.

	Formeme Detection on Automatic Trees	Manual Trees
Baseline	1.5981	1.6680
Improved	**1.6873**	**1.7092**

Table 4: The impact of an improved design of formemes on mutual information (in bits) of Czech and English formemes of aligned t-tree nodes.

5.2. Evaluation of Data Quality

The distribution of filter scores in sentence pairs as shown in Figure 5 suggests that most of the corpus is clean, containing correct sentence pairs.

We also evaluated the quality of CzEng 1.0 extrinsically by conducting a small machine translation experiment. We trained contrastive phrase-based Moses SMT systems (Koehn et al., 2007)—the first one on 1 million sentence pairs from CzEng 0.9, the other on the same amount of data from CzEng 1.0. Another contrastive pair of MT systems was based on small in-domain data only: 100k sentences from the *news* sections of CzEng 0.9 and 1.0, respectively. For each setting, we extracted the random sentence pairs 5 times to avoid drawing conclusions from possibly biased data selection.

For tuning and evaluation, test sentences from WMT 2010 and 2011 were used, respectively. These sets are from the news domain. We used the News Crawl Corpus 2011 data to train the language model.

We measure the translation quality using the standard SMT metric BLEU (Papineni et al., 2002). Table 3 shows the mean BLEU score and standard deviation for each data set. In the setting with 1 million random sentence pairs, using data from CzEng 1.0 is noticeably beneficial for MT quality—the absolute BLEU gain is roughly 0.4 points. This improvement stems from the overall quality of the data, the distribution of domains in CzEng 1.0 is also likely to play a certain role.

On the other hand, using only the news data reverses the situation—CzEng 1.0 data lead to a system with slightly worse performance. We verified our results using Welch two-sample t-test and found that in both cases the difference is statistically significant on 99% confidence level.

An explanation is suggested by the last two columns. The filtering has probably caused a loss in vocabulary size (distinct token types) for both English and Czech in the news domain but not across domains.

6. The Joy of Parallelism

Here we mention several steps in CzEng automatic annotation that make use of the parallel data for improved output quality.[7]

6.1. Tokenizer

CzEng 1.0 uses TrTok, a fast re-implementation of the trainable tokenizer (Klyueva and Bojar, 2008) for sentence segmentation. Its main advantage is the fact that different data sources may need different segmentation patterns (e.g. legislation texts need segment breaks after commas) and TrTok can be guided to follow the patterns by providing enough sample data in the desired form.

By examining segments that were aligned to 1-2 and 2-1 clusters, we often find them to be a consequence of a mismatch in segmentation rules for Czech and English. Such snippets of parallel data can thus directly serve as additional training data for TrTok.

6.2. Formemes

Table 4 compares the mutual information (MI) of Czech and English formemes of t-tree nodes aligned one-to-one for the baseline set of formemes and the improved set of formemes measured on the Prague Czech-English Dependency Treebank 2.0 (PCEDT 2.0, (Bojar et al., 2012)).[8] The higher the MI, the easier the transfer phase in structual machine translation (Žabokrtský et al., 2008) is expected. We measure the MI in two setups—we either utilize the manual trees provided in PCEDT 2.0 directly,[9] or take just the sentences from PCEDT 2.0 and apply to them the automatic annotation pipeline which we use for the whole CzEng 1.0 corpus.

Our initial measurements showed a slight MI drop on the automatic trees, which led us to the discovery of several bugs in both formeme detection rules and conversion of a-trees to t-trees (e.g. problems with infinitive and passive verb forms detection or coordinated modal verbs).

[7] We leave aside the joy of parallel *processing* of the data, very useful i.a. in debugging on large datasets.

[8] `http://ufal.mff.cuni.cz/pcedt2.0`

[9] The used t-trees were manual for both languages; however, only automatic a-trees are available on the Czech part in the PCEDT 2.0.

The corrected analysis pipeline and formeme detection show an MI increase for both manual and automatic trees (see Table 4), which indicates that the new set of formemes is likely to improve the MT transfer phase. Again, we used here the parallel view to fine-tune a monolingual processing step.

6.3. Co-Reference—Future Work

The automatic co-reference annotation for one of the languages in the parallel corpus could be improved if we employed the information from the other language side.
English is considered to be lacking grammatical gender (except for pronouns) and the majority of nouns in English are referred to by a pronoun in neuter gender. On the other hand, Czech distinguishes between four grammatical genders whose distribution among nouns is rather balanced and, moreover, personal pronouns usually agree in gender with a noun they co-refer with.
Thus, we suggest to incorporate the results of Czech co-reference resolution into the English resolver, which might limit the number of antecedent candidates that are in consideration. Conversely, Czech is a pro-drop language, which allows us to utilize the English side to potentially project some of the pronouns that are elided in Czech.

7. Conclusion

We presented CzEng 1.0, the new release of a large Czech-English parallel corpus with rich automatic annotation. The corpus is freely available for non-commercial research and educational purposes at our web site:

`http://ufal.mff.cuni.cz/czeng`

CzEng 1.0 can serve as large training data for linguistically uninformed approaches, e.g. to machine translation, but it can also be directly used in experimenting with cutting-edge NLP tasks such as co-reference resolution validated across languages. We have also provided two examples of exploiting the parallelism of the data to improve monolingual processing: sentence segmentation and formeme definition.

8. Acknowledgements

The work on this project was supported by the project EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003+7E11051 of the Czech Republic), Czech Science Foundation grants P406/10/P259 and 201/09/H057, GAUK 4226/2011, 116310, and the FAUST project (FP7-ICT-2009-4-247762 of the EU and 7E11041 of the Czech Republic). This work has been using language resources developed and/or stored and/or distributed by the LINDAT-Clarin project of the Ministry of Education of the Czech Republic (project LM2010013).

9. References

Ondřej Bojar, Adam Liška, and Zdeněk Žabokrtský. 2010. Evaluating Utility of Data Sources in a Large Parallel Czech-English Corpus CzEng 0.9. In *Proceedings of the Seventh International Language Resources and Evaluation (LREC'10)*, pages 447–452, Valletta, Malta, May. ELRA, European Language Resources Association.

Ondřej Bojar, Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Silvie Cinková, Eva Fučíková, Marie Mikulová, Petr Pajas, Jan Popelka, Jiří Semecký, Jana Šindlerová, Jan Štěpánek, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2012. Announcing Prague Czech-English Dependency Treebank 2.0. In *Proceedings of the Seventh International Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, May. ELRA, European Language Resources Association. In print.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden, July. Association for Computational Linguistics. Revised August 2010.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.

Jan Hajič. 2004. Complex Corpus Annotation: The Prague Dependency Treebank. In *Insight into Slovak and Czech Corpus Linguistics*, Bratislava, Slovakia. Jazykovedný ústav Ľ. Štúra, SAV.

Natalia Klyueva and Ondřej Bojar. 2008. UMC 0.1: Czech-Russian-English Multilingual Corpus. In *Proc. of International Conference Corpus Linguistics*, pages 188–195, October.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

David Mareček. 2009. Improving word alignment using alignment of deep structures. In Václav Matoušek and Pavel Mautner, editors, *Text, Speech and Dialogue*, volume 5729 of *Lecture Notes in Computer Science*, pages 56–63. Springer Berlin / Heidelberg. 10.1007/978-3-642-04208-9_11.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 523–530, Vancouver.

Marie Mikulová, Alevtina Bémová, Jan Hajič, Eva Hajičová, Jiří Havelka, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Petr Pajas, Jarmila Panevová, Magda Razímová, Petr Sgall, Jan Štěpánek, Zdeňka Urešová, Kateřina Veselá, and Zdeněk

Col.	Sample	Explanation
1	subtitles-b2-00train-f00001-s8	ID specifying the domain, block number, train/dev/test section, file number and sentence within the file.
2	0.99261036	Filter score indicating the quality of the sentence pair. The score of 1 is perfect pair, pairs below 0.3 are removed.
	Czech	

3	`Zachránil|zachránit_:W|VpYS---XR-AA---|1|0|Pred`
	`mi|já|PH-S3--1-------|2|1|Obj`
	`milovaný|milovaný_^(*2t)|AAIS1----1A----|4|5|Atr`
	`krk|krk|NNIS1-----A----|5|1|Obj .|.|Z:-----…`
		Czech a-layer (surface-syntactic tree) in factored form: word-form|lemma|morphological-tag|index-in-sentence|index-of-governor|syntactic-function.

4	`zachránit|PRED|1|0|complex|v:fin|v|-|neg0|ant|ind|decl|-|...` `#PersPron|ADDR|2|1|complex|n:3|n.pron.def.pers|sg|-|-|...`	Czech t-layer (tectogrammatical tree): t-lemma|functor|index-in-tree|index-of-governor|nodetype|formeme|semantic-part-of-speech|... and many detailed t-layer attributes.
5	0-0 1-1 2-2 3-3 4-4	Correspondence between Czech a-layer and t-layer for content words. Indexed from 0.
6		Correspondence between Czech a-layer and t-layer for auxiliary words. Indexed from 0.
	English	

7	`He|he|PRP|1|2|Sb saved|save|VBD|2|0|Pred my|my|PRP$|3|4|Atr`
	`ever-lovin|ever-lovin|NN|4|6|Atr '|'|''|5|6|AuxG`
	`neck|neck|NN|6|2|Obj .|.|7|0|AuxK`
		English a-layer (surface-syntactic tree) in factored form: word-form|lemma|tag|index-in-sentence|index-of-governor|syntactic-function.

8	`#PersPron|ACT|1|2|complex|n:subj|n.pron.def.pers|sg|-|-|...`
	`save|PRED|2|0|complex|v:fin|v|-|neg0|ant|ind|decl|-|-|...`
	`#PersPron|APP|3|4|complex|n:poss|n.pron…`
		English t-layer (tectogrammatical tree): t-lemma|functor|index-in-tree|index-of-governor|nodetype|formeme|semantic-part-of-speech|... and many detailed t-layer attributes.

9	0-0 1-1 2-2 3-3 5-4	Correspondence between English a-layer and t-layer for content words. Indexed from 0.
10	4-4	Correspondence between English a-layer and t-layer for auxiliary words. Indexed from 0.
	Cross-Language Alignments Between Surface Czech and English Always indexed from 0, Czech-English.	
11	0-1 1-2 2-2 3-3 4-5 5-6	GIZA++ alignments "there" for cs2en.
12	0-0 0-1 2-2 3-3 3-4 4-5 5-6	GIZA++ alignments "back" for cs2en.
13	0-0 0-1 1-2 2-2 3-3 3-4 4-5 5-6	GIZA++ alignments symmetrized using grow-diag-final-and for cs2en.
14	0-0 0-1 1-2 2-2 3-3 3-4 4-5 5-6	GIZA++ alignments symmetrized using grow-diag-final-and for en2cs (not the inverse of column 13).
	Cross-Language Alignments Between T-Layer Czech and English Always indexed from 0, Czech-English.	
15	0-1 1-2 2-2 3-3 4-4	T-alignment "there" for cs2en.
16	0-0 0-1 2-2 3-3 4-4	T-alignment "back" for cs2en.
17		Additional rule-based t-alignment linking esp. generated nodes like `#Perspron;`.

Table 5: An example and explanation of the "export format" of CzEng 1.0. Each row in the table corresponds to one tab-delimited column of the line-oriented text files.

Žabokrtský. 2006. Annotation on the tectogrammatical level in the Prague Dependency Treebank. Annotation manual. Technical Report 30, ÚFAL MFF UK, Prague, Czech Rep.

Giang Linh Nguy, Václav Novák, and Zdeněk Žabokrtský. 2009. Comparison of Classification and Ranking Approaches to Pronominal Anaphora Resolution in Czech. In *Proceedings of the SIGDIAL 2009 Conference*, pages 276–285, London, UK, September. ACL.

Giang Linh Nguy. 2006. Návrh souboru pravidel pro analýzu anafor v českém jazyce. Master's thesis, MFF UK, Prague, Czech Republic. In Czech.

Franz Josef Och and Hermann Ney. 2000. A Comparison of Alignment Models for Statistical Machine Translation. In *Proceedings of the 17th conference on Computational linguistics*, pages 1086–1090. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: Modular NLP framework. In Hrafn Loftsson, Eirikur Rögnvaldsson, and Sigrun Helgadottir, editors, *Proceedings of the 7th International Conference on Advances in Natural Language Processing (IceTAL 2010)*, volume 6233 of *LNCS*, pages 293–304, Berlin / Heidelberg. Iceland Centre for Language Technology (ICLT), Springer.

Jan Ptáček and Zdeněk Žabokrtský. 2006. Synthesis of Czech Sentences from Tectogrammatical Trees. In *Text, Speech and Dialogue*, pages 221–228. Springer.

Drahomíra Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. 2007. The best of two worlds: cooperation of statistical and rule-based taggers for Czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*, ACL '07, pages 67–74, Stroudsburg, PA. Association for Computational Linguistics.

Dániel Varga, László Németh, Péter Halácsy, András Kornai, Viktor Trón, and Viktor Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of the Recent Advances in Natural Language Processing RANLP 2005*, pages 590–596, Borovets, Bulgaria.

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. TectoMT: Highly Modular Hybrid MT System with Tectogrammatics Used as Transfer Layer. In *Proc. of the ACL Workshop on Statistical Machine Translation*, pages 167–170, Columbus, Ohio, USA.

English-to-Czech Factored Machine Translation

Ondřej Bojar
Institute of Formal and Applied Linguistics
ÚFAL MFF UK, Malostranské náměstí 25
CZ-11800 Praha, Czech Republic
bojar@ufal.mff.cuni.cz

Abstract

This paper describes experiments with English-to-Czech phrase-based machine translation. Additional annotation of input and output tokens (multiple factors) is used to explicitly model morphology. We vary the translation scenario (the setup of multiple factors) and the amount of information in the morphological tags. Experimental results demonstrate significant improvement of translation quality in terms of BLEU.

1 Introduction

Statistical phrase-based machine translation (SMT) systems currently achieve top performing results.[1] Known limitations of phrase-based SMT include worse quality when translating to morphologically rich languages as opposed to translating from them (Koehn, 2005). One of the teams at the 2006 summer engineering workshop at Johns Hopkins University[2] attempted to tackle these problems by introducing separate FACTORS in SMT input and/or output to allow explicit modelling of the underlying language structure. The support for factored translation models was incorporated into the Moses open-source SMT system[3].

In this paper, we report on experiments with English-to-Czech multi-factor translation. After a brief overview of factored SMT and our data (Sections 2 and 3), we summarize some possible translating scenarios in Section 4. Section 5 studies the level of detail useful for morphological representation and Section 6 compares the results to a setting with more data available, albeit out of domain. The second part (Section 7) is devoted to a brief analysis of MT output errors.

1.1 Motivation for Improving Morphology

Czech is a Slavic language with very rich morphology and relatively free word order. The Czech morphological system (Hajič, 2004) defines 4,000 tags in theory and 2,000 were actually seen in a big tagged corpus. (For comparison, the English Penn Treebank tagset contains just about 50 tags.) In our parallel corpus (see Section 3 below), the English vocabulary size is 35k distinct token types but more than twice as big in Czech, 83k distinct token types.

To further emphasize the importance of morphology in MT to Czech, we compare the standard BLEU (Papineni et al., 2002) of a baseline phrase-based translation with BLEU which disregards word forms (lemmatized MT output is compared to lemmatized reference translation). The theoretical margin for improving MT quality is about 9 BLEU points: the same MT output scores 12 points in standard BLEU and 21 points in lemmatized BLEU.

2 Overview of Factored SMT

In statistical MT, the goal is to translate a source (foreign) language sentence $f_1^J = f_1 \dots f_j \dots f_J$ into a target language (Czech) sentence $c_1^I = c_1 \dots c_j \dots c_I$. In phrase-based SMT, the assumption is made that the target sentence can be constructed by segmenting source sentence into phrases, translating each phrase and finally composing the

[1] http://www.nist.gov/speech/tests/mt/
[2] http://www.clsp.jhu.edu/ws2006/
[3] http://www.statmt.org/moses/

Proceedings of the Second Workshop on Statistical Machine Translation, pages 232–239,
Prague, June 2007.

target sentence from phrase translations, s_1^K denotes the segmentation of the input sentence into K phrases. Among all possible target language sentences, we choose the sentence with the highest probability,

$$\hat{e}_1^{\hat{I}} = \underset{I,c_1^I,K,s_1^K}{\operatorname{argmax}} \{Pr(c_1^I|f_1^J, s_1^K)\} \quad (1)$$

In a log-linear model, the conditional probability of c_1^I being the translation of f_1^J under the segmentation s_1^K is modelled as a combination of independent feature functions $h_1(\cdot,\cdot,\cdot)\dots h_M(\cdot,\cdot,\cdot)$ describing the relation of the source and target sentences:

$$Pr(c_1^I|f_1^J, s_1^K) = \frac{\exp(\sum_{m=1}^M \lambda_m h_m(c_1^I, f_1^J, s_1^K))}{\sum_{c_1'^{I'}} \exp(\sum_{m=1}^M \lambda_m h_m(c_1'^{I'}, f_1^J, s_1^K))} \quad (2)$$

The denominator in 2 is used as a normalization factor that depends on the source sentence f_1^J and segmentation s_1^K only and is omitted during maximization. The model scaling factors λ_1^M are trained either to the maximum entropy principle or optimized with respect to the final translation quality measure.

Most of our features are phrase-based and we require all such features to operate synchronously on the segmentation s_1^K and independently of neighbouring segments. In other words, we restrict the form of phrase-based features to:

$$h_m(c_1^I, f_1^J, s_1^K) = \sum_{k=1}^K \tilde{h}_m(\tilde{c}_k, \tilde{f}_k) \quad (3)$$

where $\tilde{f}_k$ represents the source phrase and $\tilde{c}$ represents the target phrase k given the segmentation s_1^K.

2.1 Decoding Steps

In factored SMT, source and target words f and c are represented as tuples of F and C FACTORS, resp., each describing a different aspect of the word, e.g. its word form, lemma, morphological tag, role in a verbal frame. The process of translation consists of DECODING steps of two types: MAPPING steps and GENERATION steps. If more steps contribute to the same output factor, they have to agree on the outcome, i.e. partial hypotheses where two decoding steps produce conflicting values in an output factor are discarded.

A MAPPING step from a subset of source factors $S \subseteq \{1\dots F\}$ to a subset of target factors $T \subseteq \{1\dots C\}$ is the standard phrase-based model (see e.g. (Koehn, 2004a)) and introduces a feature in the following form:

$$\tilde{h}_m^{\text{map}:S\to T}(\tilde{c}_k, \tilde{f}_k) = \log p(\tilde{f}_k^S|\tilde{c}_k^T) \quad (4)$$

The conditional probability of $\tilde{f}_k^S$, i.e. the phrase $\tilde{f}_k$ restricted to factors S, given $\tilde{c}_k^T$, i.e. the phrase $\tilde{c}_k$ restricted to factors T is estimated from relative frequencies: $p(\tilde{f}_k^S|\tilde{c}_k^T) = N(\tilde{f}^S, \tilde{c}^T)/N(\tilde{c}^T)$ where $N(\tilde{f}^S, \tilde{c}^T)$ denotes the number of co-occurrences of a phrase pair $(\tilde{f}^S, \tilde{c}^T)$ that are consistent with the word alignment. The marginal count $N(\tilde{c}^T)$ is the number of occurrences of the target phrase $\tilde{c}^T$ in the training corpus.

For each mapping step, the model is included in the log-linear combination in source-to-target and target-to-source directions: $p(\tilde{f}^T|\tilde{c}^S)$ and $p(\tilde{c}^S|\tilde{f}^T)$. In addition, statistical single word based lexica are used in both directions. They are included to smooth the relative frequencies used as estimates of the phrase probabilities.

A GENERATION step maps a subset of target factors T_1 to a disjoint subset of target factors T_2, $T_{1,2} \subset \{1\dots C\}$. In the current implementation of Moses, generation steps are restricted to word-to-word correspondences:

$$\tilde{h}_m^{\text{gen}:T_1\to T_2}(\tilde{c}_k, \tilde{f}_k) = \log \prod_{i=1}^{\text{length}(\tilde{c}_k)} p(\tilde{c}_{k,i}^{T_1}|\tilde{c}_{k,i}^{T_2}) \quad (5)$$

where $\tilde{c}_{k,i}^T$ is the i-th words in the k-th target phrase restricted to factors T. We estimate the conditional probability $p(\tilde{c}_{k,i}^{T_2}|\tilde{c}_{k,i}^{T_1})$ by counting over words in the target-side corpus. Again, the conditional probability is included in the log-linear combination in both directions.

In addition to features for decoding steps, we include arbitrary number of target language models over subsets of target factors, $T \subseteq \{1\dots C\}$. Typically, we use the standard n-gram language model:

$$h^T_{\mathrm{LM}_n}(f_1^J, c_1^I) = \log \prod_{i=1}^{I} p(c_i^T | c_{i-1}^T \dots c_{i-n+1}^T) \quad (6)$$

While generation steps are used to enforce "vertical" coherence between "hidden properties" of output words, language models are used to enforce sequential coherence of the output.

Operationally, Moses performs a stack-based beam search very similar to Pharaoh (Koehn, 2004a). Thanks to the synchronous-phrases assumption, all the decoding steps can be performed during a preparatory phase. For each span in the input sentence, all possible translation options are constructed using the mapping and generation steps in a user-specified order. Low-scoring options are pruned already during this phase. Once all translation options are constructed, Moses picks source phrases (all output factors already filled in) in arbitrary order, subject to a reordering limit, producing output in left-to-right fashion and scoring it using the specified language models exactly as Pharaoh does.

3 Data Used

The experiments reported in this paper were carried out with the News Commentary (NC) corpus as made available for the SMT workshop[4] of the ACL 2007 conference.[5]

The Czech part of the corpus was tagged and lemmatized using the tool by Hajič and Hladká (1998), the English part was tagged MXPOST (Ratnaparkhi, 1996) and lemmatized using the Morpha tool (Minnen et al., 2001). After some final cleanup, the corpus consists of 55,676 pairs of sentences (1.1M Czech tokens and 1.2M English tokens). We use the designated additional tuning and evaluation sections consisting of 1023, resp. 964 sentences.

In all experiments, word alignment was obtained using the grow-diag-final heuristic for symmetrizing GIZA++ (Och and Ney, 2003) alignments. To reduce data sparseness, the English text was lowercased and Czech was lemmatized for alignment estimation. Language models are based on the target side of the parallel corpus only, unless stated otherwise.

3.1 Evaluation Measure and MERT

We evaluate our experiments using the (lowercase, tokenized) BLEU metric and estimate the empirical confidence using the bootstrapping method described in Koehn (2004b).[6] We report the scores obtained on the test section with model parameters tuned using the tuning section for minimum error rate training (MERT, (Och, 2003)).

4 Scenarios of Factored Translation English→Czech

We experimented with the following factored translation scenarios.

The baseline scenario (labelled T for translation) is single-factored: input (English) lowercase word forms are directly translated to target (Czech) lowercase forms. A 3-gram language model (or more models based on various corpora) checks the stream of output word forms. The baseline scenario thus corresponds to a plain phrase-based SMT system:

English		Czech	
lowercase	⟶	lowercase	+LM
lemma		lemma	
morphology		morphology	

In order to check the output not only for word-level coherence but also for morphological coherence, we add a single generation step: input word forms are first translated to output word forms and each output word form then generates its morphological tag.

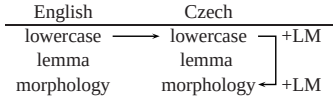
Two types of language models can be used simultaneously: a (3-gram) LM over word forms and a (7-gram) LM over morphological tags.

We used tags with various levels of detail, see section 5. We call this the "T+C" (translate and check) scenario:

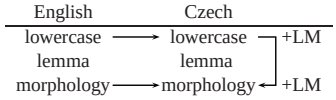
[4] http://www.statmt.org/wmt07/

[5] Our preliminary experiments with the Prague Czech-English Dependency Treebank, PCEDT v.1.0 (Čmejrek et al., 2004), 20k sentences, gave similar results, although with a lower level of significance due to a smaller evaluation set.

[6] Given a test set of sentences, we perform 1,000 random selections with repetitions to estimate 1,000 BLEU scores on test sets of the same size. The empirical 90%-confidence upper and lower bounds are obtained after removing top and bottom 5% of scores. For conciseness, we report the average of the distance between to standard BLEU value and the empirical upper and lower bound after the "±" symbol.

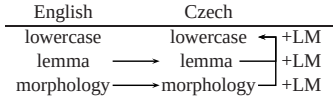


As a refinement of T+C, we also used T+T+C scenario, where the morphological output stream is constructed based on both output word forms and input morphology. This setting should reinforce correct translation of morphological features such as number of source noun phrases. To reduce the risk of early pruning, the generation step operationally precedes the morphology mapping step. Again, two types of language models can be used in this "T+T+C" scenario:



The most complex scenario we used is linguistically appealing: output lemmas (base forms) and morphological tags are generated from input in two independent translation steps and combined in a single generation step to produce output word forms. The input English text was not lemmatized so we used English word forms as the source for producing Czech lemmas.

The "T+T+G" setting allows us to use three types of language models. Trigram models are used for word forms and lemmas and 7-gram language models are used over tags:



4.1 Experimental Results: Improved over T

Table 1 summarizes estimated translation quality of the various scenarios. In all cases, a 3-gram LM is used for word forms or lemmas and a 7-gram LM for morphological tags.

The good news is that multi-factored models always outperform the baseline T.

Unfortunately, the more complex multi-factored scenarios do not bring any significant improvement over T+C. Our belief is that this effect is caused by search errors: with multi-factored models, more hypotheses get similar scores and future costs of partial hypotheses might be estimated less reliably. With the limited stack size (not more than 200 hypotheses of the same number of covered input words), the decoder may more often find sub-optimal solutions. Moreover, the more steps are used, the more model weights have to be tuned in the minimum error rate training. Considerably more tuning data might be necessary to tune the weights reliably.

	BLEU
T+T+G	13.9±0.7
T+T+C	13.9±0.6
T+C	13.6±0.6
Baseline: T	12.9±0.6

Table 1: BLEU scores of various translation scenarios.

5 Granularity of Czech Part-of-Speech

As stated above, the Czech morphological tag system is very complex: in theory up to 4,000 different tags are possible. In our T+T+C scenario, we experiment with various simplifications of the system to find the best balance between richness and robustness of the statistics available in our corpus. (The more information is retained in the tags, the more severe data sparseness is.)

Full tags (1200 unique seen in the 56k corpus): Full Czech positional tags are used. A tag consists of 15 positions, each holding the value of a morphological property (e.g. number, case or gender).[7]

POS+case (184 unique seen): We simplify the tag to include only part and subpart of speech (distinguishes also partially e.g. verb tenses). For nouns, pronouns, adjectives and prepositions[8], also the case is included.

CNG01 (621 unique seen): CNG01 refines POS. For nouns, pronouns and adjectives we include not only the case but also number and gender.

[7] In principle, each of the 15 positions could be used as a separate factor. The set of necessary generation steps to encode relevant dependencies would have to be carefully determined.

[8] Some Czech prepositions select for a particular case, some are ambiguous. Although the case is never shown on surface of the preposition, the tagset includes this information and Czech taggers are able to infer the case.

CNG02 (791 unique seen): Tag for punctuation is refined: the lemma of the punctuation symbol is taken into account; previous models disregarded e.g. the distributional differences between a comma and a question mark. Case, number and gender added to nouns, pronouns, adjectives, prepositions, but also to verbs and numerals (where applicable).

CNG03 (1017 unique seen): Optimized tagset:

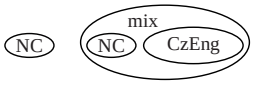
- Tags for nouns, adjectives, pronouns and numerals describe the case, number and gender; the Czech reflexive pronoun *se* or *si* is highlighted by a special flag.
- Tag for verbs describes subpart of speech, number, gender, tense and aspect; the tag includes a special flag if the verb was the auxiliary verb *být (to be)* in any of its forms.
- Tag for prepositions includes the case and also the lemma of the preposition.
- Lemma included for punctuation, particles and interjections.
- Tag for numbers describes the "shape" of the number (all digits are replaced by the digit *5* but number-internal punctuation is kept intact). The tag thus distinguishes between 4- or 5-digit numbers or the precision of floating point numbers.
- Part of speech and subpart of speech for all other words.

5.1 Experimental Results: CNG03 Best

Table 2 summarizes the results of T+T+C scenario with varying detail in morphological tag.

	BLEU
Baseline: T (single-factor)	12.9±0.6
T+T+C, POS+case	13.2±0.6
T+T+C, CNG01	13.4±0.6
T+T+C, CNG02	13.5±0.7
T+T+C, full tags	13.9±0.6
T+T+C, CNG03	14.2±0.7

Table 2: BLEU scores of various granularities of morphological tags in T+T+C scenario.



$$\text{weighted} = \alpha\text{NC} + \beta\text{mix}$$

Scenario	Phrases from	LMs	BLEU
T	NC	NC	12.9±0.6
T	mix	mix	11.8±0.6
T	mix	weighted	11.8±0.6
T+C CNG03	NC	NC	13.7±0.7
T+C CNG03	mix	mix	13.1±0.7
T+C CNG03	mix	weighted	13.7±0.7
T+C full tags	NC	NC	13.6±0.6
T+C full tags	mix	mix	13.1±0.7
T+C full tags	mix	weighted	13.8±0.7

Figure 1: The effect of additional data in T and T+C scenarios.

Our results confirm improvement over the single-factored baseline. Detailed knowledge of the morphological system also proves its utility: by choosing the most relevant features of tags and lemmas but avoiding sparseness, we can improve on BLEU score by about 0.3 absolute over T+T+C with full tags.

6 More Out-of-Domain Data in T and T+C Scenarios

In order to check if the method scales up with more parallel data available, we extend our training data using the CzEng parallel corpus (Bojar and Žabokrtský, 2006). CzEng contains sentence-aligned texts from the European Parliament (about 75%), e-books and stories (15%) and open source documentation. By "Baseline" corpus we denote NC corpus only, by "Large" we denote the combination of training sentences from NC and CzEng (1070k sentences, 13.9M Czech and 15.5 English tokens) where in-domain NC data amounts only to 5.2% sentences.

Figure 1 gives full details of our experiments with the additional data. We varied the scenario (T or T+C), the level of detail in the T+C scenario (full tags vs. CNG03) and the size of the training corpus. We extract phrases from either the in-domain corpus only (NC) or the mixed corpus (mix). We use either one LM per output factor, varying the corpus size (NC or mix), or two LMs per output factors with weights trained independently in the MERT proce-

dure (weighted). Independent weights allow us to take domain difference into account, but we exploit this in the target LM only, not the phrases.

The only significant difference is caused by the scenario: T+C outperforms the baseline T, regardless of corpus size. Other results (insignificantly) indicate the following observations:

- Ignoring the domain difference and using only the mixed domain LM in general performs worse than allowing MERT to optimize LM weights for in-domain and generic data separately.[9]
- CNG03 outperforms full tags only in small data setting, with large data (treating the domain difference properly), full tags perform better.

7 Untreated Morphological Errors

The previous sections described improvements gained on small data sets when checking morphological agreement using T+T+C scenario (BLEU raised from 12.9% to 13.9% or up to 14.2% with manually tuned tagset, CNG03). However, the best result achieved is still far below the margin of lemmatized BLEU (21%), as mentioned in Section 1.1.

When we searched for the unexploited morphological errors, visual inspection of MT output suggested that local agreement (within 3-word span) is relatively correct but Verb-Modifier relations are often malformed causing e.g. a bad case for the Modifier. To quantify this observation we performed a micro-study of our best MT output using an intuitive metric. We checked whether Verb-Modifier relations are properly preserved during the translation of 15 sample sentences.

The *source* text of the sample sentences contained 77 Verb-Modifier pairs. Table 3 lists our observations on the two members in each Verb-Modifier pair. We see that only 56% of verbs are translated correctly and 79% of nouns are translated correctly. The system tends to skip verbs quite often (27% of cases).

[9] In our previous experiments with PCEDT as the domain-specific data, the difference was more apparent because the corpus domains were more distant. In the T scenario reported here, the weighted LMs did not bring any improvement over "mix" and even performed worse than the baseline NC. We attribute this effect to some randomness in the MERT procedure.

Translation of	**Verb**	**Modifier**
... preserves meaning	56%	79%
... is disrupted	14%	12%
... is missing	27%	1%
... is unknown (not translated)	0%	5%

Table 3: Analysis of 77 Verb-Modifier pairs in 15 sample sentences.

More importantly, our analysis has shown that even in cases where both the Verb and the Modifier are lexically correct, the relation between them in Czech is either non-grammatical or meaning-disrupted in 56% of these cases. Commented samples of such errors are given in Figure 2 below. The first sample shows that a strong language model can lead to the choice of a grammatical relation that nevertheless does not convey the original meaning. The second sample illustrates a situation where two correct options are available but the system chooses an inappropriate relation, most probably because of backing off to a generic pattern verb-noun$^{accusative}_{plural}$. This pattern is quite common for expressing the object role of many verbs (such as *vydat*, see Correct option 2 in Figure 2), but does not fit well with the verb *vyběhnout*. While the target-side data may be rich enough to learn the generalization vyběhnout–s–*instr*, no such generalization is possible with language models over word forms or morphological tags only. The target side data will be hardly ever rich enough to learn this particular structure in all correct morphological and lexical variants: *vyběhl–s–reklamou*, *vyběhla–s–reklamami*, *vyběhl–s–prohlášením*, *vyběhli–s–oznámením*, ... We would need a mixed model that combines verb lemmas, prepositions and case information to properly capture the relations.

Unfortunately, our preliminary experiments that made use of automatic Czech dependency parse trees to construct a factor explicitly highlighting the Verb (lexicalized) its Modifiers (case and the lemma of the preposition, if present) and boundary symbols such as punctuation or conjunctions and using a dummy token for all other words did not bring any improvement over the baseline. A possible reason is that we employed only a standard 7-gram language model to this factor. A more appropriate treatment

is to disregard the dummy tokens in the language model at all and use an n-gram language model that looks at last $n-1$ non-dummy items.

8 Related Research

Class-based LMs (Brown et al., 1992) or factored LMs (Bilmes and Kirchhoff, 2003) are very similar to our T+C scenario. Given the small differences in all T+... scenarios' performance, class-based LM might bring equivalent improvement. Yang and Kirchhoff (2006) have recently documented minor BLEU improvement using factored LMs in single-factored SMT to English. The multi-factored approach to SMT of Moses is however more general.

Many researchers have tried to employ morphology in improving word alignment techniques (e.g. (Popović and Ney, 2004)) or machine translation quality (Nießen and Ney (2001), Koehn and Knight (2003), Zollmann et al. (2006), among others, for various languages; Goldwater and McClosky (2005), Bojar et al. (2006) and Talbot and Osborne (2006) for Czech), however, they focus on translating *from* the highly inflectional language.

Durgar El-Kahlout and Oflazer (2006) report preliminary experiments in English to Turkish single-factored phrase-based translation, gaining significant improvements by splitting root words and their morphemes into a sequence of tokens. In might be interesting to explore multi-factored scenarios for different Turkish morphology representation suggested the paper.

de Gispert et al. (2005) generalize over verb forms and generate phrase translations even for unseen target verb forms. The T+T+G scenario allows a similar extension if the described generation step is replaced by a (probabilistic) morphological generator.

Nguyen and Shimazu (2006) translate from English to Vietnamese but the morphological richness of Vietnamese is comparable to English. In fact the Vietnamese vocabulary size is even smaller than English vocabulary size in one of their corpora. The observed improvement due to explicit modelling of morphology might not scale up beyond small-data setting.

As an alternative option to our verb-modifier experiments, structured language models (Chelba and Jelinek, 1998) might be considered to improve clause coherence, until full-featured syntax-based MT models (Yamada and Knight (2002), Eisner (2003), Chiang (2005) among many others) are tested when translating to morphologically rich languages.

9 Conclusion

We experimented with multi-factored phrase-based translation aimed at improving morphological coherence in MT output. We varied the setup of additional factors (translation scenario) and the level of detail in morphological tags. Our results on English-to-Czech translation demonstrate significant improvement in BLEU scores by explicit modelling of morphology and using a separate morphological language model to ensure the coherence. To our knowledge, this is one of the first experiments showing the advantages of using multiple factors in MT.

Verb-modifier errors have been studied and a factor capturing verb-modifier dependencies has been proposed. Unfortunately, this factor has yet to bring any improvement.

10 Acknowledgement

The work on this project was partially supported by the grants Collegium Informaticum GAČR 201/05/H014, grants No. ME838 and GA405/06/0589 (PIRE), FP6-IST-5-034291-STP (Euromatrix), and NSF No. 0530118.

References

Jeff A. Bilmes and Katrin Kirchhoff. 2003. Factored language models and generalized parallel backoff. In *Proc. of NAACL 2003*, pages 4–6.

Ondřej Bojar and Zdeněk Žabokrtský. 2006. CzEng: Czech-English Parallel Corpus, Release version 0.5. *Prague Bulletin of Mathematical Linguistics*, 86:59–62.

Ondřej Bojar, Evgeny Matusov, and Hermann Ney. 2006. Czech-English Phrase-Based Machine Translation. In *Proc. of FinTAL 2006*, pages 214–224, Turku, Finland.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Ciprian Chelba and Frederick Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proc. of ACL 1998*, pages 225–231, San Francisco, California.

Input:	Keep on investing.
MT output:	Pokračovalo investování. (grammar correct here!)
Gloss:	Continued investing. (Meaning: The investing continued.)
Correct:	Pokračujte v investování.

Input:	brokerage	firms	rushed out	ads ...
MT Output:	brokerské	firmy	vyběhl	reklamy
Gloss:	brokerage	firms$_{pl.fem}$	ran$_{sg.masc}$	ads$^{pl.voc,sg.gen}_{pl.nom,pl.acc}$
Correct option 1:	brokerské	firmy	vyběhly	s reklamami$_{pl.instr}$
Correct option 2:	brokerské	firmy	vydaly	reklamy$_{pl.acc}$

Figure 2: Two sample errors in translating Verb-Modifier relation from English to Czech.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proc. of ACL 2005*, pages 263–270.

Martin Čmejrek, Jan Cuřín, Jiří Havelka, Jan Hajič, and Vladislav Kuboň. 2004. Prague Czech-English Dependecy Treebank: Syntactically Annotated Resources for Machine Translation. In *Proc. of LREC 2004*, Lisbon, Portugal.

Adrià de Gispert, José B. Mariño, and Josep M. Crego. 2005. Improving statistical machine translation by classifying and generalizing inflected verb forms. In *Proc. of Eurospeech 2005*, pages 3185–3188, Lisbon, Portugal.

İlknur Durgar El-Kahlout and Kemal Oflazer. 2006. Initial Explorations in English to Turkish Statistical Machine Translation. In *Proc. of the Workshop on Statistical Machine Translation, ACL 2006*, pages 7–14, New York City.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proc. of ACL 2003, Companion Volume*, pages 205–208, Sapporo, Japan.

Sharon Goldwater and David McClosky. 2005. Improving statistical MT through morphological analysis. In *Proc. of HLT/EMNLP 2005*, pages 676–683.

Jan Hajič and Barbora Hladká. 1998. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In *Proc. of COLING/ACL 1998*, pages 483–490, Montreal, Canada.

Jan Hajič. 2004. *Disambiguation of Rich Inflection (Computational Morphology of Czech)*. Nakladatelství Karolinum, Prague.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proc. of EACL 2003*, pages 187–193.

Philipp Koehn. 2004a. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *Proc. of AMTA 2004*, pages 115–124.

Philipp Koehn. 2004b. Statistical Significance Tests for Machine Translation Evaluation. In *Proc. of EMNLP 2004*, Barcelona, Spain.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of MT Summit X*.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

T.P. Nguyen and A. Shimazu. 2006. Improving Phrase-Based SMT with Morpho-Syntactic Analysis and Transformation. In *Proc. of AMTA 2006*, pages 138–147.

Sonja Nießen and Hermann Ney. 2001. Toward hierarchical models for statistical machine translation of inflected languages. In *Proc. of Workshop on Data-driven methods in machine translation, ACL 2001*, pages 1–8.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of ACL 2003*, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL 2002*, pages 311–318.

M. Popović and H. Ney. 2004. Improving Word Alignment Quality using Morpho-Syntactic Information. In *Proc. of COLING 2004*, Geneva, Switzerland.

Adwait Ratnaparkhi. 1996. A Maximum Entropy Part-Of-Speech Tagger. In *Proc. of EMNLP 1996*, Philadelphia, USA.

David Talbot and Miles Osborne. 2006. Modelling lexical redundancy for machine translation. In *Proc. of COLING and ACL 2006*, pages 969–976, Sydney, Australia.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical MT. In *Proc. of ACL 2002*, pages 303–310.

Mei Yang and Katrin Kirchhoff. 2006. Phrase-based backoff models for machine translation of highly inflected languages. In *Proc. of EACL 2006*.

Andreas Zollmann, Ashish Venugopal, and Stephan Vogel. 2006. Bridging the inflection morphology gap for arabic statistical machine translation. In *Proc. of HLT/NAACL*.

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 99 APRIL 2013 39–58

The Design of Eman, an Experiment Manager

Ondřej Bojar, Aleš Tamchyna

Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

Abstract

We present `eman`, a tool for managing large numbers of computational experiments. Over the years of our research in machine translation (MT), we have collected a couple of ideas for efficient experimenting. We believe these ideas are generally applicable in (computational) research of any field. We incorporated them into `eman` in order to make them available in a command-line Unix environment.

The aim of this article is to highlight the core of the many ideas. We hope the text can serve as a collection of experiment management tips and tricks for anyone, regardless their field of study or computer platform they use. The specific examples we provide in `eman`'s current syntax are less important but they allow us to use concrete terms. The article thus also fills the gap in `eman` documentation by providing some high-level overview.

1. Introduction

Computational sciences including computational linguistics and computer science require broad experimenting to support theories and evaluate various techniques or methods. Very often, even the authors of some novel idea cannot guess the best possible method parameters and some form of search for them is desirable. This becomes more apparent if the method combines several independent modules or processing steps, each of which may or may not have been evaluated independently of the overall goal.

Another common aspect of natural sciences is the overarching strive for reproducibility. A novel method is never completely trusted until validated by a few independent laboratories, a program has to be tested and evaluated on a range of inputs and so on.

We pinpoint these two aspects of science by noting that: research = **re**producible **search**.

 Corresponding author: bojar@ufal.mff.cuni.cz
Cite as: Ondřej Bojar, Aleš Tamchyna. The Design of Eman, an Experiment Manager. The Prague Bulletin of Mathematical Linguistics No. 99, 2013, pp. 39–58. doi: 10.2478/pralin-2013-0003.

In this article, we describe a very general tool that facilitates both reproducibility and search for the best configuration and parameters of complex experimental pipelines. Our `eman` also supports the collaboration of several people on the experiment.

`Eman` is open-source software, freely available for both non-commercial and commercial use.[1] The most recent version of the tool as well as other documentation is accessible at:

`http://ufal.mff.cuni.cz/eman`

The article is structured as follows: in Section 2, we explain what we perceive as the state-of-the-art techniques in efficient experimenting, highlighting the design goals of our new tool. Section 3 introduces our terminology and the basic building blocks of experiments in `eman`'s terms. Section 4 summarizes the first area of `eman`'s utility: navigation in the space of steps and experiments. Section 5 is devoted to the idea of cloning experiments and Section 6 describes the third key contribution: a general technique for collecting and interpreting the results. We conclude by introducing `eman`'s support for teamwork (Section 7), related tools (Section 8) and our future plans (Section 9).

While we show some calls of `eman` commands in their exact syntax, the main goal of this article is to describe the underlying general ideas, not to serve as a reference guide for the tool. For this, the user is advised to the manual page of `eman` which can be obtained by running:

```
eman --man
```

2. Design Objectives

The design of `eman` builds on our experience that the following features of experimentation environment are essential:

Reuse of results. In order to save both computation time and disk space, we need to reuse as many intermediate results as possible.

Encapsulation. Scientific experiments usually consist of complex sequences of processing steps, each carried out using a different tool that itself often needs some analysis, debugging, tweaking or optimization. To simplify switching and keeping focus, `eman` promotes encapsulation of each logical step into a separate directory. This directory should be as self-contained as reasonable, so when the researcher later inspects it, all the inputs and outputs are in one place.

Detailed records. Detailed logging of program outputs as well as of commands issued is essential for ensuring reproducibility, debugging and analysis of errors and comparison of results. We extend this to recording also the exact versions of (third-party) tools used in the experiment and also the procedure needed to

[1] `Eman` is licensed under the Creative Commons Attribution-ShareAlike License 3.0 (CC-BY-SA).

obtain and install the tools. This is achieved by treating the (source) code of the tools as input data of the experiment and including the compilation of the tools in the pipeline of the experiment. The reuse of intermediate results ensures the code is compiled only once.

Immutability. To simplify the record keeping, we opt for immutability of all data that is created in the experiment. Whenever some intermediate result is created based on some settings, `eman` never changes it. Modifications of the run are of course possible, but they always obtain a new identifier and reside in a new directory.

Hacking welcome. Admittedly, research prototype software is often quickly patched and far from anything that could be called a stable release. Furthermore, and this is a more important issue, research software does not always fit the purpose in new experiments. It is thus common that the tools have to be adapted or that a manual intervention is necessary after a random unexpected failure. `Eman` introduces a great deal of flexibility of experiment design – experiments are composed of individual steps which are further split into several lifetime stages – to allow for such an intervention.

Cloning. Research partially comprises of examining a range of minor modifications of a setup. In `eman`'s view, as it will be described below, experiments are defined by arbitrary variables and such setup modifications usually amount to setting these variables differently. Section 5 provides examples of one-line commands that take an existing experiment and apply a given set of modifications to it (such as setting a parameter differently or reversing the source and target language in an MT experiment). Finally, the necessary minimum of new processing steps are created and launched, reusing the steps common to both setups.

Cloning is in fact such a powerful idea that the relatively simple implementation of it in `eman` (regular expressions applied to experiment configuration files) allowed to create the tool Prospector, an automatic researcher (Tamchyna and Bojar, 2013). Prospector automatically searches the "space of possible MT systems" by evaluating various settings specified by its configuration file. The search can be guided by any metric, e.g. the well-known BLEU (Papineni et al., 2002) as calculated in the final evaluation step. Several search algorithms are implemented (greedy, exhaustive, genetic, random).

Prospector allows researches to avoid the tedious work of e.g. finding optimal parameters or meta-parameters for the MT decoder (beam size etc.) or any other experimental settings. It is freely available and distributed along with `eman`.

Parallelism. The parallelizations common in contemporary computer science (multiple processor cores, clusters of computers) allow for parallel execution of experiments. This is highly desirable because each individual experiment often takes a long time. Carrying out experiments in a strictly serial order would waste researchers' time and not fully exploit the available computational resources.

On the other hand, the researcher can easily lose track and focus when running many experiments in parallel.

`Eman` naturally allows to submit individual processing steps to a computer cluster, but more importantly, `eman` is designed to simplify the orientation in the large number of experiments already performed or in execution (see Section 4) and to some extent also the foreseen ones (see Section 6.3). The design also allows to derive (clone) new experiments from old ones even before the old ones complete.

Collaboration. The most recent feature of `eman` is the support for distributed experimenting. Currently we require a common filesystem (such as NFS), but that is reasonably easy to set up even across large distances. Individual processing steps can be launched by different researchers at different sites. The simple command "`eman add-remote`" issued once allows to include all the partial results of a remote site in the local environment. Circular inclusion is permitted allowing multiple researchers to "work at a common desk", reusing other people's processing steps (not just the programs but also the outputs of their particular runs), or to reinterpret their results (e.g. by creating new tabular views).

The same mechanism can be beneficial even for a single researcher as it allows to strictly separate some core source data (such as multiple training sets that nevertheless needed some preparation) from different branches of experiments.

Succinct notation. Shortcuts and abbreviations are very useful for improving the efficiency of the operating researcher. `Eman` provides shortcuts at several occasions, which is very useful e.g. for checking the status of the experiments over SSH in the cell phone.

3. Seeds, Steps, Experiments

Each *experiment* consists of atomic tasks called *steps*. In the context of MT, steps correspond e.g. to training a language model, translating a test set or running tuning. The individual steps depend on each other – the experiment is then a DAG (directed acyclic graph) of steps.

Each step has a type such as `tm` (translation model) or `translate`. The code which is executed when the step is run is generated by the corresponding *seed*. In the terminology of object-oriented programming (OOP), seeds can be viewed as classes and steps as their instances. Unlike in OOP, `eman`'s positive stance to hacking allows different steps (instances) of the same type (class) run code customized arbitrarily, not just using proper subclassing.

In our particular implementation of `eman`, each step is simply a directory named using the pattern "`s.steptype.abcHASH.20121215-1234`" where the date and the hash value make the name unique. Seeds are then simply programs (in any language of the researcher's choice) which interpret some Unix environment variables and generate

```
+- s.compress.370c2483.20121108-1216
|  | CMD=bzip2
|  | CMDARGS=
|  | DATASTEP=s.data.aaf8c8b1.20121108-1149
|  +- s.data.aaf8c8b1.20121108-1149
|  |  | CMD="cat ../binary.test"
|  |  | SIZE=10000
|  |  | TYPE=binary
```

Figure 1. Example of an eman traceback.

executable code (again, in any language). The code is stored in the step directory and later run (once all predecessors are ready and the step is started).

`Eman` is used in a directory called *playground* – all steps are created there, based on seeds in the subdirectory `eman.seeds`. The "`eman add-remote`" allows to link remote playgrounds to the current one. By adding a remote playground, the directory structure is not changed but `eman` suddenly knows about steps coming from the remote playgrounds, it can show their properties and include them in local experiments.

3.1. A Sample Experiment

For illustration, we implemented a "compression playground" which provides an environment for evaluating compression algorithms. This sample playground contains two seeds:

data Imports data into the playground – the data can be generated by any command (specified by the variable `CMD`) and the user can limit the amount of data using the variable `SIZE`.

compress Given some data, compress it using the command given in the variable `CMD` with some optional `CMDARGS` and calculate the compression rate.

Figure 1 shows an example of an experiment in this playground in `eman`'s format. This *traceback* is a full definition of the experiment. The seeds were "instantiated" to steps with some variable values (e.g. the compression command is `bzip2`) and connected to form a DAG – note that the dependency is explicitly captured in the variable `DATASTEP`.

3.2. Lifetime of a Step

Figure 2 depicts the lifetime of a step.

New steps are created using "`eman init STEPTYPE`". `Eman` creates a new directory in the current playground and copies the corresponding seed into it. Then the seed is executed – at this stage, the seed only performs basic sanity checks to determine

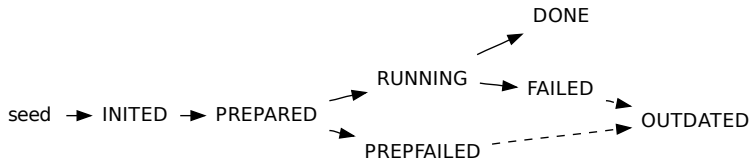


Figure 2. The lifetime of a processing step in eman

whether all required variables are defined etc. If everything succeeds, the step is registered in `eman` and receives the status `INITED`.

The seed is executed once more when the user runs "`eman prepare SPEC`". At this point, the seed creates an executable file `eman.command` which contains the step code. `Eman` checks whether the file was created and sets the step status to `PREPARED`.

Finally, the user runs "`eman start SPEC`" and the step is started. Its status changes to `RUNNING`. Once the step terminates, its status is either `DONE` or `FAILED`.

Steps at any stage, including the `FAILED` ones, can still serve as a basis for creating new steps with the same or similar variables, see below. For the purposes of marking that the user has already handled a failure, one more state, `OUTDATED`, was introduced. Upon request ("`--outdate`") `eman` not only creates a new instance of a failed step but also moves the failed one to the outdated state.

There are many shortcuts for user convenience – "`eman start`" on an `INITED` step automatically runs "`eman prepare`". The whole process of creating and running a step can even be done in one command (and it often is): "`eman init --start`".

The acyclicity of the lifetime diagram (no directed loops) is in line with the design objectives of immutability and detailed records. One should want to keep the logs of a failure and redo the job in a fresh instance of the step. (Indeed, this is what the command "`eman redo`" does, see Section 5.2.) In practice, a step can be very costly and fail at some late stage of execution. It would be wasteful to rerun it from scratch. The fact that each step includes its code (the file "`eman.command`", cf. the encapsulation objective) allows to manually fix this code and to jump to a recorded point shortly before the failure. After these manual changes in "`eman.command`", calling "`eman continue`" puts the failed step back to the `RUNNING` state and submits it to the cluster again (or runs it locally, depending on the user's environment).

3.3. Motivation for the Three Stages

What are the benefits of breaking the execution of a single processing step into the stages of initialization, preparation and the run itself?

The *initialization* is vital, it turns a blank directory into a valid `eman` step with variables defined. From this point on, the step can be incorporated into complex experiments and it can be used as a basis for cloning. There is thus no need to wait until the step finishes, we can plan ahead (and even submit for execution) other steps that will build on the future outputs.

After the initialization, the user has a chance to tweak the seed (and thus influence the actual command that will be performed). This is the point where we depart the OOP by allowing different instances run customized code.

The init phase should be very quick, it is run interactively and often repeated for many steps when cloning whole experiments.

The *preparation* phase is thus meant to get all input data in place, so that the user can check them before the actual computation, e.g. submitting the step to the cluster. In our MT experiments, the preparation phase was originally responsible for things like cutting a given subsection or subset of annotation features from the training data. As our training data grew, running these filters during the (still interactive) preparatory phase became inconvenient, so we changed our seeds and shifted even the obtaining of input data into the actual run. For `eman`, this makes no difference. The choice what happens at what phase is entirely up to the user; any of the phases can be even empty.

3.4. From Steps to Experiments

Steps are combined to form experiments. There is no pre-defined interface for communication between steps. Each step has access to its (direct) predecessors via variables – it can extract whatever files the previous step has created from its directory. The same step can thus serve several purposes at once, it just needs to produce outputs relevant to the respective successors.

The initial setup of experiments is somewhat tedious, the user has to run a sequence of commands like:

```
SOURCEDATASTEP=s.mydata.12345678.20121215-1234 eman init myprocessor
```

The user has to manually set the variable `SOURCEDATASTEP` to the name of the previously initialized `mydata` step. Once the full cascade of steps, i.e. an experiment, is set up, it is easier to derive variations of it using cloning, see Section 5.

3.5. Referring to Steps vs. Experiments

Note that the pointer to a step directory "`s.steptype.123`" can mean either just the single step that was carried out in the directory, or the whole experiment, i.e. the directed acyclic structure of steps that culminates with the given step. These two notions should not be confused.

It is the particular `eman` command that resolves the ambiguity between a step and an experiment. So for instance, "`eman prepare`" prepares an individual step regard-

less the status of its predecessors. Depending on what a particular step requires, the preparation may fail because the predecessors are still in the `INITED` state only and do not provide relevant data. The command "`eman start`" is more useful as it operates on the whole *experiment*. In other words, it ensures that the whole DAG of steps is first `PREPARED` and then submits all steps that were not finished yet to the cluster, introducing any necessary job dependencies.

4. Navigation in the Playground of Steps and Experiments

As the user creates experiments or derives clones of them (Section 5), the playground becomes quickly filled with step directories of unhelpful names like "`s.tm.1a53fg63.201`. Finding a particular step can then be difficult and time-consuming. This section briefly summarizes the four main techniques `eman` provides to ease the navigation in the playground: listing details of individual steps, finding (selecting) steps with given properties, examining the structure of experiments, i.e. how steps depend on one another, and manually tagging steps.

One more aspect of playground structure remains to be harnessed in a future version of `eman`: the history how steps were derived from other steps. Currently, `eman` only records the immediate origin of a derived step in the file "`eman.derived_from`".

4.1. Listing Steps and Their Details

The command "`eman ls`" prints steps in the current playground. The user can filter the listing based on the step type and request additional information – most importantly step variables, status and tags (see Section 4.4) – using command-line options. The following example query returns all steps of the type "`align`" and prints their variables, status and disk usage:

```
eman ls align --vars --stat --dus
```

Some shortcuts are again provided by means of commands `eman vars`, `stat` and `tags` that print the required information about all steps, or a particular step:

```
eman vars s.tm.1a5
```

Note that it is not necessary to specify the full step name, any part of it (not necessarily the beginning) long enough to make it unique within the playground is sufficient.

4.2. Finding Existing Steps

The command "`eman select`" (optionally abbreviated to "`sel`") provides a flexible means for finding steps with specified properties. The following few examples are just a brief demonstration of the query language.

- Steps which were created today and failed:

```
eman sel today f
```

- Last (most recent) five steps of the type "`align`":
  ```
  eman sel t align l 5
  ```
- Language model steps (i.e. steps of the type "`lm`") trained on word lemmas (`vre` stands for "a **v**ariable matches **r**egular **e**xpression"):
  ```
  eman sel t lm vre lemma
  ```
- Language models of order other than three (note the word `not`):
  ```
  eman sel t lm not vre ORDER=3
  ```
- MERT (Och, 2003) steps with a (possibly indirect) predecessor of the type `align` whose variables match the expression "lemma" (presumably a word alignment step done on word lemmas; `br` stands for backward recursion and matches properties in preceding steps):
  ```
  eman sel t mert br t align vre lemma
  ```
- Translation model ("`tm`") steps which were evaluated on a given test set (`fr` means forward recursion):
  ```
  eman sel t tm fr vre TESTCORP=wmt12
  ```

The syntax is very succinct which allows to write complex queries with very little effort. Users of `eman` frequently log in to their cluster using cell-phone SSH and type simply "`eman sel f`" to see if any experiments need their attention.

4.3. Dependencies and Users of Steps

`Eman` provides commands to list predecessors and successors of steps. Direct predecessors (dependencies) can be obtained using the command "`eman deps`" while "`eman traceback`" or "`eman tb`" prints the full *traceback* (i.e. DAG) of steps.

`Eman` assumes that an experiment is defined by the structure of dependencies and the values of their variables; this implies that the command "`eman tb --vars FINAL-STEP`" outputs a full, unambiguous specification of the whole experiment.

An example of a traceback with variables was already given in Figure 1.

Analogously to the predecessors, direct and indirect successors can be listed using the commands "`eman users`" and "`eman tf`" (traceforward), respectively.

4.4. Tagging of Steps

Steps that are somehow special or often referred to, e.g. because they were manually tweaked before submission or because they represent the baseline or the current best result, can be "tagged". Tags are simple labels associated with a particular step.

Tags are assigned to steps using the command "`eman add-tag`":

```
eman add-tag BASELINE s.evaluator.123456
```

Later, tags can be used as step identifiers as long as they are unambiguous. So we can e.g. double check what was our baseline configuration:

```
eman tb --vars BASELINE
```

The tags are stored in the step directory in the file `eman.tags`. Upon re-tagging ("`eman retag`"), the labels are recursively propagated to the step successors (however their `eman.tags` files do not change, the propagation is done in `eman`'s internal index only). While this is useful for organizing results, see Section 6, it makes tags refer to more steps and thus no longer usable as step specifiers. In future versions of `eman`, we may thus remove or somehow restrict the tag propagation feature.

5. Cloning of Experiments

The previous sections described techniques for reusing intermediate results across experiments. Now we describe `eman` commands that allow to reuse the configurations of individual steps and whole experiments. The added twist is that when an existing experiment is "cloned", the variables may be arbitrarily changed.

5.1. Replicating Individual Steps

Cloning of an existing step means creating a new instance of the same step type, reusing most of the variable values. For instance, we may want to create somewhat larger test case for our compression experiment, and we already have the data step "`s.data.aaf8`" ready, as illustrated in Figure 1. The following command will create a new instance using the data seed and run it right away:

```
SIZE=500000 eman clone s.data.aaf8 --start
```

The above command works as an abbreviation of "`eman init data`" where all the variables would have to be specified:

```
SIZE=500000 CMD="cat ../binary.test" TYPE=binary eman init data
--start
```

The cloning will work even without changing any of the variables. In general, it is better to avoid multiple runs of the same configuration, but some computations are non-deterministic and running several copies allows to estimate confidence intervals of the result (Clark et al., 2011). In MT, the prototypical example is the minimum error-rate training, MERT. Creating four more replications of a MERT run is trivial:

```
for i in 2 3 4 5; do eman clone --start s.mert.123; done
```

Replicating a step with identical variables is also useful when a step fails. The command "`eman clone`" as described so far operates on individual steps, so any (failed) dependencies will not get recreated. A better option is described in the following section.

5.2. Redoing Experiments

When an experiment fails, "`eman redo`" can be used to re-create the necessary steps in the whole experiment pipeline. Redo will check the whole traceback of the given experiment and replicate any steps that are failed or outdated. When doing this, the correct links between dependencies are honored, so whenever a step gets redone, its successors will get redone as well.

With large-scale experiments, various technical problems often come into play, making the redo command very useful in day-to-day experimenting. A particular common reason for a failure is a full local temporary disk or memory limits set too low for the given input data, which leads to jobs being killed by the cluster. `Eman`, in cooperation with the scheduling environment, can set the requirements on available memory and disk, so the following usage pattern is quite common:

```
eman redo s.myFailedExp.123 --mem 30g --disk 80g --start --outdate
```

Note that "`eman redo`" walks only the traceback, not the traceforward of the given experiment. It is thus important to ask for a redo of the *final* steps of failed experiments.

5.3. Deriving Whole Experiments

By mixing the idea of modifying variables and redoing whole experiments, we arrive at the full power of experiment cloning.

We have already mentioned, that the traceback with variables (i.e. the output of "`eman tb --vars FINALSTEP`") is the complete description of an experiment. The user can modify some variables in the textual form of the traceback and clone it:

```
eman clone < traceback.modified
```

When constructing steps from such a textual traceback, `eman` automatically discovers steps which can be re-used and only creates the parts of the experiment which are really needed. Experienced `eman` users often create the traceback, substitute some values and create the modified experiment on one line:

```
eman tb -s /oldvalue/newvalue/ | eman clone --dry-run
```

The parameter `-s` defines a substitution which is applied on the whole traceback and supports full Perl regular expressions. The "`--dry-run`" is useful for a quick check before creating the many step directories or "`--start`"ing the new experiment.

We have found cloning of experiments to be extremely useful and versatile in practice. Multiple settings of an MT system can be created and evaluated easily by defining the base experiment and cloning it several times with modified variable values.

With cloning, e.g. reversing the translation direction of an experiment is a trivial change. Similarly, one can easily repeat an experiment for multiple language pairs, change datasets, adjust language model order or modify factors for word alignment.

```
data     var /SIZE=(.*)/SIZE$1B/             /000B\$/kB/
data     var /TYPE=(.*)/TYPE$1/
compress var /CMD=(.*)/CMD$1/
compress var /CMDARGS=(.*)/ARGS$1/
compress var /CMDARGS=.*?-([0-9]).*/LEVEL$1/
```

Figure 3. Sample "eman.autotags" configuration.

6. Making Sense of Results

By a *result*, we mean a small token, usually a number, that was observed or measured during the run of an experiment. In eman's view, results are small bits of information available somewhere in the output files of a step.

Eman provides a set of tools for collecting and interpreting results.

6.1. Autotagging (Tags Based on Variables)

We have already introduced manual tags (Section 4.4) that can be later used to identify e.g. results based on a particular dataset or using a particular version of a program. In addition to tags, eman provides "autotags" that are created automatically from variables of steps using regular expressions and substitutions. The main purpose of automatic tags is to select relevant information from the variables and make it available for the interpretation of results, see below.

The user configures automatic tagging by writing rules into the file "eman.autotags". Each rule consists of the type of steps to which it applies, a regular expression that is matched against the step variables and optionally a regular-expression substitution to be applied on the match – to beautify it in a way.

The tagging rules implemented for our compression example are shown in Figure 3. Each line defines one rule – on the first line, we tell eman to match variables of data steps and look for the pattern "SIZE=.*". We extract the size and prefix it with the word "SIZE". We also perform a simple substitution to shorten the value – we replace "000B" with "kB". The data step shown in Figure 1 is assigned the tag "SIZE10kB" according to this rule.

6.2. Collecting Results

The first task when working with results is to collect them from all the many steps in the playground to a single place. This is achieved using the command "eman collect": all results will appear in the file "eman.results" in the playground directory.

```
s.compress.ba3c96ac.20121217-1415 DONE ratio .47770000000000000000  ARGS-5 CMDbzip2 LEVEL5 SIZE10kB TYPEhexrand
s.compress.bb9c7f1e.20121217-2145 DONE ratio .52478125000000000000  ARGS"-5 --rsyncable" CMDgzip LEVEL5 SIZE512kB TYPEhexrand
s.compress.c45a5afd.20121217-1414 DONE ratio .53680000000000000000  ARGS-9 CMDgzip LEVEL9 SIZE10kB TYPEhexrand
s.compress.99ab03f6.20121217-1436 DONE ratio .43619140625000000000  ARGS-5 CMDbzip2 LEVEL5 SIZE512kB TYPEhexrand
s.compress.02a5f93f.20121217-1436 DONE time  0.042                  ARGS-5 CMDgzip LEVEL5 SIZE512kB TYPEhexrand
s.compress.0545633f.20121217-1413 DONE time  0.002                  ARGS-4 CMDgzip LEVEL4 SIZE10kB TYPEhexrand
s.compress.07151d39.20121217-0116 DONE time  0.003                  ARGS CMDgzip SIZE10kB TYPEhexrand
s.compress.c45a5afd.20121217-1414 DONE TAG   ARGS-9 CMDgzip LEVEL9 SIZE10kB TYPEhexrand
s.compress.7694fe26.20121217-1436 DONE TAG   ARGS CMDbzip2 SIZE512kB TYPEhexrand
```

Figure 4. A few random sample lines from the file "`eman.results`". Each line contains the step name, its status, the name of the result and its value and finally all the tags and autotags assigned to this step.

The specification, what exactly should `eman` extract from a step directory, is provided by the user in the file "`eman.results.conf`". For instance, the configuration line:

```
ratio  →  s.compress.*/ratio  →  CMD: cat
```

specifies that steps of the type "`compress`" measure a particular property, namely the compression ratio that they achieved on some give data. The value can appear anywhere in a file in the step directory as long as a Unix one-line command can extract it. Here, the file "`ratio`" contains just the value of interest, so simply `cat`ting it does the job.

The possibility to run a custom "result extractor" makes collecting of results very flexible: anything can be made important. One can easily introduce new properties to observe at any later time, as long as they were recorded somewhere. Together with remote playgrounds (Section 7), one can re-interpret other people's experiments.

The file "`eman.results`" is useful on its own already. For instance, the user can quickly check if the top-scoring setup is still the same, e.g.:

```
grep ratio eman.results | sort -rn -k4 | head -n 1
```

For the purposes of the following section, we provide a snippet of the results file in Figure 4.

6.3. Tabulation of Results

Lonesome numbers do not have any meaning. In order to be able to interpret the observations and discuss them, individual results have to be compared and contrasted to other results. One practical issue is that a set of results can be dissected and contrasted in an endless number of ways.

`Eman` provides a succinct but extremely powerful tool for "putting relevant numbers next to each other". The technique is based on the "`eman.results`" file and one more user configuration file, "`eman.tabulate`". Running "`eman tabulate`" reorganizes the results based on the configuration and produces "`eman.niceresults`".

```
=== Compression ratios of different algorithms ===
(512k of random hex data)

TABLE
required: compress ratio
required: SIZE512
forbidden: OUTDATED LEVEL[2-46-8]
cols: CMD([^\s]*)
rows: LEVEL([0-9]) rsyncable
rowsort: CMDgzip
ENDTABLE
```

Figure 5. Sample "eman.tabulate" configuration.

6.3.1. Prose with Automatic Tables

The file "eman.tabulate" is a regular text file. Any comments, observations or discussion can be simply written there. Eman copies everything verbatim, except for sections surrounded by lines saying "TABLE" and "ENDTABLE". These sections will get expanded to tables of results. The number of tables in the file is not limited and each table can provide a different view of the results.

One can in principle use "eman.tabulate" as the LaTeX source of a scientific paper where tables are constructed automatically from the available results.

In the following sections, we describe how eman processes the configuration given in Figure 5 to obtain the table in Figure 6.

6.3.2. Selecting Results to Show

The first stage of tabulation is the selection of lines from "eman.results" that should be listed in the table. This filtering allows to provide different views on the playground.

The filtering is achieved by two sets of regular expressions. Only the lines that contain all the "required" expressions and do not contain any of the "forbidden" expressions make it to the table.

Technically, the regular expressions are delimited by space in the "eman.tabulate" config, so a single "required:" line can specify several requirements. To match a space, one can use e.g. "\s".

As each line of the results file contains a lot of details (see Figure 4), the filtering is quite powerful: we can even match e.g. the date in the step name to require steps inited during a particular day. In our example in Figure 5, we are interested in the "ratio" results of any "s.compression.*" step. The autotags provide the information

O. Bojar, A. Tamchyna Eman – an Experiment Manager (39–58)

```
=== Compression ratios of different algorithms ===
(512k of random hex data)

Common properties: compress ratio SIZE512
Forbidden properties: OUTDATED LEVEL[2-46-8]

                 CMDbzip2              CMDgzip
LEVEL1           .46033593750000000000 .55463281250000000000
LEVEL5 rsyncable                     - .52478125000000000000
LEVEL5           .43619140625000000000 .51907031250000000000
                 .43503125000000000000 .51825585937500000000
LEVEL9                               - .51824414062500000000
```

Figure 6. Sample results of the tabulation.

about the file size that was used in the experiments and we require the 512kB tests. Finally, we avoid all steps in the `OUTDATED` state and we also exclude some compression levels ("`forbidden: LEVEL[2-46-8]`") to make the table shorter.

6.3.3. Constructing Row and Column Label for a Result

Each result (i.e. a single line from the results file) that survives the filtering is examined in order to construct its "`row`" and "`col`"umn label. The same regexp mechanism as above is used here, except now the successfully matched regexp is appended to the respective label.

In our example, the columns are simply the compression algorithms – these are found in the autotag starting with "`CMD`". Rows are a little bit more interesting. In general, we want to see the compression level ("`LEVEL([0-9])`"), but in a few experiments, we also used the gzip flag "`--rsyncable`", so we need to distinguish these runs. Adding a second regex "`rsyncable`" that may or may not be found in the result line makes the distinction.

The round brackets in the regexes express important parts of the match. The respective portion of the regex will be replaced by the actual string matched. So the "level" regex appears as a few distinct tokens like "`LEVEL1`", or "`LEVEL9`" in the final table, see Figure 6. Without the round brackets ("`LEVEL[0-9]`"), we would get the same token for all lines, namely "`LEVEL[0-9]`". This can be useful to skipping unimportant differences, i.e. specifying a "gappy pattern".

The order of the regexes is also important, because labels are constructed left-to-right. So regexes that construct the beginning of row or column label need to appear in the configuration first.

Not all result lines match all row/column regexes. That is fine, the label is then simply shorter. As an example, we see the default run of the two compression algorithms where the row label is empty – no level was specified at all.

Not all settings are meaningful or used across all experiments. This is also fine, the cells will then contain just a dash. In our example, it is the "rsyncable" option, which is not available in bzip2, and the level-9 bzip2 experiment which we forgot to run for the purposes of Section 6.3.6.

There are a few other minor tricks for handling cases like multiple matches of the same regex, but these are beyond the scope of this article.

6.3.4. Putting the Table Together, Solving Conflicts

Having established the row and column labels for each value, it is trivial to construct the table. Values sharing the column label will appear in the same column, values sharing the row label will appear in the same row. This gives us a two-dimensional view on the results.

If two or more distinct values share the same row *and* column label, `eman` reports a conflict and the user has two options. If such a conflict is not desirable then some regex (and perhaps also some tag or autotag) should be added to filter out unwanted values or put the conflicting values on different rows or columns. There are however cases where we have deliberately run the very same experiment several times and some randomness or outside condition leads to different results. In this case, one adds the following line to the table specification:

```
collectdelim:,
```

This switch instructs `eman` to indeed show all the results in a single cell, delimited by the given string (a comma in our example).

6.3.5. Sorting Rows and Columns

Finally, the user can specify the full label of the column that should be used to sort the rows ("`rowsort`") and/or the full label of the row that should be used to sort the columns ("`colsort`").

Note that adding regexes that construct row and column labels can easily change the labels so sorting fails to find the given criterion.

6.3.6. Back to Experimenting

`Eman` consults the file "`eman.results`" when resolving step specifiers. This neat trick allows to go directly back from the (tabulated) results to experimenting.

We can ask questions like: what exact configuration did I use to produce the compression ratio 0.5368:

```
eman tb --vars 5368
```

It is wise to double-check that the numbers we contrasted by putting them on the same line or column actually differ only in the properties we are mentioning. In bash, this amounts to inspecting the diff of the two tracebacks, e.g. in the editor vim:

```
vimdiff <(eman tb --vars 4361) <(eman tb --vars 5190)
```

It is also easy to use the cloning mechanism (Section 5.3) to start experiments whose results will fill missing cells. We pick an existing result from the given row (or column, whichever is more convenient) and apply the necessary change to it. We exemplify it by filling the level-9 compression experiment by bzip2 that was missing in Figure 6. The bzip2 run is derived from the corresponding gzip experiment:

```
eman tb 518244 -s /gzip/bzip2/ | eman clone --start
```

7. Team Experimenting

The command "`eman add-remote`" is implemented in a very light-weight fashion. The user provides the path to the remote playground and an alias – `eman` then simply creates a symbolic link to the directory in the local playground and registers the link in the file `eman.subdirs`.

Remote steps then become equivalent to steps in the local playground – they can be used in experiments, cloned and even modified (e.g. started, outdated) if the file system permissions allow it (otherwise, `eman` automatically switches to read-only mode).

`Eman` does not search the remote playground recursively (i.e. it does not explore its remote playgrounds), which makes this feature quite flexible; even circular dependencies are possible, although they do create a soft-link loop in the filesystem.

Commands such as "`eman ls`" or "`eman select`" list only local steps by default. To consider remote playgrounds, the option "`--remote`" has to be used. `Eman` can also display the playground of each step in the listing if "`--dir`" is given.

Finally, since step directories are no longer local subdirs of the playground, the command "`eman path`" is useful to get the full pathname of a step.

8. Related Tools

Two similar tools come from the MT environment: Ducttape and EMS. Ducttape (formerly LoonyBin; `https://github.com/jhclark/ducttape`; Clark et al., 2010) is functionally similar to the combination of `eman` and Prospector (included in `eman` package). The user specifies "hyperworkflows", packed sets of experiments, where a number of variables has a number of requested values. Hyperworkflows are actually more flexible than that, separate hyperworkflow branches can have different step structure. Given a hyperworkflow, Ducttape runs either the full Cartesian product of variable values or a subset of it based on some "realization plan". Implemented in Java, it originally provided only a graphical user interface but now there is also a command-line interface and a minimal web-interface available.

Experiment Management System (EMS; Koehn, 2010), is distributed with the Moses translation system (Koehn et al., 2007) and it is primarily intended for it. Its general management capabilities are again centered around distinct runs of the complete experiment. Data reuse is achieved by noticing that some partial output from a previous run is still valid. This is against our encapsulation objective.

Taverna (`http://www.taverna.org.uk/`) is a widely used complex workflow management tool. It introduces the Taverna language to describe workflows, provides a graphical user interface including an editor of workflows and various servers and clients for running workflows or providing services that can be used as processing blocks in workflows remotely. The remote processing is perhaps the biggest advantage: research institutes provide web-based services directly usable in user's workflows. Compared to `eman`'s 4k lines of Perl, Taverna's command-line tool is 151 MB. Taverna originated in bioinformatics but it is being used in many other fields of research. The only Taverna application in NLP so far are probably the PANACEA tools (`http://www.panacea-lr.eu/`) for compiling various linguistic resources from texts.

Cluster or grid computing environments, e.g. Pegasus (`http://pegasus.isi.edu/`), also have workflow managers like DAGMan (Couvares et al., 2007). These allow to express dependencies between jobs but focus on automatic recovery from job failures in an unreliable cluster environment, not on experiment variation or any interpretation of results.

9. Open Issues and Future Development

There are certainly limitations of the current version of `eman`. The most serious issue from the practical point of view is that the indexing of steps walks many directories and files.[2] With a larger number of steps, this becomes inconveniently slow. A principled solution would use clever incremental updates of only the bits that got invalid due to some change. Unfortunately, this is rather tricky: e.g. changing the autotag configuration would require to propagate new tags to existing steps etc. but `eman` does not get automatically called when the user edits the file "`eman.autotags`".

We have also mentioned, that some inspection and reuse of the *derivation history* for steps is desirable. This would allow further shortcuts in experimenting and new types of observations, e.g. why does the foobar switch make the baseline experiment faster but it slows down our improved setup?

Finally, `eman` has no visual output, but it would be quite easy to display the various dependencies between steps using e.g. the graphviz library (Gansner and North, 2000).

[2] `eman` accumulates an index of steps during regular operations. Full reindexing is required only occasionally and done upon request ("`eman reindex`" for the core index of steps and their variables, "`eman retag`" for autotag application and propagation and "`eman collect`" for the collection of results).

10. Conclusion

We presented `eman`, an open-source experiment manager for command-line Unix environment.

Hopefully, we highlighted and explained a couple of ideas that are generally useful for speed up and a better guidance of experimenting. We feel the following features are the most important ones: keeping detailed records, reusing intermediate results and reusing whole experiments by cloning new variants of them. We also provided a couple of suggestions for organizing and examining obtained results.

For readers interested in `eman` specifically, this article should provide a high-level overview spiced with example calls and commands.

11. Acknowledgment

The work on this project was partially supported by the grants P406/11/1499 of the Grant Agency of the Czech Republic, FP7-ICT-2011-7-288487 (MosesCore) of the European Union and 7E11042 of the Ministry of Education (EU project FP7-ICT-2010-6-257528).

Bibliography

Clark, Jonathan H., Jonathan Weese, Byung Gyu Ahn, Andreas Zollmann, Qin Gao, Kenneth Heafield, and Alon Lavie. The Machine Translation Toolpack for LoonyBin: Automated Management of Experimental Machine Translation HyperWorkflows. *Prague Bulletin of Mathematical Linguistics*, 93:117–126, 2010. ISSN 0032-6585.

Clark, Jonathan H., Chris Dyer, Alon Lavie, and Noah A. Smith. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proc. of ACL/HLT 2011*, pages 176–181, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-2031`.

Couvares, Peter, Tevfik Kosar, Alain Roy, Jeff Weber, and Kent Wenger. Workflow Management in Condor. In *Workflows for e-Science*, pages 357–375. Springer London, 2007. ISBN 978-1-84628-519-6. doi: 10.1007/978-1-84628-757-2_22. URL `http://dx.doi.org/10.1007/978-1-84628-757-2_22`.

Gansner, Emden R. and Stephen C. North. An open graph visualization system and its applications to software engineering. *SOFTWARE - PRACTICE AND EXPERIENCE*, 30(11): 1203–1233, 2000.

Koehn, Philipp. An Experimental Management System. *Prague Bulletin of Mathematical Linguistics*, 94:87–96, Sept. 2010. ISSN 0032-6585.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of ACL 2007, Companion Volume Proceedings of the Demo and*

Poster Sessions, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P07/P07-2045.

Och, Franz Josef. Minimum error rate training in statistical machine translation. In *Proc. of ACL 2003*, pages 160–167, Sapporo, Japan, 2003. ACL. URL http://aclweb.org/anthology-new/P/P03/#1000.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proc. of ACL 2002*, pages 311–318. Association for Computational Linguistics, 2002. URL http://aclweb.org/anthology-new/P/P02/.

Tamchyna, Aleš and Ondřej Bojar. No free lunch in factored phrase-based machine translation. In *Proc. of CICLing 2013*, volume 7817 of *Lecture Notes in Computer Science*, pages 210–223, Samos, Greece, 2013. Springer-Verlag.

Address for correspondence:
Ondřej Bojar
bojar@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics,
Charles University in Prague
Malostranské náměstí 25
118 00 Praha 1, Czech Republic

2010 Failures in English-Czech Phrase-Based MT *

Ondřej Bojar and Kamil Kos
Charles University in Prague, Institute of Formal and Applied Linguistics (ÚFAL)
Malostranské náměstí 25, Praha 1, CZ-11800, Czech Republic
bojar@ufal.mff.cuni.cz, kamilkos@email.cz

Abstract

The paper describes our experiments with English-Czech machine translation for WMT10[1] in 2010. Focusing primarily on the translation to Czech, our additions to the standard Moses phrase-based MT pipeline include two-step translation to overcome target-side data sparseness and optimization towards SemPOS, a metric better suited for evaluating Czech. Unfortunately, none of the approaches bring a significant improvement over our standard setup.

1 Introduction

Czech is a flective language with very rich morphological system. Translation between Czech and English poses different challenges for each of the directions.

When translating from Czech, the word order usually needs only minor changes (despite the issue of non-projectivity, a phenomenon occurring at 2% of words but in 23% of Czech sentences, see Hajičová et al. (2004) and Holan (2003)). A much more severe issue is caused by the Czech vocabulary size. Fortunately, this can be to a certain extent mitigated by backing-off to Czech lemmas if the exact forms are not available.

We are primarily interested in the harder task of translating to Czech and most of the paper deals with this direction. After a brief specification of data sets, pre-processing and evaluation method in this section, we provide details on the issue of Czech vocabulary size (Section 2). We describe our current attempts at generating Czech word forms in Section 3. Partly due to the large vocabulary size of Czech, BLEU score (Papineni et al., 2002) correlates rather poorly with human judgments. We summarize our efforts to use a better metric in the model optimization in Section 4. The final Section 5 lists the exact configurations of our English↔Czech primary submissions for WMT10, including the back-off to lemmas we use for Czech-to-English.

1.1 Data and Pre-Processing Pipeline

Throughout the paper, we use CzEng 0.9 (Bojar and Žabokrtský, 2009)[2] as our main parallel corpus. Following CzEng authors' request, we did not use sections 8* and 9* reserved for evaluation purposes.

As the baseline training dataset ("Small" in the following) only the news domain of CzEng (126k parallel sentences) is used. For large-scale experiments ("Large" in the following) and our primary WMT10 submissions, we use all CzEng domains except `navajo` and add the EMEA corpus (Tiedemann, 2009)[3,4] of 7.5M parallel sententes.

As our monolingual data we use by default only the target side of the parallel corpus. For experiments reported here, we also use the monolingual data provided by WMT10 organizers for Czech. Our primary WMT10 submission includes further monolingual data, see Section 5.1.

We use a slightly modified tokenization rules compared to CzEng export format. Most notably, we normalize English abbreviated negation and auxiliary verbs ("couldn't" → "could not") and attempt at normalizing quotation marks to distinguish between the opening and closing one follow-

*The work on this project was supported by the grants EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic), GAČR P406/10/P259, and MSM 0021620838. Thanks to David Kolovratník for the help with manual evaluation.

[1] `http://www.statmt.org/wmt10/`

[2] `http://ufal.mff.cuni.cz/czeng`

[3] `http://urd.let.rug.nl/tiedeman/OPUS`

[4] Unfortunately, the EMEA corpus is badly tokenized on the Czech side. Most frequently, fractional numbers are split into several tokens (e.g. "3, 14"). We attempted to reconstruct the original detokenized form using a small set of regular expressions.

Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 60–66,
Uppsala, Sweden, 11-16 July 2010.

	Large	Small	Dev
Sents	7.5M	126.1k	2.5k
Czech Tokens	79.2M	2.6M	55.8k
English Tokens	89.1M	2.9M	49.9k
Czech Vocabulary	923.1k	138.7k	15.4k
English Vocabulary	646.3k	64.7k	9.4k
Czech Lemmas	553.5k	60.3k	9.5k
English Lemmas	611.4k	53.8k	7.7k

Table 1: Corpus and vocabulary sizes.

ing proper typesetting rules.

The rest of our pre-processing pipeline matches the processing employed in CzEng (Bojar and Žabokrtský, 2009).[5] We use "supervised truecasing", meaning that we cast the case of the lemma to the form, relying on our morphological analyzers and taggers to identify proper names, all other words are lowercased.

The differences in relations between Czech and English Large and Small datasets can be attributed either to domain differences or possibly due to noise in CzEng.

1.2 Evaluation

We use WMT10 development sets for tuning (news-test2008) and evaluation (news-test2009). The official scores on news-test2010 are given only in the main WMT10 paper and not here.

The BLEU scores reported in this paper are based on truecased word forms in the original tokenization as provided by the decoder. Therefore they are likely to differ from figures reported elsewhere.

The $\pm$ value given with each BLEU score is the average of the distances to the lower and upper empirical 95% confidence bounds estimated using bootstrapping (Koehn, 2004).

2 Issues of Czech Vocabulary Size

Table 1 summarizes the differences of Czech and English vocabulary sizes in our parallel corpora. We see that the vocabulary size of Czech forms (truecased) is more than double compared to English in the Small dataset and significantly larger in the Large dataset as well. On the other hand, the number of distinct Czech and English lemmas is nearly identical.

[5] Due to the subsequent processing, incl. parsing, the tokenization of English follows PennTreebenk style. The rather unfortunate convention of treating hyphenated words as single tokens increases our out-of-vocabulary rate. Next time, we will surely post-tokenize the parsed text.

	Distortion Limit				
TOpts	3	6	10	30	40
1	0.2	0.3	0.3	0.3	0.3
5	0.8	0.9	1.0	1.0	1.0
10	1.1	1.3	1.5	1.5	1.5
20	1.2	1.5	1.7	1.7	1.7
50	1.2	1.5	1.7	1.7	1.7
100	1.2	1.5	1.7	1.7	1.7

Table 3: Percentage of sentences reachable in Czech-to-English small setting with various distortion limits and translation options per coverage (TOpts) (BLEU score 14.76$\pm$0.44).

2.1 Out-of-Vocabulary Rates

Table 2 lists out-of-vocabulary (OOV) rates of our Small and Large data setting given the development corpus. We calculate the rates for both the complete corpus and the restricted set of phrases extracted from the corpus. (Note that higher-order n-gram rates are estimated using phrases as independent units, no combination of phrases is performed.) We also list the effective OOV rate for English-to-Czech translation where all (English) words from each source sentence can be also produced in the hypothesis.

We see that in the small setting, the OOV rate is almost double for Czech than for English. The OOV is significantly decreased by enlarging the corpus or lemmatizing the word forms.

If we consider only the words available in the phrase tables, the issue of Czech with limited data is striking: 10–12% of devset tokens are not available in the training data.

2.2 Reachability of Training and Reference Translations

Schwartz (2008) extended Moses to support "constraint decoding", that is to perform an exhaustive search through the space of hypotheses in order to reach the reference translation (and get its score).

The current implementation of the exhaustive search in Moses is in fact subject to several configuration parameters, most importantly the number of translation options considered for each span (`-max-trans-opt-per-coverage`) and the distortion limit (`-distortion-limit`).

Given his aim, Schwartz (2008) uses the output of four MT systems translating from different languages to English as the references and notes that only around 10% of the reference translations are reachable by an independent Swedish-English MT system.

		n-grams Out of Corpus Voc.				n-grams Out of Phrase-Table Voc.			
Dataset	Language	1	2	3	4	1	2	3	4
Large	Czech	2.2%	30.5%	70.2%	90.3%	3.9%	44.1%	82.2%	95.6%
Large	English	1.5%	13.7%	47.3%	78.8%	2.1%	22.4%	63.5%	89.1%
Large	Czech + English input sent	1.5%	29.4%	69.6%	90.1%	3.1%	42.8%	81.5%	95.3%
Small	Czech	6.7%	48.1%	83.0%	95.5%	12.5%	65.4%	91.9%	98.6%
Small	English	3.6%	28.1%	68.3%	90.9%	6.3%	45.4%	84.3%	97.0%
Small	Czech + English input sent	5.2%	46.6%	82.4%	95.2%	10.6%	63.7%	91.2%	98.3%
Small	Czech lemmas	4.1%	36.3%	75.8%	92.8%	5.8%	52.6%	87.7%	97.4%
Small	English lemmas	3.4%	24.6%	64.6%	89.4%	6.9%	53.2%	87.9%	97.5%
Small	Czech + English input sent lemmas	3.1%	35.7%	75.6%	92.8%	5.1%	38.1%	80.8%	96.2%

Table 2: Out-of-vocabulary rates.

	Distortion Limit				
TOpts	3	6	10	30	40
1	0.4	0.4	0.4	0.4	0.4
5	1.5	1.9	2.0	2.0	2.0
10	2.5	3.2	3.5	3.5	3.5
20	3.7	5.0	5.5	5.6	5.6
50	4.9	6.7	8.0	8.6	8.6
100	5.3	7.6	9.1	9.4	9.4

Table 4: Percentage of sentences reachable in Czech-to-English large setting, two alternative decoding paths to translate from Czech lemma if the form is not available in the translation table (BLEU score 18.70±0.46).

We observe that reaching man-made reference translations in Czech-to-English translation is far harder. Table 3 provides the figures for small data setting (and no phrase table filtering). The best reachability we can hope for is given in Table 4 where we allow to use source word lemmas if the exact form is not available. We see that the default limits (50 translation options per span and distortion limit of 6) leave us with only 6.7% sentences reachable.

While not directly important for your training, the figures still underpin the issue of sparse data in Czech-English translation.

3 Targetting Czech Word Forms

Bojar (2007) experimented with several translation scenarios, including what we will call MorphG, i.e. the independent translation of lemma to lemma and tag to tag followed by a generation step to produce target-side word form. With the small training set available then, the MorphG model performed equally well as a simpler direct translation followed by target-side tagging and an additional n-gram model over morphological tags. Koehn and Hoang (2007) reports even a large loss with MorphG for German-to-English if the alternative of direct form-to-form translation is not available.

Bojar et al. (2009b) applied the two alternative decoding paths (direct form-to-form and MorphG, labelled "T+C+C&T+T+G") to English-Czech but they were able to use only 84k sentences. For the full training set of 2.2M sentences, the model was too big to fit in reasonable disk limits. More importantly, already in the small data setting, the complex model suffered from little stability due to abundance of features (5 features per phrase-table plus tree features for three LMs), so nearly the same performance on the development set gave largely varying quality on the independent test set.

The most important issue of the MorphG setup, however, is the explosion of translation options. Due to the "synchronous factors" approach of Moses (Koehn and Hoang, 2007), all translation options have to be fully constructed before the main search begins. The MorphG model however licenses too many possible combinations of lemmas, tags and final word forms, so the pruning of translation options strikes hard, causing search errors. For more details, see Bojar et al. (2009a) where a similar issue occurs for treelet-based translation.

3.1 Two-Step Translation

In order to avoid the explosion of the translation options[6], we experimented with two-step translation.

The first step translates from English to lemmatized Czech augmented to preserve important semantic properties known from the source phrase. The second step is a monotone translation from the lemmas to fully inflected Czech. The idea behind the delimitation is that all the morphological properties of Czech words that can be established

[6] and also motivated when we noticed that reading MT output to *lemmatized* Czech is sometimes more pleasant and informative than regular phrase-based output

Data Size		Simple		Two-Step	
Parallel	Mono	BLEU	SemPOS	BLEU	SemPOS
Small	Small	10.28±0.40	29.92	10.38±0.38	30.01
Small	Large	12.50±0.44	31.01	12.29±0.47	31.40
Large	Large	14.17±0.51	33.07	14.06±0.49	32.57

Table 5: Performance of direct (Simple) and two-step factored translation in small and large data setting.

regardless the English source should not cause parallel data sparseness and clutter the search. Instead, they should be decided based on context in the second phase only.

Specifically, the intermediate Czech represents most words as tuples containing only: lemma, negation, grade (of adjectives and adverbs), number (of nouns, adjectives, verbs) and detailed part of speech (constraining also e.g. verb tense of Czech verbs). Some words are handled separately:

- Pronouns, punctuation and the verbs "být" (to be) and "mít" (to have) are represented using their lowecased full forms because they are very frequent, often auxiliary to other words and their exact form best captures the available and necessary detail of many morphological and syntactic properties.
- Prepositions are represented using their lemmas and case because the case of a noun phrase is actually introduced by the governing word (e.g. the verb that subcategorized for the noun phrase or the preposition for prepositional phrases).

Table 5 compares the scores of the simple phrase-based and the two-step translation via augmented Czech lemmas as described above. The small and large parallel data denote the datasets described in Section 1.1. The small monolingual set means just the news domain of CzEng, while the large monolingual set means WMT10 monolingual Czech texts (and no CzEng data). Note that the monolingual data serve three purposes in the two-step approach: the language model for the first phase, the translation model in the second phase (monotone and restricted to phrase-length of 1; longer phrases did not bring significant improvement either), and the language model of the second phase. Ignoring the opportunity to use the monolingual set as the language model in the first phase already hurts the performance.

We see that the results as evaluated both by BLEU and SemPOS (see Section 4 below) are rather mixed but not that surprising. There is a negligible gain in the Small-Small setting, a mixed outcome in the Small-Large and a little loss in the Large-Large setting.

	Two--Step	Both Fine	Both Wrong	Simple	Total
Two-Step	**23**	4	8	-	**35**
Both Fine	7	14	17	5	43
Both Wrong	8	1	28	2	39
Simple	-	3	7	**23**	33
Total	**38**	22	60	30	150

Table 6: Manual micro-evaluation of Simple (12.50±0.44) vs. Two-step (12.29±0.47) model in the Small-Large setting.

The most interesting result is the Small-Large setting: BLEU (insignificantly) prefers the simple and SemPOS the two-step model. It thus seems that a large target-side LM is sufficient to improve the BLEU score, despite the untackled issue of bilingual data sparseness.

We carried out a quick manual evaluation of 150 sentences by two annotators (one of the authors and a third person; systems anonymized): for each input segment, either one of the outputs is distinguishably better or both are equally wrong or equally acceptable. As listed in the confusion matrix in Table 6, each annotator independently marginally prefers the two-step approach but the intersection does not confirm that.[7] One good thingis that the annotators do not completely contradict each other's preference.

Ultimately, we did not use the two-step approach in our primary submission, but we feel there is still some unexploited potential in this phrase-based approximation of the technique separating properties of words handled in the translation phase from properties implied by the target-side (grammatical) context only. Certainly, the representation of the intermediate language can

[7] Of the 23 sentences improved by the two-step setup, about three quarters indeed had an improvement in lexical coverage or better morphological choice of a word. Of the 23 sentences where the two-step model hurts, about a half suffered from errors related to superfluous auxiliary words in Czech that seem to be introduced by a bias towards word-for-word translation. This bias is not inherent to the model, only the (normalized) phrase penalty weight happened to get nearly three times bigger than in the simple model.

be still improved, and more importantly, the second phase of monotone decoding could be handled by a more appropriate model capable of including more additional (source) context features.[8]

4 Optimizing towards SemPOS

In our setup, we use minimum error-rate training (MERT, Och (2003)) to optimize weights of model components. In the standard implementation in Moses, BLEU (Papineni et al., 2002) is used as the objective function, despite its rather disputable correlation with human judgments of MT quality.

Kos and Bojar (2009) introduced SemPOS, a metric that performs much better in terms of correlation to human judgments when translating to Czech. Naturally, we wanted to optimize towards SemPOS.

SemPOS computes the overlapping of autosemantic (content-bearing) word lemmas in the candidate and reference translations given a fine-grained semantic part of speech (sempos[9]), as defined in Hajič et al. (2006), and outputs average overlapping score over all sempos types.

The SemPOS metric outperformed common metrics as BLEU, TER (Snover et al., 2006) or an adaptation of Meteor (Lavie and Agarwal, 2007) for Czech on test sets from WMT08 (Callison-Burch et al., 2008).

4.1 Integrating SemPOS to MERT

In our experiments we used Z-MERT (Zaidan, 2009), a recent implementation of the MERT algorithm, to optimize model parameters.

The SemPOS metric requires to remove all auxiliary words and to identify the (deep-syntactic) lemmas and semantic part of speech for autosemantic words. When employed in MERT training, the whole n-best list of candidates has to processed like this at each iteration.

We use the TectoMT platform (Žabokrtský and Bojar, 2008)[10] for the linguistic processing. TectoMT follows the complete pipeline of tagging, surface-syntactic analysis and deep-syntactic analysis, which is the best but rather costly way to obtain the required information.

Therefore, we use two different ways of obtaining lemmas and semantic parts of speech in the MERT loop:

- indeed apply TectoMT processing to the n-best list at each iteration (parallelized to 15 CPUs),
- apply TectoMT to the *training data*, express the (deep) lemma and sempos as additional factors using a blank value for auxiliary words, and using Moses factored translation to translate from English forms to triplets of Czech form, deep lemma and sempos.

	BLEU	SemPOS	Iters	Time
TectoMT	10.11±0.40	29.69	20	2d12.0h
in MERT	9.53±0.39	29.69	10	1d12.0h
Factored	9.46±0.37	29.36	10	2.4h
translation	8.20±0.37	29.68	-	-
	6.96±0.33	27.79	9	1.7h

Table 7: Five independent MERT runs optimizing towards SemPOS with semantic parts of speech and lemmas provided either by TectoMT on the fly or by Moses factored translation.

Table 7 lists several ZMERT runs when optimizing a simple form→form phrase-based model (small data setting) towards SemPOS. One observation is that using TectoMT in the MERT loop is unbearably costly and we avoided it in the subsequent experiments. More importantly, from the huge differences in the final BLEU as well as SemPOS scores (evaluated on the independent test set), we see how unstable the search is.

SemPOS, while good at comparing different MT systems, is very bad at comparing candidates from a single system in an n-best list. This can be easily explained by its low sensitivity to precision: SemPOS disregards word forms as well as all auxiliary words. This is a good thing to compare very different candidates (where each of the systems already struggled to produce a coherent output) but is of very little help when comparing candidates of a single system, because these candidates tend to differ rather in forms than in lexical choice.

4.2 Combination of SemPOS and BLEU

To compensate for some of the shortcomings of SemPOS, we also attempted to optimize towards a linear combination of SemPOS and BLEU. This should increase the suitability of the metric for MERT optimization because BLEU will take correct word forms into account while SemPOS should promote better lexical choice (possibly not confirmed by BLEU due to a different word form than in the reference).

Table 8 provides the results of various weight

[8] We are grateful to Trevor Cohn for the suggestion.

[9] In the following text we will use SemPOS to denote the SemPOS metric. When speaking about the semantic part of speech, we will write sempos type or sempos tag.

[10] `http://ufal.mff.cuni.cz/tectomt/`

W.	BLEU	SemPOS	W.	BLEU	SemPOS
1:0	10.42±0.38	29.91	3:1	10.30±0.39	30.03
1:1	10.15±0.39	29.81	10:1	10.17±0.40	29.58
1:1	9.42±0.37	29.30	1:2	10.11±0.38	29.80
2:1	10.37±0.38	29.95	1:10	9.44±0.40	29.74

Table 8: Optimizing towards a linear combination of BLEU and SemPOS (weights in this order), small data setting.

	BLEU	SemPOS
BLEU alone	14.08±0.50	32.44
SemPOS-BLEU (1:1)	13.79±0.55	33.17

Table 9: Optimizing towards BLEU and/or SemPOS in large data setting.

settings, including the optimization towards BLEU alone using ZMERT implementation. We see that the stability is much better, only few runs suffered a minor loss (including 1:1 in one case). Unfortunately, the differences in final BLEU and SemPOS scores are all within confidence intervals when trained on the small dataset.

Table 9 documents that in our large data setting, MERT indeed achieves slightly higher SemPOS (and lower BLEU) when optimizing towards it. This corresponds with the intuition that with more variance in lexical choices available in the phrase tables, SemPOS can help to balance model features. The current set of weights is rather limited, so our future experiments should focus on actually providing means to e.g. domain adaptation by using features indicating the applicability of a phrase in a specific domain.

5 Our Primary Submissions to WMT10

5.1 English-to-Czech Translation

Given the little or no improvements achieved by the many configurations we tried, our English-to-Czech primary submission is rather simple:

- Standard GIZA++ word alignment based on both source and target lemmas.
- Two alternative decoding paths; forms always truecased: form+tag→form & form→form.
  The first path is more specific and helps to preserve core syntactic elements in the sentence. Without the tag, ambiguous English words could often all translate as e.g. nouns, leading to no verb in the Czech sentence. The default path serves as a back-off.
- Significance filtering of the phrase tables (Johnson et al., 2007) implemented for Moses by Chris Dyer; default settings of filter value `a+e` and the cut-off 30.
- Two separate 5-gram Czech LMs of truecased forms each of which interpolates models trained on the following datasets; the interpolation weights were set automatically using SRILM (Stolcke, 2002) based on the target side of the development set:[11]
  - Interpolated CzEng domains: news, web, fiction. The rationale behind the selection of the domains is that we prefer prose-like texts for LM estimation (and not e.g. technical documentation) while we want as much parallel data as possible.
  - Interpolated monolingual corpora: WMT09 monolingual, WMT10 monolingual, Czech National Corpus (Kocek et al., 2000) sections SYN2000+2005+2006PUB.
- Lexicalized reordering (`or-bi-fe`) based on forms.
- Standard Moses MERT towards BLEU.

	Large	Small
Backed-off by source lemmas	18.95±0.45	14.95±0.48
form→form only	18.41±0.44	14.73±0.47

Table 10: Translation from Czech better when backed-off by source lemmas.

5.2 Czech-to-English Translation

For Czech-to-English translation we experimented with far fewer configuration options. Our primary submission is configured as follows:

- Two alternative decoding paths; forms always truecased: form→form & lemma→form.
- Significance filtering as in Section 5.1.
- 5-gram English LM based on CzEng English side only.[12]
- Lexicalized reordering (`or-bi-fe`) based on forms.
- Standard Moses MERT towards BLEU.

Table 10 documents the utility of the additional decoding path from Czech lemmas in both small and large setting, surprisingly less significant in the small setting. Later experiments with system combination by Kenneth Heafield indicated that while our system is not among the top three, it brings an advantage to the combination.

6 Conclusion

We provided an extensive documentation of Czech data sparseness issue for machine translation. We attempted to tackle the problem of constructing the target-side form by a two-step translation setup and the problem of unreliable automatic evaluation by employing a new metric in MERT loop, neither with much success so far. Both of the attempts however deserve further exploration. Additionally, we provide the exact configurations of our WMT10 primary submissions.

[11] The subsequent MERT training using the same development test may suffer from overestimating the language model weights, but we did not observe the issue, possibly due to only moderate overlap of the datasets.

[12] We attempted to use a second LM trained on English Gigaword by Chris Callison-Burch, but we observed a drop in BLEU score from 18.95±0.45 to 18.03±0.44 probably due to different tokenization guidelines applied.

References

Ondřej Bojar and Zdeněk Žabokrtský. 2009. CzEng 0.9: Large Parallel Treebank with Rich Annotation. *Prague Bulletin of Mathematical Linguistics*, 92.

Ondřej Bojar, Miroslav Janíček, and Miroslav Týnovský. 2009a. Evaluation of Tree Transfer System. Project Euromatrix - Deliverable 3.4, Institute of Formal and Applied Linguistics, Charles University in Prague.

Ondřej Bojar, David Mareček, Václav Novák, Martin Popel, Jan Ptáček, Jan Rouš, and Zdeněk Žabokrtský. 2009b. English-Czech MT in 2008. In *Proc. of Fourth Workshop on Statistical Machine Translation, ACL*, Athens, Greece.

Ondřej Bojar. 2007. English-to-Czech Factored Machine Translation. In *Proc. of the Second Workshop on Statistical Machine Translation, ACL*, Prague, Czech Republic, June.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further Meta-Evaluation of Machine Translation. In *Proc. of the Third Workshop on Statistical Machine Translation, ACL*, Columbus, Ohio.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková Razímová. 2006. Prague Dependency Treebank 2.0. LDC2006T01, ISBN: 1-58563-370-4.

Eva Hajičová, Jiří Havelka, Petr Sgall, Kateřina Veselá, and Daniel Zeman. 2004. Issues of Projectivity in the Prague Dependency Treebank. *The Prague Bulletin of Mathematical Linguistics*, 81.

Tomáš Holan. 2003. K syntaktické analýze českých(!) vět. In *MIS 2003*. MATFYZPRESS.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proc. of EMNLP-CoNLL*, Prague, Czech Republic.

Jan Kocek, Marie Kopřivová, and Karel Kučera, editors. 2000. *Český národní korpus - úvod a příručka uživatele*. FF UK - ÚČNK, Prague.

Philipp Koehn and Hieu Hoang. 2007. Factored Translation Models. In *Proc. of EMNLP*.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proc. of EMNLP*, Barcelona, Spain.

Kamil Kos and Ondřej Bojar. 2009. Evaluation of Machine Translation Metrics for Czech as the Target Language. *The Prague Bulletin of Mathematical Linguistics*, 92.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proc. of the Second Workshop on Statistical Machine Translation, ACL*, Prague, Czech Republic.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of ACL*, Sapporo, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*, Philadelphia, Pennsylvania.

Lane Schwartz. 2008. Multi-source translation methods. In *Proc. of AMTA*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proc. of AMTA*.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. of Intl. Conf. on Spoken Language Processing*, volume 2.

Jörg Tiedemann. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In *Proc. of Recent Advances in NLP (RANLP)*.

Zdeněk Žabokrtský and Ondřej Bojar. 2008. TectoMT, Developer's Guide. Technical Report TR-2008-39, Institute of Formal and Applied Linguistics, Charles University in Prague.

Omar F. Zaidan. 2009. Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems. *The Prague Bulletin of Mathematical Linguistics*, 91.

Improving Translation Model by Monolingual Data*

Ondřej Bojar and Aleš Tamchyna
bojar@ufal.mff.cuni.cz, a.tamchyna@gmail.com
Institute of Formal and Applied Linguistics,
Faculty of Mathematics and Physics, Charles University in Prague

Abstract

We use target-side monolingual data to extend the vocabulary of the translation model in statistical machine translation. This method called "reverse self-training" improves the decoder's ability to produce grammatically correct translations into languages with morphology richer than the source language esp. in small-data setting. We empirically evaluate the gains for several pairs of European languages and discuss some approaches of the underlying back-off techniques needed to translate unseen forms of known words. We also provide a description of the systems we submitted to WMT11 Shared Task.

1 Introduction

Like any other statistical NLP task, SMT relies on sizable language data for training. However the parallel data required for MT are a very scarce resource, making it difficult to train MT systems of decent quality. On the other hand, it is usually possible to obtain large amounts of monolingual data.

In this paper, we attempt to make use of the monolingual data to reduce the sparseness of surface forms, an issue typical for morphologically rich languages. When MT systems translate into such languages, the limited size of parallel data often causes the situation where the output should include a word form never observed in the training data. Even though the parallel data do contain the desired word in other forms, a standard phrase-based decoder has no way of using it to generate the correct translation.

Reverse self-training addresses this problem by incorporating the available monolingual data in the translation model. This paper builds upon the idea outlined in Bojar and Tamchyna (2011), describing how this technique was incorporated in the WMT Shared Task and extending the experimental evaluation of reverse self-training in several directions – the examined language pairs (Section 4.2), data size (Section 4.3) and back-off techniques (Section 4.4).

2 Related Work

The idea of using monolingual data for improving the translation model has been explored in several previous works. Bertoldi and Federico (2009) used monolingual data for adapting existing translation models to translation of data from different domains. In their experiments, the most effective approach was to train a new translation model from "fake" parallel data consisting of target-side monolingual data and their machine translation into the source language by a baseline system.

Ueffing et al. (2007) used a boot-strapping technique to extend translation models using monolingual data. They gradually translated additional source-side sentences and selectively incorporated them and their translations in the model.

Our technique also bears a similarity to de Gispert et al. (2005), in that we try to use a back-off for surface forms to generalize our model and produce translations with word forms never seen in the original parallel data. However, instead of a rule-based approach, we take advantage of the available

* This work has been supported by the grants EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic), P406/10/P259, and MSM 0021620838.

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 330–336,
Edinburgh, Scotland, UK, July 30–31, 2011.

	Source English		Target Czech	Czech Lemmatized
Parallel (small)	a cat chased…	=	**kočka** honila…	*kočka honit…*
	I saw a cat	=	viděl jsem **kočku**	*vidět být kočka*
	I read about a dog	=	četl jsem o psovi	*číst být o pes*
Monolingual (large)	?		četl jsem o **kočce**	*číst být o kočka*
	I read about a cat	←	Use reverse translation backed-off by lemmas.	

Figure 1: The essence of reverse self-training: a new phrase pair ("about a cat" = "o **kočce**") is learned based on a small parallel corpus and large target-side monolingual texts.

data and learn these forms statistically. We are therefore not limited to verbs, but our system is only able to generate surface forms observed in the target-side monolingual data.

3 Reverse Self-Training

Figure 1 illustrates the core of the method. Using available parallel data, we first train an MT system to translate from the target to the source language. Since we want to gather new word forms from the monolingual data, this reverse model needs the ability to translate them. For that purpose we use a factored translation model (Koehn and Hoang, 2007) with two alternative decoding paths: form→form and back-off→form. We experimented with several options for the back-off (simple stemming by truncation or full lemmatization), see Section 4.4. The decoder can thus use a less sparse representation of words if their exact forms are not available in the parallel data.

We use this reverse model to translate (much larger) target-side monolingual data into the source language. We preserve the word alignments of the phrases as used in the decoding so we directly obtain the word alignment in the new "parallel" corpus. This gives us enough information to proceed with the standard MT system training – we extract and score the phrases consistent with the constructed word alignment and create the phrase table.

We combine this enlarged translation model with a model trained on the true parallel data and use Minimum Error Rate Training (Och, 2003) to find the balance between the two models. The final model has four separate components – two language models (one trained on parallel and one on monolingual data) and the two translation models.

We do not expect the translation quality to improve simply because more data is included in training – by adding translations generated using known data, the model could gain only new combinations of known words. However, by using a back-off to less sparse units (e.g. lemmas) in the factored target→source translation, we enable the decoder to produce previously unseen surface forms. These translations are then included in the model, reducing the data sparseness of the target-side surface forms.

4 Experiments

We used common tools for phrase-based translation – Moses (Koehn et al., 2007) decoder and tools, SRILM (Stolcke, 2002) and KenLM (Heafield, 2011) for language modelling and GIZA++ (Och and Ney, 2000) for word alignments.

For reverse self-training, we needed Moses to also output word alignments between source sentences and their translations. As we were not able to make the existing version of this feature work, we added a new option and re-implemented this funcionality.

We rely on automatic translation quality evaluation throughout our paper, namely the well-established BLEU metric (Papineni et al., 2002). We estimate 95% confidence bounds for the scores as described in Koehn (2004). We evaluated our translations on lower-cased sentences.

4.1 Data Sources

Aside from the WMT 2011 Translation Task data, we also used several additional data sources for the experiments aimed at evaluating various aspects of reverse self-training.

JRC-Acquis

We used the JRC-Acquis 3.0 corpus (Steinberger et al., 2006) mainly because of the number of available languages. This corpus contains a large amount

Source	Target	Corpus Size (k sents) Para	Mono	Vocabulary Size Ratio	Baseline	+Mono LM	+Mono TM
English	Czech	94	662	1.67	40.9±1.9	43.5±2.0	*44.3±2.0
English	Finnish	123	863	2.81	27.0±1.9	27.6±1.8	28.3±1.7
English	German	127	889	1.83	34.8±1.8	36.4±1.8	37.6±1.8
English	Slovak	109	763	2.03	35.3±1.6	37.3±1.7	37.7±1.8
French	Czech	95	665	1.43	39.9±1.9	42.5±1.8	43.1±1.8
French	Finnish	125	875	2.45	26.7±1.8	27.8±1.7	28.3±1.8
French	German	128	896	1.58	38.5±1.8	40.2±1.8	*40.5±1.8
German	Czech	95	665	0.91	35.2±1.8	37.0±1.9	*37.3±1.9

Table 1: BLEU scores of European language pairs on JRC data. Asterisks in the last column mark experiments for which MERT had to be re-run.

of legislative texts of the European Union. The fact that all data in the corpus come from a single, very narrow domain has two effects – models trained on this corpus perform mostly very well in that domain (as documented e.g. in Koehn et al. (2009)), but fail when translating ordinary texts such as news or fiction. Sentences in this corpus also tend to be rather long (e.g. 30 words on average for English).

CzEng

CzEng 0.9 (Bojar and Žabokrtský, 2009) is a parallel richly annotated Czech-English corpus. It contains roughly 8 million parallel sentences from a variety of domains, including European regulations (about 34% of tokens), fiction (15%), news (3%), technical texts (10%) and unofficial movie subtitles (27%). We do not make much use of the rich annotation in this paper, however we did experiment with using Czech lemmas (included in the annotation) as the back-off factor for reverse self-training.

4.2 Comparison Across Languages

In order to determine how successful our approach is across languages, we experimented with Czech, Finnish, German and Slovak as target languages. All of them have a rich morphology in some sense. We limited our selection of source languages to English, French and German because our method focuses on the target language anyway. We did however combine the languages with respect to the richness of their vocabulary – the source language has less word forms in almost all cases.

Czech and Slovak are very close languages, sharing a large portion of vocabulary and having a very similar grammar. There are many inflectional rules for verbs, nouns, adjectives, pronouns and numerals. Sentence structure is exhibited by various agreement rules which often apply over long distance. Most of the issues commonly associated with rich morphology are clearly observable in these languages.

German also has some inflection, albeit much less complex. The main source of German vocabulary size are the compound words. Finnish serves as an example of agglutinative languages well-known for the abundance of word forms.

Table 1 contains the summary of our experimental results. Here, only the JRC-Acquis corpus was used for training, development and evaluation. For every language pair, we extracted the first 10 percent of the parallel corpus and used them as the parallel data. The last 70 percent of the same corpus were our "monolingual" data. We used a separate set of 1000 sentences for the development and another 1000 for testing.

Sentence counts of the corpora are shown in the columns Corpus Size Para and Mono. The table also shows the ratio between observed vocabulary size of the target and source language. Except for the German→Czech language pair, the ratios are higher than 1. The Baseline column contains the BLEU score of a system trained solely on the parallel data (i.e. the first 10 percent). A 5-gram language model was used. The "+Mono LM" scores were achieved by adding a 5-gram language model trained on the monolingual data as a separate component (its weight was determined by MERT). The last column contains the scores after adding the translation model self-trained on target monolingual data. This model was also added as another component and the weights associated with it were found by MERT.

For the back-off in the reverse self-training, we used a simple suffix-trimming heuristic suitable for fusional languages: cut off the last three characters of each word always keeping at least the first three characters. This heuristic reduces the vocabulary size to a half for Czech and Slovak but it is much less effective for Finish and German (Table 2), as can be expected from their linguistic properties.

Language	Vocabulary reduced to (%)
Czech	52
Finnish	64
German	73
Slovak	51

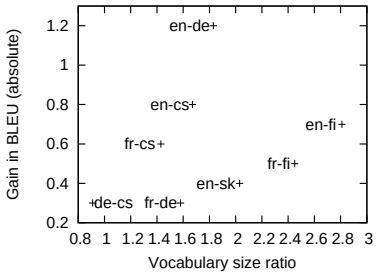
Table 2: Reduction of vocabulary size by suffix trimming

We did not use any linguistic tools, such as morphological analyzers, in this set of experiments. We see the main point of this section in illustrating the applicability of our technique on a wide range of languages, including languages for which such tools are not available.

We encountered problems when using MERT to balance the weights of the four model components. Our model consisted of 14 features – one for each language model, five for each translation model (phrase probability and lexical weight for both directions and phrase penalty), word penalty and distortion penalty. The extra 5 weights of the reversely trained translation model caused MERT to diverge in some cases. Since we used the `mert-moses.pl` script for tuning and kept the default parameters, MERT ran for 25 iterations and stopped. As a result, even though our method seemed to improve translation performance in most language pairs, several experiments contradicted this observation. We simply reran the final tuning procedure in these cases and were able to achieve an improvement in BLEU as well. These language pairs are marked with a '*' sign in Table 1.

A possible explanation for this behaviour of MERT is that the alternative decoding paths add a lot of possible derivations that generate the same string. To validate our hypothesis we examined a diverging run of MERT for English→Czech translation with two translation models. Our n-best lists contained the best 100 derivations for each translated sentence from the development data. On average (over all 1000 sentences and over all runs), the n-best list only contained 6.13 different translations of a sentence. The result of the same calculation applied on the baseline run of MERT (which converged in 9 iterations) was 34.85 hypotheses. This clear disproportion shows that MERT had much less information when optimizing our model.

Figure 2: Vocabulary ratio and BLEU score



Overall, reverse self-training seems helpful for translating into morphologically rich languages. We achieved promising gains in BLEU, even over the baseline including a language model trained on the monolingual data. The improvement ranges from roughly 0.3 (e.g. German→Czech) to over 1 point (English→German) absolute. This result also indicates that suffix trimming is a quite robust heuristic, useful for a variety of language types.

Figure 2 illustrates the relationship between vocabulary size ratio of the language pair and the improvement in translation quality. Although the points are distributed quite irregularly, a certain tendency towards higher gains with higher ratios is observable. We assume that reverse self-training is most useful in cases where a single word form in the source language can be translated as several forms in the target language. A higher ratio between vocabulary sizes suggests that these cases happen more often, thus providing more space for improvement using our method.

4.3 Data Sizes

We conducted a series of English-to-Czech experiments with fixed parallel data and a varying size of monolingual data. We used the CzEng corpus, 500 thousand parallel sentences and from 500 thousand up to 5 million monolingual sentences. We used two separate sets of 1000 sentences from CzEng for development and evaluation. Our results are summarized in Figure 3. The gains in BLEU become more significant as the size of included monolingual data increases. The highest improvement can be observed when the data are largest – over 3 points absolute. Figure 4 shows an example of the impact on translation quality – the "Mono" data are 5 million sentences.

When evaluated from this point of view, our method can also be seen as a way of considerably improving translation quality for languages with little available parallel data.

We also experimented with varying size of parallel data (500 thousand to 5 million sentences) and its effect on reverse self-training contribution. The size of monolingual data was always 5 million sentences. We first measured the percentage of test data word forms covered by the training data. We calculated the value for parallel data and for the combination of parallel and monolingual data. For word forms that appeared only in the monolingual data, a different form of the word had to be contained in the parallel data (so that the model can learn it through the back-off heuristic) in order to be counted in. The difference between the first and second value can simply be thought of as the upper-bound estimation of reverse self-training contribution. Figure 5 shows the results along with BLEU scores achieved in translation experiments following this scenario.

Our technique has much greater effect for small parallel data sizes; the amount of newly learned word forms declines rapidly as the size grows. Similarly, improvement in BLEU score decreases quickly and becomes negligible around 2 million parallel sentences.

4.4 Back-off Techniques

We experimented with several options for the back-off factor in English→Czech translation. Data from training section of CzEng were used, 1 million parallel sentences and another 5 million sentences as target-side monolingual data. As in the previous section, the sizes of our development and evaluation sets were a thousand sentences.

CzEng annotation contains lexically disambiguated word lemmas, an appealing option for our purposes. We also tried trimming the last 3 characters of each word, keeping at least the first 3 characters intact. Stemming of each word to four characters was also evaluated (Stem-4).

Table 3 summarizes our results. The last column shows the vocabulary size compared to original vocabulary size, estimated on lower-cased words.

We are not surprised by stemming performing the

Figure 3: Relation between monolingual data size and gains in BLEU score

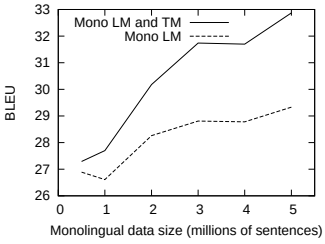
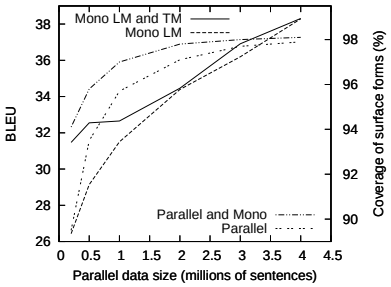


Figure 5: Varying parallel data size, surface form coverage ("Parallel", "Parallel and Mono") and BLEU score ("Mono LM", "Mono LM and TM")



System	Translation	Gloss
Baseline	Jsi tak zrcadla?	Are you$_{SG}$ so mirrors? (ungrammatical)
+Mono LM	Jsi neobjednávejte zrcadla?	Did you$_{SG}$ don't order$_{PL}$ mirrors? (ungrammatical)
+Mono TM	Už sis objednal zrcadla?	Have you$_{SG}$ ordered$_{SG}$ the mirrors (for yourself) yet?

Figure 4: Translation of the sentence "Did you order the mirrors?" by baseline systems and a reversely-trained system. Only the last one is able to generate the correct form of the word "order".

worst – the equivalence classes generated by this simple heuristic are too broad. Using lemmas seems optimal from the linguistic point of view, however suffix trimming outperformed this approach in our experiments. We feel that finding well-performing back-off techniques for other languages merits further research.

Back-off	BLEU	Vocabulary Size (%)
Baseline	31.82±3.24	100
Stem-4	32.73±3.19	19
Lemma	33.05±3.40	54
Trimmed Suffix	**33.28±3.32**	47

Table 3: Back-off BLEU scores comparison

4.5 WMT Systems

We submitted systems that used reverse self-training (`cu-tamchyna`) for English→Czech and English→German language pairs.

Our parallel data for German were constrained to the provided set (1.9 million sentences). For Czech, we used the training sections of CzEng and the supplied WMT11 News Commentary data (7.3 million sentences in total).

In case of German, we only used the supplied monolingual data, for Czech we used a large collection of texts for language modelling (i.e. unconstrained). The reverse self-training used only the constrained data – 2.3 million sentences in German and 2.2 in Czech. In case of Czech, we only used the News monolingual data from 2010 and 2011 for reverse self-training – we expected that recent data from the same domain as the test set would improve translation performance the most.

We achieved mixed results with these systems – for translation into German, reverse self-training did not improve translation performance. For Czech, we were able to achieve a small gain, even though the reversely translated data contained less sentences than the parallel data. Our BLEU scores were also affected by submitting translation outputs without normalized punctuation and with a slightly different tokenization.

In this scenario, a lot of parallel data were available and we did not manage to prepare a reversely trained model from larger monolingual data. Both of these factors contributed to the inconclusive results.

Table 4 shows case-insensitive BLEU scores as calculated in the official evaluation.

Target Language	Mono LM	+Mono TM
German	14.8	14.8
Czech	15.7	15.9

Table 4: Case-insensitive BLEU of WMT systems

5 Conclusion

We introduced a technique for exploiting monolingual data to improve the quality of translation into morphologically rich languages.

We carried out experiments showing improvements in BLEU when using our method for translating into Czech, Finnish, German and Slovak with small parallel data. We discussed the issues of including similar translation models as separate components in MERT.

We showed that gains in BLEU score increase with growing size of monolingual data. On the other hand, growing parallel data size diminishes the effect of our method quite rapidly. We also documented our experiments with several back-off techniques for English to Czech translation.

Finally, we described our primary submissions to the WMT 2011 Shared Translation Task.

References

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece, March. Association for Computational Linguistics.

Ondřej Bojar and Aleš Tamchyna. 2011. Forms Wanted: Training SMT on Monolingual Data. Abstract at Machine Translation and Morphologically-Rich Languages. Research Workshop of the Israel Science Foundation University of Haifa, Israel, January.

Ondřej Bojar and Zdeněk Žabokrtský. 2009. CzEng 0.9: Large Parallel Treebank with Rich Annotation. *Prague Bulletin of Mathematical Linguistics*, 92:63–83.

Adrià de Gispert, José B. Mariño, and Josep M. Crego. 2005. Improving statistical machine translation by classifying and generalizing inflected verb forms. In *Eurospeech 2005*, pages 3185–3188, Lisbon, Portugal.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, UK, July. Association for Computational Linguistics.

Philipp Koehn and Hieu Hoang. 2007. Factored Translation Models. In *Proc. of EMNLP*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*. The Association for Computer Linguistics.

Philipp Koehn, Alexandra Birch, and Ralf Steinberger. 2009. 462 machine translation systems for europe. In *MT Summit XII*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*, pages 388–395. ACL.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. pages 440–447, Hongkong, China, October.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Daniel Varga. 2006. The JRC-acquis: A multilingual aligned parallel corpus with 20+ languages. *CoRR*, abs/cs/0609058. informal publication.

Andreas Stolcke. 2002. Srilm — an extensible language modeling toolkit, June 06.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Semi-supervised model adaptation for statistical machine translation. *Machine Translation*, 21(2):77–94.

Chimera – Three Heads for English-to-Czech Translation

Ondřej Bojar and **Rudolf Rosa** and **Aleš Tamchyna**
Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, Prague, Czech Republic
surname@ufal.mff.cuni.cz

Abstract

This paper describes our WMT submissions CU-BOJAR and CU-DEPFIX, the latter dubbed "CHIMERA" because it combines on three diverse approaches: TectoMT, a system with transfer at the deep syntactic level of representation, factored phrase-based translation using Moses, and finally automatic rule-based correction of frequent grammatical and meaning errors. We do not use any off-the-shelf system-combination method.

1 Introduction

Targeting Czech in statistical machine translation (SMT) is notoriously difficult due to the large number of possible word forms and complex agreement rules. Previous attempts to resolve these issues include specific probabilistic models (Subotin, 2011) or leaving the morphological generation to a separate processing step (Fraser et al., 2012; Mareček et al., 2011).

TectoMT (CU-TECTOMT, Galuščáková et al. (2013)) is a hybrid (rule-based and statistical) MT system that closely follows the analysis-transfer-synthesis pipeline. As such, it suffers from many issues but generating word forms in proper agreements with their neighbourhood as well as the translation of some diverging syntactic structures are handled well. Overall, TectoMT sometimes even ties with a highly tuned Moses configuration in manual evaluations, see Bojar et al. (2011).

Finally, Rosa et al. (2012) describes Depfix, a rule-based system for post-processing (S)MT output that corrects some morphological, syntactic and even semantic mistakes. Depfix was able to significantly improve Google output in WMT12, so now we applied it on an open-source system.

Our WMT13 system is thus a three-headed creature where, hopefully: (1) TectoMT provides missing word forms and safely handles some non-parallel syntactic constructions, (2) Moses exploits very large parallel and monolingual data, and boosts better lexical choice, (3) Depfix attempts to fix severe flaws in Moses output.

2 System Description

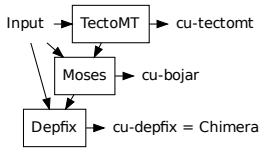


Figure 1: CHIMERA: three systems combined.

CHIMERA is a sequential combination of three diverse MT systems as depicted in Figure 1. Each of the intermediate stages of processing has been submitted as a separate primary system for the WMT manual evalution, allowing for a more thorough analysis.

Instead of an off-the-shelf system combination technique, we use TectoMT output as synthetic training data for Moses as described in Section 2.1 and finally we process its output using rule-based corrections of Depfix (Section 2.2). All steps directly use the source sentence.

2.1 Moses Setup for CU-BOJAR

We ran a couple of probes with reduced training data around the setup of Moses that proved successful in previous years (Bojar et al., 2012a).

2.1.1 Pre-processing

We use a stable pre-processing pipeline that includes normalization of quotation marks,[1] tokenization, tagging and lemmatization with tools

[1] We do not simply convert them to unpaired ASCII quotes but rather balance them and use other heuristics to convert most cases to the typographically correct form.

Case	recaser	lc→form	utc	stc
BLEU	9.05	9.13	9.70	**9.81**

Table 1: Letter Casing

included in the Treex platform (Popel and Žabokrtský, 2010).

This year, we evaluated the end-to-end effect of truecasing. Ideally, English-Czech SMT should be trained on data where only names are uppercased (and neither the beginnings of sentences, nor all-caps headlines or exclamations etc). For these experiments, we trained a simple baseline system on 1 million sentence pairs from CzEng 1.0.

Table 1 summarizes the final (case-sensitive!) BLEU scores for four setups. The standard approach is to train SMT lowercase and apply a recaser, e.g. the Moses one, on the output. Another option (denoted "lc→form") is to lowercase only the source side of the parallel data. This more or less makes the translation model responsible for identifying names and the language model for identifying beginnings of sentences.

The final two approaches attempt at "truecasing" the data, i.e. the ideal lowercasing of everything except names. Our simple unsupervised truecaser ("utc") uses a model trained on monolingual data (1 million sentences in this case, same as the parallel training data used in this experiment) to identify the most frequent "casing shape" of each token type when it appears within a sentence and then converts its occurrences at the beginnings of sentences to this shape. Our supervised truecaser ("stc") casts the case of the *lemma* on the form, because our lemmatizers for English and Czech produce case-sensitive lemmas to indicate names. After the translation, only deterministic uppercasing of sentence beginnings is needed.

We confirm that "stc" as we have been using it for a couple of years is indeed the best option, despite its unpleasingly frequent omissions of names (incl. "Spojené státy", "the United States"). One of the rules in Depfix tries to cast the case from the source to the MT output but due to alignment errors, it is not perfect in fixing these mistakes.

Surprisingly, the standard recasing worked worse than "lc→form", suggesting that two Moses runs in a row are worse than one joint search.

We consider using a full-fledged named entity recognizer in the future.

		Tokens [M]	
Corpus	Sents [M]	English	Czech
CzEng 1.0	14.83	235.67	205.17
Europarl	0.65	17.61	15.00
Common Crawl	0.16	4.08	3.63

Table 2: Basic Statistics of Parallel Data.

2.1.2 Factored Translation for Morphological Coherence

We use a quite standard factored configuration of Moses. We translate from "stc" to two factors: "stc" and "tag" (full Czech positional morphological tag). Even though tags on the target side make the data somewhat sparser (a single Czech word form typically represents several cases, numbers or genders), we do not use any back-off or alternative decoding path. A high-order language model on tags is used to promote grammatically correct and coherent output. Our system is thus less prone to errors in local morphological agreement.

2.1.3 Large Parallel Data

The main source of our parallel data was CzEng 1.0 (Bojar et al., 2012b). We also used Europarl (Koehn, 2005) as made available by WMT13 organizers.[2] The English-Czech part of the new Common Crawl corpus was quite small and very noisy, so we did not include it in our training data. Table 2 provides basic statistics of the data.

Processing large parallel data can be challenging in terms of time and computational resources required. The main bottlenecks are word alignment and phrase extraction.

GIZA++ (Och and Ney, 2000) has been the standard tool for computing word alignment in phrase-based MT. A multi-threaded version exists (Gao and Vogel, 2008), which also supports incremental extensions of parallel data by applying a saved model on a new sentence pair. We evaluated these tools and measured their wall-clock time[3] as well as the final BLEU score of a full MT system.

Surprisingly, single-threaded GIZA++ was considerably faster than single-threaded MGIZA. Using 12 threads, MGIZA outperformed GIZA++ but the difference was smaller than we expected.

Table 3 summarizes the results. We checked the difference in BLEU using the procedure by Clark et al. (2011) and GIZA++ alignments were indeed

[2] `http://www.statmt.org/wmt13/translation-task.html`

[3] Time measurements are only indicative, they were affected by the current load in our cluster.

Alignment	Wallclock Time	BLEU
GIZA++	71	**15.5**
MGIZA 1 thread	114	15.4
MGIZA 12 threads	**51**	15.4

Table 3: Rough wallclock time [hours] of word alignment and the resulting BLEU scores.

Corpus	Sents [M]	Tokens [M]
CzEng 1.0	14.83	205.17
CWC Articles	36.72	626.86
CNC News	28.08	483.88
CNA	47.00	830.32
Newspapers	64.39	1040.80
News Crawl	24.91	444.84
Total	215.93	3631.87

Table 4: Basic Statistics of Monolingual Data.

little but significantly better than MGIZA in three MERT runs.

We thus use the standard GIZA++ aligner.

2.1.4 Large Language Models

We were able to collect a very large amount of monolingual data for Czech: almost 216 million sentences, 3.6 billion tokens. Table 4 lists the corpora we used. CWC Articles is a section of the Czech Web Corpus (Spoustová and Spousta, 2012). CNC News refers to a subset of the Czech National Corpus[4] from the news domain. CNA is a corpus of Czech News Agency stories from 1998 to 2012. Newspapers is a collection of articles from various Czech newspapers from years 1998 to 2002. Finally, News Crawl is the monolingual corpus made available by the organizers of WMT13.

We created an in-domain language model from all the corpora except for CzEng (where we only used the news section). We were able to train a 4-gram language model using KenLM (Heafield et al., 2013). Unfortunately, we did not manage to use a model of higher order. The model file (even in the binarized trie format with probability quantization) was so large that we ran out of memory in decoding.[5] We also tried pruning these larger models but we did not have enough RAM.

To cater for a longer-range coherence, we trained a 7-gram language model only on the News Crawl corpus (concatenation of all years). In this case, we used SRILM (Stolcke, 2002) and pruned n-grams so that (training set) model perplexity does not increase more than 10^{-14}. The data for this LM exactly match the domain of WMT test sets.

Token	Order	Sents [M]	Tokens [M]	ARPA.gz [GB]	Trie [GB]
stc	4	201.31	3430.92	28.2	11.8
stc	7	24.91	444.84	13.1	8.1
tag	10	14.83	205.17	7.2	3.0

Table 5: LMs used in CU-BOJAR.

Finally, we model sequences of morphological tags on the target side using a 10-gram LM estimated from CzEng. Individual sections of the corpus (news, fiction, subtitles, EU legislation, web pages, technical documentation and Navajo project) were interpolated to match WMT test sets from 2007 to 2011 best. This allows even out-of-domain data to contribute to modeling of overall sentence structure. We filtered the model using the same threshold 10^{-14}.

Table 5 summarizes the resulting LM files as used in CU-BOJAR and CHIMERA.

2.1.5 Bigger Tuning Sets

Koehn and Haddow (2012) report benefits from tuning on a larger set of sentences. We experimented with a down-scaled MT system to compare a couple of options for our tuning set: the default 3003 sentences of newstest2011, the default and three more Czech references that were created by translating from German, the default and two more references that were created by post-editing a variant of our last year's Moses system and also a larger single-reference set consisting of several newstest years. The preliminary results were highly inconclusive: negligibly higher BLEU scores obtained lower manual scores. Unable to pick the best configuration, we picked the largest. We tune our systems on "bigref", as specified in Table 6. The dataset consists of 11583 source sentences, 3003 of which have 4 reference translations and a subset (1997 sents.) of which has 2 reference translations constructed by post-editing. The dataset does not include 2010 data as a heldout for other foreseen experiments.

2.1.6 Synthetic Parallel Data

Galuščáková et al. (2013) describe several possibilities of combining TectoMT and phrase-based approaches. Our CU-BOJAR uses one of the simpler but effective ones: adding TectoMT output on the test set to our training data. As a contrast to

[4] http://korpus.cz/

[5] Due to our cluster configuration, we need to pre-load language models.

English	Czech	# Refs	# Snts
newstest2011	official + 3 more from German	4	3003
newstest2011	2 post-edits of a system similar to (Bojar et al., 2012a)	2	1997
newstest2009	official	1	2525
newstest2008	official	1	2051
newstest2007	official	1	2007
Total		4	11583

Table 6: Our big tuning set (bigref).

Test Set	Used in Training	Used in Tuning	Used in Final Eval
newstest07	*could have*	en+cs	–
newstest08	en+TectoMT	en+cs	–
newstest09	*could have*	en+cs	–
newstest10	en+TectoMT	*could have*	–
newstest11	en+TectoMT	en+cs	–
newstest12	en+TectoMT	en+cs	–
newstest13	en+TectoMT	–	en+cs

Table 7: Summary of test sets usage. "en" and "cs" denote the official English and Czech sides, resp. "TectoMT" denotes the synthetic Czech.

CU-BOJAR, we also examine PLAIN Moses setup which is identical but lacks the additional synthetic phrase table by TectoMT.

In order to select the best balance between phrases suggested by TectoMT and our parallel data, we provide these data as two separate phrase tables. Each phrase table brings in its own five-tuple of scores, one of which, the phrase-penalty functions as an indicator how many phrases come from which of the phrase tables. The standard MERT is then used to optimize the weights.[6,7]

We use one more trick compared to Galuščáková et al. (2013): we deliberately overlap our training and tuning datasets. When preparing the synthetic parallel data, we use the English side of newstests 08 and 10–13. The Czech side is always produced by TectoMT. We tune on bigref (see Table 6), so the years 08, 11 and 12 overlap. (We could have overlapped also years 07, 09 and 10 but we had them originally reserved for other purposes.) Table 7 summarizes the situation and highlights that our setup is fair: we never use the target side of our final evaluation set newstest2013. Some test sets are denoted "*could have*" as including them would still be correct.

The overlap allows MERT to estimate how useful are TectoMT phrases compared to the standard phrase table not just in general but on the specific foreseen test set. This deliberate overfitting to newstest 08, 11 and 12 then helps in translating newstest13.

This combination technique in its current state is rather expensive as a new phrase table is required for every new input document. However, if we fix the weights for the TectoMT phrase table, we can avoid re-tuning the system (whether this would degrade translation quality needs to be empirically evaluated). Moreover, if we use a dynamic phrase table, we could update it with TectoMT outputs on the fly, thus bypassing the need to retrain the translation model.

[6] Using K-best batch MIRA (Cherry and Foster, 2012) did not work any better in our setup.

[7] We are aware of the fact that Moses alternative decoding paths (Birch and Osborne, 2007) with similar phrase tables clutter n-best lists with identical items, making MERT less stable (Eisele et al., 2008; Bojar and Tamchyna, 2011). The issue was not severe in our case, CU-BOJAR needed 10 iterations compared to 3 iterations needed for PLAIN.

2.2 Depfix

Depfix is an automatic post-editing tool for correcting errors in English-to-Czech SMT. It is applied as a post-processing step to CU-BOJAR, resulting in the CHIMERA system. Depfix 2013 is an improvement of Depfix 2012 (Rosa et al., 2012).

Depfix focuses on three major types of language phenomena that can be captured by employing linguistic knowledge but are often hard for SMT systems to get right:

- morphological agreement, such as:
  - an adjective and the noun it modifies have to share the same morphological gender, number and case
  - the subject and the predicate have to agree in morphological gender, number and person, if applicable
- transfer of meaning in cases where the same meaning is expressed by different grammatical means in English and in Czech, such as:
  - a subject in English is marked by being a left modifier of the predicate, while in Czech a subject is marked by the nominative morphological case
  - English marks possessiveness by the preposition 'of', while Czech uses the genitive morphological case
  - negation can be marked in various ways in English and Czech
- verb-noun and noun-noun valency—see (Rosa et al., 2013)

Depfix first performs a complex lingustic anal-

System	BLEU	TER	WMT Ranking	
			Appraise	MTurk
CU-TECTOMT	14.7	0.741	0.455	0.491
CU-BOJAR	**20.1**	0.696	0.637	**0.555**
CU-DEPFIX	20.0	**0.693**	**0.664**	0.542
PLAIN Moses	19.5	0.713	–	–
GOOGLE TR.	–	–	0.618	0.526

Table 8: Overall results.

ysis of both the source English sentence and its translation to Czech by CU-BOJAR. The analysis includes tagging, word-alignment, and dependency parsing both to shallow-syntax ("analytical") and deep-syntax ("tectogrammatical") dependency trees. Detection and correction of errors is performed by rule-based components (the valency corrections use a simple statistical valency model). For example, if the adjective-noun agreement is found to be violated, it is corrected by projecting the morphological categories from the noun to the adjective, which is realized by changing their values in the Czech morphological tag and generating the appropriate word form from the lemma-tag pair using the rule-based generator of Hajič (2004).

Rosa (2013) provides details of the current version of Depfix. The main additions since 2012 are valency corrections and lost negation recovery.

3 Overall Results

Table 8 reports the scores on the WMT13 test set. BLEU and TER are taken from the evaluation web site[8] for the *normalized* outputs, case insensitive. The normalization affects typesetting of punctuation only and greatly increases automatic scores. "WMT ranking" lists results from judgments from Appraise and Mechanical Turk. Except CU-TECTOMT, the manual evaluation used non-normalized MT outputs. The figure is the WMT12 standard interpretation as suggested by Bojar et al. (2011) and says how often the given system was ranked better than its competitor across all 18.6k non-tying pairwise comparisons extracted from the annotations.

We see a giant leap from CU-TECTOMT to CU-BOJAR, confirming the utility of large data. However, CU-TECTOMT had something to offer since it improved over PLAIN, a very competitive baseline, by 0.6 BLEU absolute. Depfix seems to slightly worsen BLEU score but slightly improve TER; the manual evaluation is similarly indecisive.

[8] http://matrix.statmt.org/

System	# Tokens	% Tokens
All	22920	76.44
Moses	3864	12.89
TectoMT	2323	7.75
Other	877	2.92

Table 9: CHIMERA components that contribute "confirmed" tokens.

System	# Tokens	% Tokens
None	21633	79.93
Moses	2093	7.73
TectoMT	2585	9.55
Both	385	1.42
CU-BOJAR	370	1.37

Table 10: Tokens missing in CHIMERA output.

4 Combination Analysis

We now closely analyze the contributions of the individual engines to the performance of CHIMERA. We look at translations of the newstest2013 sets produced by the individual systems (PLAIN, CU-TECTOMT, CU-BOJAR, CHIMERA).

We divide the newstest2013 reference tokens into two classes: those successfully produced by CHIMERA (Table 9) and those missed (Table 10). The analysis can suffer from false positives as well as false negatives, a "confirmed" token can violate some grammatical constraints in MT output and an "unconfirmed" token can be a very good translation. If we had access to more references, the issue of false negatives would decrease.

Table 9 indicates that more than 3/4 of tokens confirmed by the reference were available in all CHIMERA components: PLAIN Moses, CU-TECTOMT alone but also in the subsequent combinations CU-BOJAR and the final CU-DEPFIX.

PLAIN Moses produced 13% tokens that TectoMT did not provide and TectoMT output roughly 8% tokens unknown to Moses. However, note that it is difficult to distinguish the effect of different model weights: PLAIN *might have* produced some of those tokens as well if its weights were different. The row "Other" includes cases where e.g. Depfix introduced a confirmed token that none of the previous systems had.

Table 10 analyses the potential of CHIMERA components. These tokens from the reference were *not* produced by CHIMERA. In almost 80% of cases, the token was not available in any 1-best output; it *may* have been available in Moses phrase

tables or the input sentence.

TectoMT offered almost 10% of missed tokens, but these were not selected in the subsequent combination. The potential of Moses is somewhat lower (about 8%) because our phrase-based combination is likely to select wordings that score well in a phrase-based model. 385 tokens were suggested by both TectoMT and Moses alone, but the combination in CU-BOJAR did not select them, and finally 370 tokens were produced by the combination while they were *not* present in 1-best output of neither TectoMT nor Moses. Remember, all these tokens eventually did not get to CHIMERA output, so Depfix must have changed them.

4.1 Depfix analysis

Table 11 analyzes the performance of the individual components of Depfix. Each *evaluated* sentence was either *modified* by a Depfix component, or not. If it was *modified*, its quality could have been evaluated as better (*improved*), worse (*worsened*), or the same (*equal*) as before. Thus, we can evaluate the performance of the individual components by the following measures:[9]

$$precision = \frac{\#improved}{\#improved+\#worsened} \quad (1)$$

$$impact = \frac{\#modified}{\#evaluated} \quad (2)$$

$$useless = \frac{\#equal}{\#modified} \quad (3)$$

Please note that we make an assumption that if a sentence was modified by multiple Depfix components, they all have the same effect on its quality. While this is clearly incorrect, it is impossible to accurately determine the effect of each individual component with the evaluation data at hand. This probably skews especially the reported performance of "high-impact" components, which often operate in combination with other components.

The evaluation is computed on 871 hits in which CU-BOJAR and CHIMERA were compared.

The results show that the two newest components – Lost negation recovery and Valency model – both modify a large number of sentences. Valency model seems to have a slightly *negative* effect on the translation quality. As this is the only statistical component of Depfix, we believe that this is caused by the fact that its parameters were not tuned on the final CU-BOJAR system, as the tuning has to be done semi-manually and the final system was not available in advance. On the other hand, Lost negation recovery seems to have a highly positive effect on translation quality. This is to be expected, as a lost negation often leads to the translation bearing an opposite meaning to the original one, which is probably one of the most serious errors that an MT system can make.

[9] We use the term *precision* for our primary measure for convenience, even though the way we define it does not match exactly its usual definition.

Depfix component	Prc.	Imp.	Usl.
Aux 'be' agr.	–	1.4%	100%
No prep. without children	–	0.5%	100%
Sentence-initial capitalization	0%	0.1%	0%
Prepositional morph. case	0%	2.1%	83%
Preposition - noun agr.	40%	3.8%	70%
Noun number projection	41%	7.2%	65%
Valency model	48%	10.6%	66%
Subject - nominal pred. agr.	50%	3.8%	76%
Noun - adjective agr.	55%	17.8%	75%
Subject morph. case	56%	8.5%	57%
Tokenization projection	56%	3.0%	38%
Verb tense projection	58%	5.2%	47%
Passive actor with 'by'	60%	1.0%	44%
Possessive nouns	67%	0.9%	25%
Source-aware truecasing	67%	2.8%	50%
Subject - predicate agr.	68%	5.1%	57%
Pro-drop in subject	73%	3.4%	63%
Subject - past participle agr.	75%	6.3%	42%
Passive - aux 'be' agr.	77%	4.8%	69%
Possessive with 'of'	78%	1.5%	31%
Present continuous	78%	1.5%	31%
Missing reflexive verbs	80%	1.6%	64%
Subject categories projection	83%	3.7%	62%
Rehang children of aux verbs	83%	5.5%	62%
Lost negation recovery	90%	7.2%	38%

Table 11: Depfix components performance analysis on 871 sentences from WMT13 test set.

5 Conclusion

We have reached our chimera to beat Google Translate. We combined all we have: a deep-syntactic transfer-based system TectoMT, very large parallel and monolingual data, factored setup to ensure morphological coherence, and finally Depfix, a rule-based automatic post-editing system that corrects grammaticality (agreement and valency) of the output as well as some features vital for adequacy, namely lost negation.

Acknowledgments

This work was partially supported by the grants P406/11/1499 of the Grant Agency of the Czech Republic, FP7-ICT-2011-7-288487 (MosesCore) and FP7-ICT-2010-6-257528 (Khresmoi) of the European Union and by SVV project number 267 314.

References

Alexandra Birch and Miles Osborne. 2007. CCG Supertags in Factored Statistical Machine Translation. In *In ACL Workshop on Statistical Machine Translation*, pages 9–16.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving Translation Model by Monolingual Data. In *Proc. of WMT*, pages 330–336. ACL.

Ondřej Bojar, Miloš Ercegovčević, Martin Popel, and Omar Zaidan. 2011. A Grain of Salt for the WMT Manual Evaluation. In *Proc. of WMT*, pages 1–11. ACL.

Ondřej Bojar, Bushra Jawaid, and Amir Kamran. 2012a. Probes in a Taxonomy of Factored Phrase-Based Models. In *Proc. of WMT*, pages 253–260. ACL.

Ondřej Bojar, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. 2012b. The Joy of Parallelism with CzEng 1.0. In *Proc. of LREC*, pages 3921–3928. ELRA.

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proc. of NAACL/HLT*, pages 427–436. ACL.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proc. of ACL/HLT*, pages 176–181. ACL.

Andreas Eisele, Christian Federmann, Hervé Saint-Amand, Michael Jellinghaus, Teresa Herrmann, and Yu Chen. 2008. Using Moses to Integrate Multiple Rule-Based Machine Translation Engines into a Hybrid System. In *Proc. of WMT*, pages 179–182. ACL.

Alexander Fraser, Marion Weller, Aoife Cahill, and Fabienne Cap. 2012. Modeling Inflection and Word-Formation in SMT. In *Proc. of EACL 2012*. ACL.

Petra Galuščáková, Martin Popel, and Ondřej Bojar. 2013. PhraseFix: Statistical Post-Editing of TectoMT. In *Proc. of WMT13*. Under review.

Qin Gao and Stephan Vogel. 2008. Parallel Implementations of Word Alignment Tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57. ACL.

Jan Hajič. 2004. *Disambiguation of rich inflection: computational morphology of Czech*. Karolinum.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable Modified Kneser-Ney Language Model Estimation. In *Proc. of ACL*.

Philipp Koehn and Barry Haddow. 2012. Towards Effective Use of Training Data in Statistical Machine Translation. In *Proc. of WMT*, pages 317–321. ACL.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Machine Translation Summit X*, pages 79–86.

David Mareček, Rudolf Rosa, Petra Galuščáková, and Ondřej Bojar. 2011. Two-step translation with grammatical post-processing. In *Proc. of WMT*, pages 426–432. ACL.

Franz Josef Och and Hermann Ney. 2000. Improved Statistical Alignment Models. In *ACL*. ACL.

Martin Popel and Zdeněk Žabokrtský. 2010. TectoMT: Modular NLP Framework. In Hrafn Loftsson, Eirikur Rögnvaldsson, and Sigrun Helgadottir, editors, *IceTAL 2010*, volume 6233 of *Lecture Notes in Computer Science*, pages 293–304. Iceland Centre for Language Technology (ICLT), Springer.

Rudolf Rosa, David Mareček, and Ondřej Dušek. 2012. DEPFIX: A system for automatic correction of Czech MT outputs. In *Proc. of WMT*, pages 362–368. ACL.

Rudolf Rosa, David Mareček, and Aleš Tamchyna. 2013. Deepfix: Statistical Post-editing of Statistical Machine Translation Using Deep Syntactic Analysis. Bălgarska akademija na naukite, ACL.

Rudolf Rosa. 2013. Automatic post-editing of phrase-based machine translation outputs. Master's thesis, Charles University in Prague, Faculty of Mathematics and Physics, Praha, Czechia.

Johanka Spoustová and Miroslav Spousta. 2012. A High-Quality Web Corpus of Czech. In *Proc. of LREC*. ELRA.

Andreas Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. Intl. Conf. on Spoken Language Processing*, volume 2, pages 901–904.

Michael Subotin. 2011. An exponential translation model for target language morphology. In *Proc. of ACL/HLT*, pages 230–238. ACL.

PBML

The Prague Bulletin of Mathematical Linguistics
NUMBER 95 APRIL 2011 63–76

Analyzing Error Types in English-Czech Machine Translation

Ondřej Bojar

Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

Abstract

This paper examines two techniques of manual evaluation that can be used to identify error types of individual machine translation systems. The first technique of "blind post-editing" is being used in WMT evaluation campaigns since 2009 and manually constructed data of this type are available for various language pairs. The second technique of explicit marking of errors has been used in the past as well.

We propose a method for interpreting blind post-editing data at a finer level and compare the results with explicit marking of errors. While the human annotation of either of the techniques is not exactly reproducible (relatively low agreement), both techniques lead to similar observations of differences of the systems. Specifically, we are able to suggest which errors in MT output are easy and hard to correct with no access to the source, a situation experienced by users who do not understand the source language.

1. Introduction

The Workshop on Statistical Machine Translation (WMT)[1] is a yearly open competition in machine translation (MT) among a few languages. Regularly, system outputs are manually judged using various techniques with the side-effect of establishing a trustworthy set of manual and automatic metrics (Callison-Burch et al., 2008, 2009). The manual evaluation methods tested so far are rather black-box, allowing to rank systems but revealing little or nothing about the types of errors in state-of-the-art MT.

A ranked list of error types of a system would be an invaluable resource for the developers of the system. In this paper, we use the WMT09 manual evaluation data

[1] `http://www.statmt.org/wmt06` to `wmt10`

 Corresponding author: `bojar@ufal.mff.cuni.cz`
Cite as: Ondřej Bojar. Analyzing Error Types in English-Czech Machine Translation . The Prague Bulletin of Mathematical Linguistics No. 95, 2011, pp. 63–76. doi: 10.2478/v10108-011-0005-2.

and our manual evaluation to identify error types in outputs of four English-to-Czech MT systems. Both techniques lead to similar results and we observe expectable but interesting differences in errors the systems make.

1.1. Techniques of Manual MT Evaluation

Traditionally, MT output has been manually judged by ranking of sentences in terms of adequacy and fluency. In WMT, the two axes of ranking were joined to a single one in 2008 due to a low inter-annotator agreement (Callison-Burch et al., 2008). Since 2009, WMT extends the sentence ranking with so-called "blind post-editing". The blind post-editing is performed by two separate persons in a row: the first one (the "editor") gets only the system output and is asked to produce a fluent sentence conveying the same message, the second one (the "judge") gets the edited sentence along with the source and the reference translation to confirm whether it is still an acceptable translation.

While the sentence ranking is hard to use for analysis of errors of individual systems, the blind post-editing provides a better chance. In Section 3, we design a simple technique for searching for MT errors given post-edits and apply it to four systems translating from English to Czech.

To support the observations, we also carry out an additional manual analysis: flagging of errors in MT output, see Section 4. This is a finer variant of post-editing and allows us to identify clear differences between types of MT systems in terms of errors they make. By linking the two types of manual evaluation, we are even able to observe that the systems differ in the possibility to correct particular error types in the blind post-editing task. Errors hard to fix in this setting are the most risky when the system is used by a user who does not understand the source language.

2. Brief Overview of Systems Examined

In the paper, we consider only a small subset of WMT09 systems. Still, they represent a wide range of technologies:

Google is a commercial statistical MT system trained on unspecified amounts and sources of parallel and monolingual texts.

PC Translator is a traditional commercial MT system tuned for years primarily for English-to-Czech translation.

TectoMT is an experimental system following the traditional analysis-transfer-synthesis scenario with the transfer implemented at the deep syntactic layer of language representation, based on the theory of Functional Generative Description (Sgall et al., 1986) as implemented in the Prague Dependency Treebank (Hajič et al., 2006). For the purposes of TectoMT, the tectogrammatical layer was further simplified (Žabokrtský et al., 2008; Bojar et al., 2009).

Ondřej Bojar Analyzing Error Types in English-Czech Machine Translation (63–76)

System	PC Translator	Google	CU-Bojar	TectoMT
Ranked ≥ others	**67%**	66%	61%	48%
Edits deemed acceptable	**32%**	**32%**	21%	19%
BLEU	**.14**	**.14**	**.14**	.07
NIST	4.34	4.96	**5.18**	4.17

Table 1. Manual and automatic scores of the four MT systems examined. Best results in bold.

CU-Bojar is an experimental phrase-based system the core of which is the Moses[2] decoder (Koehn et al., 2007). Considerable effort has been invested in tuning the system for English-to-Czech translation (Bojar et al., 2009).

Table 1 compares these systems on the WMT09 dataset using some of WMT09 evaluation metrics as reported in Callison-Burch et al. (2009). We see that TectoMT was distinctly worse than the other systems and that the two commercial systems perform better than the research ones. The traditional automatic metrics BLEU and NIST partially fail to predict this.

3. Exploiting Blind Post-Edits

As outlined above, the "blind post-editing" WMT dataset consists of source sentences, MT system outputs (also called hypotheses), edited outputs (also called edits) and yes/no acceptability judgments. Naturally, there is also the reference translation but its relation to the MT output is rather loose. Most of the relations in the dataset are one-to-many: There are always more MT systems for a single input sentence (each system provides a single best candidate), there are usually several manual edits of a given hypothesis and several judgment of a given edit.

The dataset is blind in several ways: the editor knows only the text of the hypothesis and neither the system, source text nor the reference translation. The annotator does not know the system or the editor either.

The edits are completely unrestricted and not formalized. All we have are two strings: the hypothesis and the edit. Editors are allowed to rewrite the sentence from scratch (but they usually don't have the capacity to do so because they don't know more than what is in the sentence).

3.1. Basic Statistics of the Dataset

The dataset consists of 100 source sentences. For the four systems in question, 29 unique editors provided the total of 1198 edits out of which only 708 (59%) contain a

[2] http://www.statmt.org/moses

new string.[3] Others were left unedited either because they were not comprehensible at all or because they were deemed correct. We are aware of the possible bias in our error analysis caused by ignoring esp. the incomprehensible sentences. The method discussed here is unfortunately not applicable to such cases, however the flagging of errors as described in Section 4 covers all the 100 sentences. In the sequel, we focus solely on the 708 edits.

The 708 edits were judged by 20 annotators, leading to the total of 2762 items (41% of which are marked as acceptable). In the sequel, we fully multiply the dataset so that an input sentence is duplicated as many times as any edit of any of the outputs was judged. This corresponds to micro-averaging all the observations over the dataset.

The average sentence length of a hypothesis is 21.4±9.8 words and the average sentence length of an edit is 20.6±9.3 words.

3.2. Generalizing Edits

In order to learn types of errors frequently done by individual MT systems, we need to somehow generalize the actual modifications performed in the edits. We use the following simple procedure:

1. Tokenize and morphologically analyze both the hypothesis and the edit.
2. Find differences between the two sequences of tokens. Various techniques can be applied here, we use the longest common subsequence algorithm (LCS, Hunt and McIlroy (1976)) as implemented in the Perl module `Algorithm::Diff` and the Unix `diff` tool. In future we would like to model block movements in the alignment as e.g. TER (Snover et al., 2009) or CDER (Leusch and Ney, 2008) do.
3. Synchronously traverse the tokens as aligned by the diff algorithm. Each step in the traversal is called a "hunk" and corresponds to an atomic edit.
4. Collect frequencies of seen types of hunks.

Figure 1 illustrates a hypothesis and an edit. There are four basic types of hunks, with the total frequencies given in Table 2: about 40k hunks link two identical tokens (Match)[4], 7k tokens were deleted from the hypothesis (Delete) and 5k were inserted (Insert). For about 12k tokens the LCS algorithms found sufficient context to mark them as being a substitute for each other (Modify). As we see in Table 2, individual edits vary a lot in terms of the number of these coarse hunk types. The edits that were approved in the second stage contain somewhat fewer matched tokens but the average sentence length for these edits is also slightly lower: 20.1±9.1. We would like to attribute this to a negative correlation between a hypothesis length and the acceptability of its edits (the percentage of judges who accepted the edit) but the correlation is rather weak: Pearson correlation coefficient of -0.13.

[3] One of the sentences had only the uninformative edits so we were left with 99 sentences.

[4] Actually, 1396 of these hunks have the same form but the morphological analyzer tagged them differently. We still count them as Match.

Ondřej Bojar Analyzing Error Types in English-Czech Machine Translation (63–76)

	Hunk	Hypothesis	Gloss	Edit	Gloss
1		Globální	Global	Globální	
2		finanční	finance	finanční	
3		krize	crisis.fem	krize	
4		je	is	je	
5		významně	notably	významně	
6	Modify	ovlivňoval	influenced.*masc*	ovlivňovala	influenced.*fem*
7		na	at	na	
8		akciových	stock	akciových	
9		trzích	markets	trzích	
10		,	,	,	
11		které	that	které	
12	Modify	se	*aux-refl*	prudce	quickly
13	Modify	pouštějí	send out	padají	fall
14	Delete	ostře	sharply	—	—
15		.	.	.	

Figure 1. Sample hypothesis and an edit, aligned using the LCS algorithm. Most of the hunks are "Match".

	Match	Delete	Insert	Modify
Total	39604	7176	4847	12261
Avg. per approved edit	13.4±6.6	2.5±2.6	1.8±1.9	4.2±3.2
Avg. per disapproved edit	15.0±7.0	2.6±2.9	1.7±2.0	4.6±3.3

Table 2. Coarse hunk types in the dataset of 99 input sentences with a valid edit.

3.3. Interpreting Hunks

As illustrated in Figure 1, the coarse hunk types do not always correspond to the change performed. The hunk 6 is an excellent example and we can directly derive the change from it. On the other hand, the hunks 12 to 14 are misaligned for our purposes. What actually happened was that the superfluous reflexive particle *se* got deleted, the lexical value of the verb got changed and the order of the adverb and the verb got swapped. For the purposes of this evaluation, we re-interpret only the Modify hunks handling the reflexive particle as a pair of Insert and Delete hunks.

Table 3 indicates how often a specific hunk class occurred in edits of an MT system output. We group hunks to the following classes:

Word matched if the form of the word is left unchanged (regardless a possible change in the automatically produced lemma or morphological tag).

Hunk Class	Count *% Approved*	CU-Bojar	TectoMT	Google	PC Translator
Word matched	39604	9781	7158	11176	11489
	38.5	*33.3*	*30.5*	*48.0*	*38.6*
Fix morphology only	2545	693	538	638	676
	33.6	*37.4*	*26.4*	*33.1*	*35.8*
Fix lexical choice, loose	1828	203	556	445	624
	39.5	*29.1*	*34.7*	*44.3*	*43.8*
Delete POS: N	1694	382	413	464	435
	39.1	*29.6*	*39.0*	*50.0*	*36.1*
Insert POS: N	1352	279	373	305	395
	41.8	*36.6*	*37.3*	*55.1*	*39.5*
Delete POS: V	1293	190	303	289	511
	40.8	*32.6*	*33.7*	*58.5*	*38.0*
Fix lexical choice, strict	1152	211	357	181	403
	37.8	*27.5*	*28.0*	*46.4*	*48.1*
Insert POS: V	990	199	179	212	400
	40.1	*38.2*	*33.5*	*51.9*	*37.8*
…					
Delete reflexive particle	437	97	132	110	98
	35.0	*23.7*	*17.4*	*61.8*	*39.8*
…					
Insert reflexive particle	385	41	67	99	178
	40.8	*24.4*	*29.9*	*52.5*	*42.1*
…					
Fix capitalization only	102	43	11	3	45
	31.4	*34.9*	*27.3*	*0.0*	*31.1*

Table 3. Most frequent hunk classes per system.

Fix capitalization only if the only difference between the word in the edit and the hypothesis is letter case.

Fix morphology only if the lemma of word is preserved but there is a change in the word form.

Fix lexical choice if the morphological tag is preserved but the lemma changes. We distinguish two subclasses: strict fix requires the exact same morphological tag[5] while loose fix requires only the identity of the part of speech.

Insert or delete reflexive particle if the Czech auxiliary particle *se* or *si* gets inserted or deleted. The particle is interesting because it is rather important for correct sense discrimination of some verbs but it is often placed at the second position in the sentence, possibly far away from the verb. In statistical MT systems, this

[5]This is an underestimate because the tagset sometimes uses a special value of a category indicating one of several possible simple values. The proper handling would thus be to unify the tags, not check them for identity.

particle gets often mis-aligned to some English auxiliary, e.g. *is*, and is spuriously produced in MT output.

Insert or delete words of various parts of speech, e.g. nouns (N) or verbs (V).

As we see in Table 3, the most frequent fix is related to pure change of morphology. This is a natural results because Czech has a very rich morphology and choosing the correct word form is the hardest part of English-to-Czech MT. In 33.6% of edits that included this type of fix, the second annotator approved the edit as a valid translation. Individual MT systems differ in the frequency this type of fix was applied: CU-Bojar and PC Translator needed a fix of the morphology most often. Google (thanks to its large n-gram language model) performed better in terms of necessary fixes but poorer in terms of acceptability of sentences with such a fix.

The fewest fixes of morphology were needed for TectoMT, a system that generates the target word forms using a deterministic morphological generator.

PC Translator seems to have the worst lexical choice (both strict and loose) followed by TectoMT. We are not surprised to see that CU-Bojar and Google need far fewer fixes of lexical choice as n-gram language models and longer phrases handle at least local lexical coherence well.

The acceptability judgments of edits with the following hunk classes are also noteworthy: fixing morphology in Google output is harder (leads to fewer edits accepted) than fixing lexical choice while quite the opposite holds for CU-Bojar. Again, we tend to attribute the difference to the language model size where it failed to guide CU-Bojar to the correct form and it misled Google to producing sequences output of bad words.

The reflexive particle was superfluously produced by TectoMT most often. Sentences with the superfluous particle were hard to correct (low acceptability rate) for TectoMT, where the sentence structure was probably distorted altogether, and easy to correct for Google, where the *se* was probably inserted as a mis-translation of an English auxiliary word.

Another frequent type of fixes is the insertion and deletion of nouns and verbs. We assume that a significant portion of these cases are word movements. Finally, we see that pure capitalization fixes are rare.

4. Flagging of Errors

To complement the manual judgments of WMT09, we carried out an additional manual evaluation of the four systems by marking errors in their output. We used an error classification inspired by Vilar et al. (2006), see Figure 2. Note that our annotators do not provide us with the full text of a corrected version of the hypothesis. Given our current experience, we believe that each of the annotators implicitly uses some "target acceptable output" and marks the changes necessary to reach it. Unlike in e.g. HTER (Snover et al., 2009), we have not recorded these target acceptable outputs in this exercise.

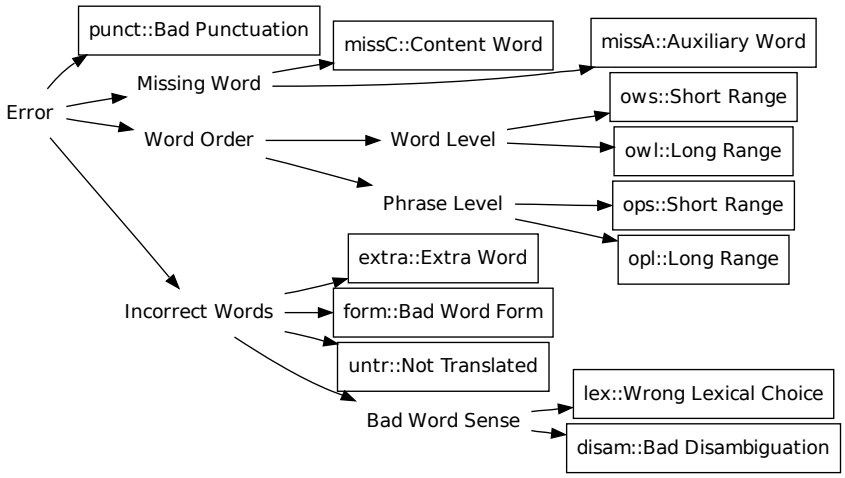


Figure 2. Error classification for manual flagging of errors. Boxes indicate the error flags used in our annotation.

Words appearing in the hypotheses can be marked as wrong for several reasons: they may not be translated despite they should be (untr), they may convey wrong meaning (Bad Word Sense; see below for details), they may be expressed in a bad morphological form (form) or they may be simply superfluous (extra). The annotators can add words that should have been in the hypothesis but they are missing (missC and missA). The set of allowed flags also covers some less important errors like punctuation or various types of word order issues. Short-range flags indicate that swapping a single unit with the next one would fix the problem, long-range flags indicate that the unit should be moved somewhere further away. If the misplaced words form a contiguous sequence ("phrase"), only one flag for the whole sequence should be used.

We used 200 sentences in total and 100 of them were the same sentences as annotated in the blind post-editing task. The annotation was carried out by 18 native Czech speakers to share the workload. Most of the sentences were annotated twice, 14% were annotated three times and 9% only once.

The instruction was to annotate as few errors as necessary to change the hypothesis to an acceptable output. An example of the annotation is given in Figure 3.[6] Unlike

Source	Perhaps there are better times ahead.
Reference	Možná se tedy blýská na lepší časy.
Gloss	*Perhaps it is flashing for better times.*
	Možná, že **extra::**tam jsou lepší **disam::**krát **lex::**dopředu.
	Perhaps, that there are better multiply to-front.
	Možná **extra::**tam jsou příhodnější časy vpředu.
	Perhaps there are favorable times in-front.
missC::v_budoucnu	Možná **form::**je lepší časy.
missC::in-future	*Perhaps is better times.*
	Možná jsou lepší časy **lex::**vpřed.
	Perhaps are better times to-front.

Figure 3. Flagging errors in outputs of four MT systems. English glosses are provided only for illustration purposes.

in the WMT09 blind post-editing, our annotators had access to the source and the reference. The identity of the MT system was hidden.

4.1. Agreement When Flagging Errors

The agreement when flagging tokens is relatively low. Excluding sentences with a single annotation, there were 5905 tokens flagged by at least one annotator. 43.6% of these tokens were flagged by all (two or three) annotators, regardless the number or type of error flags.

We attribute the low agreement to the fact that the annotators often diverge in the target acceptable output as well as in the set of marked corrections that lead to the target output. The agreement also drops if one of the annotators is willing to accept even slightly distorted output or forgets to mark some errors.

Table 4 provides the agreement for individual flag types on sentences with exactly two annotations. The highest agreement is achieved when labeling words not translated by the system but it is still surprisingly low. The flag `neg` was used by some annotators as a refinement of a bad form. We merge it with `form` annotations in other evaluations but we see that the agreement about negation is reasonable. The very low agreement in `case`, `opl` and `ops` is caused by only few annotators marking errors of this type.

We expected the `disam` and `lex` categories to be hard to distinguish. Disambiguation errors mean that the system has "misunderstood" the source word and picked a

[6] To avoid any systematic distortion of systems' outputs, our annotators were required to preserve the original space-delimited tokens. Several flags could have been assigned to a single token and this was often the case of tokens containing inappropriate punctuation, e.g. "I punct::form::doesn't, sleep." Some annotators also added special error marks for other minor errors such as letter case and bad tokenization. A few judgments also indicated that the sentence is totally wrong and not word marking individual errors (1 for PC Translator, 4 for Google and 6 for CU-Bojar and TectoMT).

Flag Type	Flagged by One	Flagged by Two	Agreement	Flag Type	Flagged by One	Flagged by Two	Agreement
untr	61	72	54.1	tok	24	4	14.3
neg	8	7	46.7	owl	116	17	12.8
extra	461	345	42.8	lex	559	63	10.1
form	1009	625	38.2	case	73	4	5.2
disam	912	310	25.4	opl	23	0	0
punct	304	98	24.4	ops	57	0	0
ows	258	69	21.1	Any	2614	2323	47.0

For each flag type we count tokens annotated by only one of two annotators and by both of them. Agreement = Two/(One + Two)

Table 4. Tokens flagged by one or two annotators.

clearly distinct wrong sense. All other (unexplained) bad lexical choices were marked `lex`. As we see, the agreement for `lex` is indeed very low. If we treat `lex` and `disam` as a single category, the agreement rises to 39.7%, more than the flag for erroneous word form.

In the following, we use all items that were flagged by any annotator. If a word is marked with the same flag by two annotators, we count it as two items.

4.2. Error Types by Individual MT Systems

Table 5 documents an important difference in error types made by individual systems. While CU-Bojar produced the fewest words with a bad sense (587), it missed by far the most content words (199). This is in line with the high score of the system in terms of NIST or BLEU and lower manual scores (see Table 1). Given the underlying technology, it also suggests a certain overfitting in the tuning of the underlying log-linear model, e.g. the penalty for producing a word set too high. On the other end of the scale is PC Translator which had the fewest content words missing (42) but did not score particularly well in terms of lexical choice (800). Google seems to choose a good balance (72 missed content words, 670 wrong lexical choices).

We also see that systems with n-gram LMs perform better for some less serious phenomena like local word order (ows) and punctuation (punct).

Finally note that the overall number of errors or serious errors marked by humans does not correlate with other manual evaluations (Table 1). The number of errors marked in PC Translator's output, the best ranked system, was higher than e.g. Google. Admittedly, the set of flagged sentences is not the same but still it comes from exactly the same test set of WMT09 and covers the blind post-editing subset. This again indicates, how difficult the evaluation of MT is even for humans.

	Google	CU-Bojar	PC Translator	TectoMT	Total
disam	406	379	569	659	2013
lex	211	208	231	340	990
Total bad word sense	617	587	800	999	3003
missA	84	111	96	138	429
missC	72	199	42	108	421
Total missed words	156	310	138	246	850
form	783	735	762	713	2993
extra	381	313	353	394	1441
untr	51	53	56	97	257
Total serious errors	1988	1998	2109	2449	8544
ows	117	100	157	155	529
punct	115	117	150	192	574
owl	43	57	50	44	194
ops	26	14	25	15	80
letter case	13	45	24	21	103
opl	10	11	11	13	45
tokenization	7	12	10	6	35
Total errors	2319	2354	2536	2895	10104

Table 5. Flagged errors by type and system.

4.3. Errors Easy and Hard to Fix in Blind Post-Editing

Table 6 indicates which errors of a particular system are easy to fix in blind post-editing and which are particularly hard. The higher the number, the easier to fix errors of that kind. We obtained the scores as the difference in error distributions in top and bottom 25% of sentences when sorted by the average acceptability of post-edits of the sentence.[7] For instance, 30.30% of errors made by Google in 25% most easily post-editable sentences were errors in form. The percentage of errors in form rises to 32.90% if we look at 25% sentences that were hardest to post-edit. Table 6 shows the difference of these figures, indicating that errors in form by Google are relatively hard to fix (-2.60) in blind post-editing.

This kind of evaluation confirms our expectations about similarities and differences of the examined MT systems and it is in accordance with the post-edits alone, see Section 3.3: lexical choice is a problem hard to fix for every system. Although the "lex" category is very similar to "disam", they were probably easy to distinguish in the output of TectoMT: we know that TectoMT's dictionary is not clean and often

[7] As we know from previous section, each edit was judged by several judges. We denote the percentage of approvals as the "acceptability" of an edit and average those numbers over all edits of a hypothesis. Note that the order of sentences by the average acceptability of its post-edits is different for each system.

System	Easy to Fix	Hard to Fix
CU-Bojar	form (11.0), tok (3.3), punct (2.9)	disam (-4.0), extra (-4.9), lex (-5.8)
TectoMT	missA (4.4), disam (4.2), ows (2.2)	untr (-1.6), missC (-2.3), lex (-7.3)
Google	missA (6.6), punct (6.1), ows (3.5)	form (-2.6), missC (-2.9), lex (-8.3)
PC Translator	ows (7.3), punct (5.3), missA (2.1)	disam (-2.7), extra (-7.7), lex (-7.9)

Table 6. Errors easy and hard to fix in blind post-editing.

suggests a rather weird lexical choice, no language model is applied to disambiguate better. This is confirmed in our table: such clear disambiguation flaws were easy to fix even without access to the source sentence because most post-editors speak English and could guess what the original word was.

The interesting difference between Google and CU-Bojar, both using phrase-based translation and n-gram language model, mentioned in Section 3.3 is more pronounced here. While errors in form in CU-Bojar's output are easy to fix (11.0), they are rather hard to fix in Google's output (-2.6). We attribute the difference to the strength of Google's language model: errors in form include errors in negation and the overall more or less fluent output can easily mislead post-editors. CU-Bojar uses a smaller language model and the errors in form probably cause output more incoherent than deceiving. Similarly, errors in form are not among the most serious problems in PC Translator output. While other systems confuse post-editors by missing content words (missC), PC Translator tends to confuse them by additional words (extra).

5. Conclusion

This paper attempted to reveal and quantify differences between error types various MT systems make when translating from English to Czech. The first dataset used consisted of the WMT09 blind post-edits. To complement this type of evaluation, we manually marked errors in the same set of system outputs.

Both types of manual evaluation can be used to reveal more about individual MT systems. While the reproducibility of each of the evaluations is relatively low (annotators diverge in errors they mark or post-edit), the overall picture provided by both evaluation types is rather similar: Statistical systems were somewhat better in lexical choice (probably thanks to the language model) while the fewest morphological errors can be achieved either by a large language model or a deterministic morphological generator. The drawback of a powerful language model is the risk of misleading: a fluent output is not a good translation of the source text.

We have suggested a method for detailed analysis of blind post-editing data. Given the availability of this manually created resource for various language pairs at WMT evaluation campaigns, we hope researchers will be able to focus on most serious errors of their specific MT systems.

Ondřej Bojar Analyzing Error Types in English-Czech Machine Translation (63–76)

Acknowledgement

The work on this project was supported by the grants P406/10/P259, P406/11/1499, and the project EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic).

We are grateful to all our student annotators and also to the anonymous reviewers for their comments on previous versions of the paper.

Bibliography

Bojar, Ondřej, David Mareček, Václav Novák, Martin Popel, Jan Ptáček, Jan Rouš, and Zdeněk Žabokrtský. English-Czech MT in 2008. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, March 2009. Association for Computational Linguistics.

Callison-Burch, Chris, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W08/W08-0309`.

Callison-Burch, Chris, Philipp Koehn, Christof Monz, and Josh Schroeder. Findings of the 2009 workshop on statistical machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, March 2009. Association for Computational Linguistics.

Hajič, Jan, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková Razímová. Prague Dependency Treebank 2.0. LDC2006T01, ISBN: 1-58563-370-4, 2006.

Hunt, James W. and M. Douglas McIlroy. An Algorithm for Differential File Comparison. Computing Science Technical Report 41, Bell Laboratories, June 1976.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P07/P07-2045`.

Leusch, Gregor and Hermann Ney. BLEUSP, INVWER, CDER: Three improved MT evaluation measures. In *NIST Metrics for Machine Translation Challenge*, Waikiki, Honolulu, Hawaii, Oct. 2008.

Sgall, Petr, Eva Hajičová, and Jarmila Panevová. *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects*. Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands, 1986.

Snover, Matthew, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. Fluency, adequacy, or HTER?: exploring different human judgments with a tunable MT metric.

In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 259–268, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://portal.acm.org/citation.cfm?id=1626431.1626480.

Vilar, David, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. Error Analysis of Machine Translation Output. In *International Conference on Language Resources and Evaluation*, pages 697–702, Genoa, Italy, May 2006.

Žabokrtský, Zdeněk, Jan Ptáček, and Petr Pajas. TectoMT: Highly Modular Hybrid MT System with Tectogrammatics Used as Transfer Layer. In *Proc. of the ACL Workshop on Statistical Machine Translation*, pages 167–170, Columbus, Ohio, USA, 2008.

Address for correspondence:
Ondřej Bojar
bojar@ufal.mff.cuni.cz
Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25
11800 Praha, Czech Republic

Scratching the Surface of Possible Translations

Ondřej Bojar, Matouš Macháček, Aleš Tamchyna, and Daniel Zeman

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
{bojar,machacek,tamchyna,zeman}@ufal.mff.cuni.cz

Abstract. One of the key obstacles in automatic evaluation of machine translation systems is the reliance on a few (typically just one) human-made reference translations to which the system output is compared. We propose a method of capturing millions of possible translations and implement a tool for translators to specify them using a compact representation. We evaluate this new type of reference set by edit distance and correlation to human judgements of translation quality.

Keywords: machine translation, evaluation, reference translations.

1 Introduction

The relationship between a sentence in a natural language as written down and its meaning is a very complex phenomenon. Many variations of the sentence preserve the meaning while other superficially very small changes can distort or completely reverse it. In order to process and produce sentences algorithmically, we need to somehow capture the semantic identity and similarity of sentences.

The issue has been extensively studied from a number of directions, starting with thesauri and other lexical databases that capture synonymy of individual words (most notably the WordNet [1], [2]), automatic paraphrasing of longer phrases or even sentences (e.g. [3], [4], [5]) or textual entailment [6]. We are still far away from a satisfactory solution.

The field of machine translation (MT) makes the issue tangible in a couple of ways, most importantly within the task of automatic MT evaluation. Current automatic MT evaluation techniques rely on the availability of some reference translation, i.e. the translation as produced by a human translator. Obtaining such reference translations is relatively costly, so most datasets provide only one reference translation, see e.g. [7].

Figure 1 illustrates the situation: while there are many possible translations of a given input sentence, only a handful of them are available as reference translations. The sets of hypotheses considered or finally selected by the MT system can be completely disjoint from the set of reference translations. Indeed, only about 5–10% of reference translations were *reachable* for Czech-to-English translation [8], and about a third of words in a system output are not confirmed by the reference despite not containing any errors based on manual evaluation [9]. Relying mostly on unreachable reference translations is detrimental for MT system development. Specifically, automatic MT evaluation methods perform worse and consequently automatic model optimization suffers.

I. Habernal and V. Matousek (Eds.): TSD 2013, LNAI 8082, pp. 465–474, 2013.

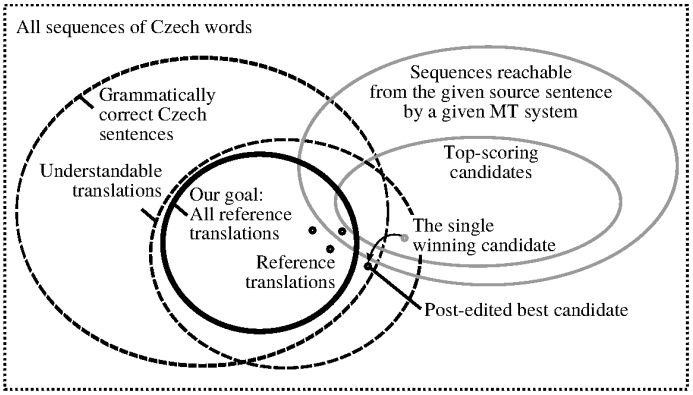


Fig. 1. The space of all considerable translations of a given source sentence. Human-produced sets are denoted using black lines, machine output is in grey.

We would like to bring the sets of acceptable and reachable translations closer to each other, providing more space for optimal hypothesis selection. This paper presents one possible step in that direction, namely significantly enlarging the set of reference translations. As outlined above, the dataset we created could serve well in research well beyond the MT field, e.g. in an analysis of *sentence-level* paraphrases.

In Section 2, we describe our annotation tool for producing large numbers of correct translations and relate it to a similar tool developed for English [10]. Section 3 provides basic statistics about the number of references we collected and Section 4 carefully analyzes and discusses their utility in MT evaluation.

2 Annotation Tools for Producing Many References

The most inspiring work for our experiment was that of Dreyer and Marcu [10]. Their annotators produce "translation networks", a compact representation of many references, to be used in their HyTER evaluation metric.

We experimented with their annotation interface developed primarily for English but found it rather cumbersome for Czech and other languages with richer morphology and a higher degree of word order freedom.

2.1 Recursive Transition Networks

The approach of [10] is based on recursive transition networks (RTN, [11]), a formalism with the power of context-free grammars. The main building block of the annotation tool in [10] is called a "card" and it covers multiple translations of a short phrase. By combining cards, a large network for the whole sentence can be built. Every path through the network represents a new sentence and the annotators construct the networks so that all such sentences are synonymous. See Figure 2 for an example.

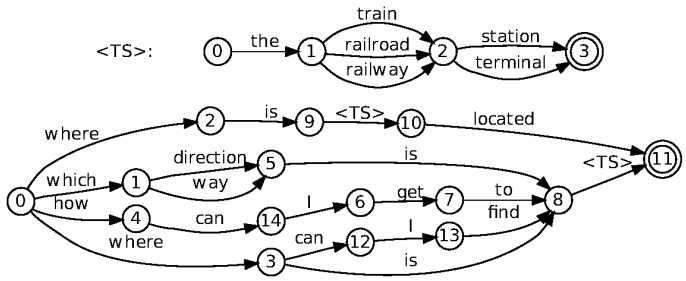


Fig. 2. An example of English recursive transition network from [10]

Reordering of phrases (not discussed in [10]) is possible within the RTN framework by changing the order of cards. The need of such a mechanism in English may seem negligible, yet there are situations such as direct speech where mutual positions of large blocks of text are perfectly interchangeable; similar patterns occur in Czech as well:

- He wanted to step down, he said, "so I could work with more freedom."
- He said: "I want to step down so I could work with more freedom."
- "I want to step down so I could work with more freedom," he said.

More importantly, it is difficult to specify *conditions* under which particular cards can combine in RTN. In morphologically rich languages, we often have to translate a phrase differently based on morpho-syntactic rules. For instance, functions of English verb arguments are determined using word order and prepositions. In Czech, they are determined using prepositions and morphological cases. Verbs subcategorize for noun phrases in particular cases. Consequently, if a verb is replaced by a synonym (or if the verb phrase is passivized or nominalized), the required case for the arguments of the verb may change. The case of the noun must then be reflected by its adjectival modifiers.

Let us illustrate this by an example. Consider the sentence *"The city council approved a new regulation."* The cards that would model pieces of this sentence in English could be combined more or less freely (within the fixed word order):[1]

the (city council / local government) (approved / gave blessing to / agreed with) a new (regulation / directive / decree)

If we ignored morphology, we could get a very similar picture with the Czech equivalents of the phrases:

(městská rada / zastupitelstvo města) (schválila / požehnala / souhlasila s) nový (předpis / směrnici / nařízení)

[1] We are aware that certain domains are much more sensitive to meaning distortions (directive vs. regulation) or inappropriate register (approve vs. give blessing). Our definition of meaning equivalence is not strict. We permit slight divergence if it can be reasonably expected that a human translator will pick either of the alternatives.

So far the alternatives within each part of the sentence differ *lexically*. However, the lexical selections have morphological implications and thus we also have to define *morphological* alternatives:

- One subject is feminine *(městská rada)*, the other is neuter *(zastupitelstvo města)*. Their gender must be reflected by the verbs *(schválila / schválilo // požehnala / požehnalo // souhlasila s / souhlasilo s)*.
- On a similar note, the three synonyms for *regulation* differ in gender, which dictates different suffixes for the adjective *new: nový předpis / novou směrnici / nové nařízení*.
- Each of the three verbs subcategorizes for a different case: *schválila* requires object in accusative, *požehnala* in dative and the preposition in *souhlasila s* requires instrumental. Thus we have *nový předpis / novému předpisu / novým předpisem // novou směrnici / nové směrnici / novou směrnicí // nové nařízení / novému nařízení / novým nařízením*.

2.2 Unification-Based Annotation

The RTN framework gives us a powerful tool to *combine* "cards". We would ideally want a tool that also lets us specify the *constraints* that must be fulfilled if two cards are to be combined.

We thus created our own compact representation for languages similar to Czech. Our main building block called *bubble*, comparable to the cards of [10], is defined by:

- the set (possibly discontinuous) of source language tokens it covers;
- the set of conditions it meets;
- the set of translation alternatives in the target language. Every alternative in the set covers the same set of source tokens and meets the same conditions.

A translation alternative is composed of *atoms* (tokens of the target language) and/or *slots* (positions in the translation alternative, specifying properties of other bubbles that are permitted to fill the slot). Where an RTN would refer to a smaller transition network (card) by its name, we refer to a smaller bubble by enumerating the constraints it must meet. For instance, we ask that the bubble covers the source word *regulation* (it may cover more words but this one must be among them) and that its form is in the accusative. Obviously we could achieve the same result in RTN by using more explicative card names, e.g. *regulation-acc*. Our approach is equivalently expressive but it increases annotators' comfort as well as maintainability of the whole system. While RTNs could be rewritten as a context-free grammar, our approach can be thought of as a unification grammar.

Typical creation of a translation network is analogous to traversing the dependency tree structure that models the syntax of the sentence. One starts at the verb, defines its translations and creates slots for its arguments (and adjuncts). Each argument typically receives its own bubble. The bubble can define alternations for a whole noun phrase, or it can again use slots to separate description of a modifier that could be reused elsewhere. Occasionally a bubble represents a subordinated clause and the process is applied recursively.

Unlike in common unification grammars, we are not forced to annotate a full syntactic tree. It is possible e.g. to create one flat bubble for the whole sentence. The only guiding principle in creating nested bubbles is economy: a set of alternations useful at two or more places is a candidate for a new bubble.

Along the same lines, the decomposition of the sentence into bubbles does not need to reflect linguistic constituents. Sometimes it is practical to take punctuation as the root, rather than the verb; high-level word order decisions drive the distribution of commas and quotation marks around clauses; co-ordination could be preferred over dependency etc. The set of possible constraints is not restricted in any way (e.g. to morphological categories and their values). The annotator is free to introduce arbitrary constraints, e.g. for ensuring good co-reference patterns, auxiliaries in coordination, rhematizers and negation interplay or even style and register features.

We developed two annotation environments in which translators create the compact representations. The Prolog programming language appears to be ideally suited for evaluation of constraints and expansion of bubbles. Several translators encode their annotations directly in Prolog. For those less technically capable we also designed a web-based graphical interface.

2.3 Prolog Interface

Roughly 300 lines of pre-programmed Prolog code provide the necessary set of predicates that check constraints (bubble-slot compatibility) and make sure that all tokens of the source sentence are covered. The translator essentially creates a set of clauses for the predicate `option()`, each of those encodes a bubble. Every `option` lists the source words covered, the conditions met and the target sequence consisting of atoms and slots. A slot refers at least to one source word that must be covered by the bubble in the slot; optionally, it also specifies additional conditions that must be met. Example:[2]

```
% option(+SrcWordsCovered, +ConditionsMet, +OutputAtomsAndSlots).
option([the, city, council], [], [městská, rada, [approved, fem]]).
  % "The city council" can be translated as "městská rada" and then
  % it requires the translation of "approved" in feminine gender.
option([approved], [fem], [schválila, [regulation, acc]]).
option([approved], [fem], [souhlasila, s, [regulation, ins]]).
  % Different translations of "approved" require "regulation" in
  % either "acc"usative or "ins"trumental cases.
option([a, new, regulation], [acc], [nový, předpis]).
option([a, new, regulation], [ins], [novým, předpisem]).
option([a, new, regulation], [acc], [novou, směrnici]).
option([a, new, regulation], [ins], [novou, směrnicí]).
```

A few additional constructs such as the logical `or()` and the possibility to simply drop a token further expand the tools the translators have at their disposal; note however that the syntax that the annotators have to grasp is extremely simple and virtually no knowledge of programming in general or Prolog in particular is required.

[2] A pre- and post-processor enables using uppercase letters and non-English letters freely in the `option()` predicates.

2.4 Web Interface

Some translators will be scared by any programming language, regardless how easy it may be. Others may face technical issues regarding installation and running a Prolog interpreter on their laptops. In order to accommodate all translators, we also developed a web-based graphical interface. It works as a wrapper for the Prolog engine: bubbles defined in the browser are sent to the server, converted to Prolog clauses and evaluated. The server then sends back either the full list of translations (which is only practical for shorter sentences) or the list of differences against the previous state.

The main problem of the web interface was that we did not anticipate that many bubbles and translations generated. The tool implemented is thus too heavy in terms of processing time, network load and even the required screen space.

3 Collected Data

We selected 50 sentences from the WMT11 test set [12] for our annotation. This particular test set was used in various experiments before and we can thus use:

- manual system rankings of the official WMT11 manual evaluation (see also [13] for a discussion of the rankings)
- the single official reference translation of WMT11 (denoted "O" in the following)
- three more reference translations that come from the German version of the test set [14] (denoted "G" in the following)
- two manual post-edits of a phrase-based MT system similar to those participating in the WMT11 competition (denoted "P"; this set contains only 1997 sentences, each post-edited by two independent annotators.)

Six translators were involved in the task, producing 77 sets of references altogether. Of the 50 sentences, 24 were translated by one annotator, 25 by two and one sentence has three annotations. Each annotator was instructed to spend at most 2 hours translating one sentence. More than 1 hour was needed for a typical sentence.

Utilizing the existing versions, one of the translators used German as the input language; the others used English. All of the annotators had access to one pre-existing human translation in English, German, Spanish and French, and up to four Czech translations, which they could use for inspiration.[3]

We combine networks produced by different translators using simple union of the sets of target sentences (*finite-state union* of [10]). This set is denoted "D" in the following.

Table 1 shows basic statistics of the annotations. In all cases, annotators who used the Prolog interface were more productive, creating over 255 thousand reference translations per sentence on average, compared to roughly 49 thousand references produced by the web users.

[3] The texts are news articles of mixed origin. One of the pre-existing "translations" was actually the original text.

Table 1. Basic statistics of our annotations

Annotator	Interface	# of Sents.	Avg. Sent. Length	Avg. Number of Refs.
A	Prolog	3	14.0	483k
B		5	22.5	246k
C		20	24.1	223k
D	Web	25	25.1	54k
E		19	23.0	26k
F		5	15.7	111k

Great when a film has several target groups , but a shame if they are mutually exclusive

Je výborné , když má film větší počet cílových skupin , jen je smutné , že se navzájem vylučují .
Není k zahození , pokud má film vícero cílových skupin , jen je smůla , když se navzájem vylučují .
Skvělé , že má film víc cílových skupin , je ale politováníhodné , pokud se navzájem neslučují .
Prima , pokud má film více cílových skupin , jen je smutné , pokud se vzájemně neslučují .

Je výhodné , když snímek má několik cílových skupin , smutné ale je , pokud jsou navzájem neslučitelné .
Skvělé je , když snímek má několik cílových skupin , škoda ale je , pokud se tyto skupiny vzájemně vylučují .
Dobré je , když má film více cílových skupin , je jen smolné , pokud jsou tyto navzájem neslučitelné .
Výhodou je , pokud se snímek zaměří na několik cílových skupin , je ale smolné , pakliže jsou vzájemně neslučitelné .

Fig. 3. Four random samples from 196k (Web) and 943k (Prolog) variations of one sentence

For the 25 sentences annotated by two translators, we measured the overlap between the sets of references. We created the union and intersection of the two sets (after tokenizing the sentences) and simply measured the ratio. The results provide further evidence of the richness of possible translations – on average, the overlap was only 4.4% of the produced references. With the exception of two very short sentences, the overlap was always below 10%. See Figure 3 for some examples.

4 Analysis

4.1 String Similarity

Our metric of similarity is based on the well-known Levenshtein distance [15] and returns a value between 0 (completely different) and 1 (identical strings):

$$\text{similarity}(x, y) = 1 - \frac{\text{levenshtein}(x, y)}{\max(\text{length}(x), \text{length}(y))}$$

In order to quantify the diversity of the produced manifold translations, we sampled pairs of translations of each sentence, looking for the pair with the smallest similarity. For four sentences, translations with similarity below 0.1 were found in the samples. The minimum similarity averaged over sentences in our dataset was 0.24 ± 0.13 (standard deviation). This result indicates how much the surface realizations differed while preserving identical meaning.

We also measured the string similarity between system outputs and various reference translations. Table 2 summarizes the results. Unsurprisingly, each of the two manually post-edited translations (P_1 and P_2) are much closer to the original system outputs

472 O. Bojar et al.

Table 2. String similarity between system outputs and reference translations

System	String Similarity to the Given Reference			
	D (closest)	O	P_1	P_2
online-B	0.65	0.55	0.66	0.65
cu-tamchyna	0.65	0.52	0.68	0.69
cu-bojar-contrastive	0.65	0.52	0.64	0.66
cu-bojar	0.65	0.51	0.65	0.68
uedin-contrastive	0.64	0.54	0.64	0.65
uedin	0.64	0.54	0.64	0.65
cu-tamchyna-contrastive	0.64	0.50	0.66	0.66
cu-marecek	0.64	0.52	0.67	0.68
jhu-contrastive	0.62	0.52	0.61	0.61
jhu	0.62	0.51	0.59	0.61
cu-zeman	0.61	0.52	0.61	0.62
cu-popel	0.60	0.51	0.59	0.61
commercial1	0.57	0.48	0.57	0.57
commercial2	0.56	0.46	0.57	0.57

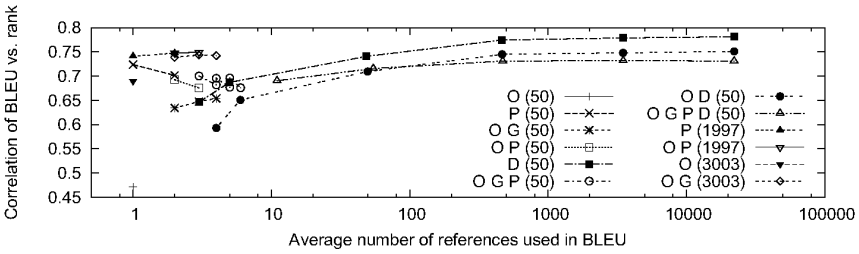


Fig. 4. Correlation of BLEU and manual system rankings with varying sets of references

than the official reference translation (O). However, our annotators managed to produce references (D) which are almost exactly as close as the post-edited translations, even though they did not have access to them or the system outputs (the table shows the similarity with the closest reference translation found).

4.2 Correlation with Manual Ranking

We evaluate the utility of the manifold reference translations by measuring the Pearson correlation between manual MT system evaluation and the common automatic MT evaluation method BLEU [16]. BLEU was originally tested with 4 reference translations and the number of reference translations is known to strongly influence its performance. In spite of that, BLEU is very often used with just a single reference translation, hoping that a larger test set (more sentences) will compensate the deficiency.

Figure 4 plots Pearson correlation of the official WMT11 system rankings and BLEU when varying the size of the test set (50, 1997 or 3003 sentences as noted in the legend)

and the average number of references per sentence. The sources O, P, and G provide us with 1, 2 or 3 references respectively. Our new dataset (denoted "D") consists of all the references generated from the web or Prolog annotation interface by any of the annotators. The number of references for each sentence differs. We shuffle them and take up to 5, 50, 500, ..., 50k items from the beginning.

The very baseline correlation is 0.47 ("O 50" in the chart), obtained with only the single official reference translation on the test set reduced to the 50 sentences where we have our extended references. Using the full test set ("O 3003"), the correlation jumps to 0.69. The 1, 2 or 3 additional references coming from German ("O G 3003") indicate how well the "standard" BLEU should fare: around 0.74. These four references (one official and three coming from German) on the small set of 50 sentences lead to quite a low result: 0.65.

A notable result is 0.72 ("P 50") obtained on the 50-sentences test set when we use the post-edited translations instead of the official translations. The official translations are obviously more distant from what the systems are capable of producing given the source. With distant references, large portions of output are not scored and systems may differ greatly in the translation quality of those unscored parts. It thus seems sensible to manually post-edit just 50 sentences coming from a baseline version of an MT system and evaluate modifications of the system on this small but tailor-made test set rather than on a larger less-matching set, perhaps even if it had 4 reference translations. The post-edits may however still suffer some problems: in our case, using not just one of the post-edit versions but both of them, the correlation drops to 0.70.

Lines in the chart extending beyond 10 references include our large reference sets. The limit on the number of references has to be taken with caution: subsampling 50 references from a 100k set constructed in 2 hours of annotation is bound to give better results than stopping the construction as soon as it generates 50 options. We nevertheless see that around 5k references, the correlation curves flatten. This could be attributed both to the inherent limitations of BLEU (evaluating the precision of up to 4-grams of words) as well as still the low coverage of the test set. Dreyer and Marcu mention that even billions of references obtained from two annotators still did not include many of the translations suggested by a third annotator.

The solid baseline 0.69 ("O 3003") is reached shortly after 5 random items from our annotation only (the second point on the line "D 50"), a much smaller set of sentences with many possible translations. Using up to 50k of the possible translations leads to the correlation of 0.78. However, we cannot claim that spending 2 hours $\times$ 2 annotators $\times$ 50 sentences, i.e. about 200 hours of work, is better than translating 3003 sentences (also about 200 hours of work), because our annotators *did* have access to several versions of each of the sentences.

5 Conclusion

We developed a method and two annotation environments for producing many (possibly all) acceptable translations of a sentence. We performed a small scale experiment in which human translators processed 50 sentences. Obviously this annotation process is very costly (1 to 2 hours per sentence) and we cannot expect anyone to annotate large

datasets this way. However, even the small sample provides a completely new point of view for the evaluation of machine translation output (tens or hundreds of thousands of references per average sentence). As expected, automatic evaluation against larger sets of references shows higher correlation with human judgment of translation quality.

A surprising observation is that just 50 post-edited translations serve as an equal or better reference than 3003 independent translations (correlation 0.70–0.72 vs. 0.69).

The annotated data we created is available to the research community. Besides machine translation, it can be also used to evaluate other NLP tasks, ranging from paraphrasing to grammar development or parsing.

Acknowledgements. This work was partially supported by the grants P406/11/1499 of the Grant Agency of the Czech Republic, FP7-ICT-2011-7-288487 (MosesCore) of the European Union and 1356213 of the Grant Agency of the Charles University. We are grateful to Markus Dreyer for a preview of their annotation interface.

References

1. Miller, G.A.: WordNet: A lexical database for English. Commun. ACM 38(11), 39–41 (1995)
2. Pala, K., Čapek, T., Zajíčková, B., et al.: Český WordNet 1.9 PDT (2010)
3. Bannard, C., Callison-Burch, C.: Paraphrasing with bilingual parallel corpora. In: Proc. of ACL, Ann Arbor, Michigan, USA, pp. 597–604 (2005)
4. Kauchak, D., Barzilay, R.: Paraphrasing for Automatic Evaluation. In: Proc. of NAACL/HLT, New York City, USA, pp. 455–462 (2006)
5. Denkowski, M., Lavie, A.: Meteor-next and the meteor paraphrase tables: Improved evaluation support for five target languages. In: Proc. of WMT and MetricsMATR, pp. 339–342. ACL, Uppsala (2010)
6. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. Journal of Artificial Intelligence Research 38, 135–187 (2010)
7. Callison-Burch, C., Koehn, P., Monz, C., et al.: Findings of the 2012 Workshop on Statistical Machine Translation. In: Proc. of WMT, pp. 22–64. ACL, Montréal (2012)
8. Bojar, O., Kos, K.: 2010 Failures in English-Czech Phrase-Based MT. In: Proc. WMT and MetricsMATR, pp. 60–66. ACL, Uppsala (2010)
9. Bojar, O., Kos, K., Mareček, D.: Tackling Sparse Data Issue in Machine Translation Evaluation. In: Proc. of ACL Short Papers, pp. 86–91. ACL, Uppsala (2010)
10. Dreyer, M., Marcu, D.: HyTER: Meaning-Equivalent Semantics for Translation Evaluation. In: Proc. of NAACL/HLT, Montréal, Canada, pp. 162–171 (2012)
11. Woods, W.A.: Transition network grammars for natural language analysis. Commun. ACM 13(10), 591–606 (1970), doi:10.1145/355598.362773
12. Callison-Burch, C., Koehn, P., Monz, C., Zaidan, O.: Findings of the 2011 Workshop on Statistical Machine Translation. In: Proc. of WMT, pp. 22–64. ACL (2011)
13. Bojar, O., Ercegovčević, M., Popel, M., Zaidan, O.: A Grain of Salt for the WMT Manual Evaluation. In: Proc. of WMT, pp. 1–11. ACL, Edinburgh (2011)
14. Bojar, O., Zeman, D., Dušek, O.: Additional German-Czech reference translations of the WMT'11 test set
15. Levenshtein, V.: Binary codes capable of correcting deletions, insertions and reversals. In: Soviet Physics-Doklandy, vol. 10 (1966)
16. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A Method for Automatic Evaluation of Machine Translation. In: Proc. of ACL, Philadelphia, Pennsylvania, pp. 311–318 (2002)

Tackling Sparse Data Issue in Machine Translation Evaluation *

Ondřej Bojar, Kamil Kos, and David Mareček
Charles University in Prague, Institute of Formal and Applied Linguistics
{bojar,marecek}@ufal.mff.cuni.cz, kamilkos@email.cz

Abstract

We illustrate and explain problems of n-grams-based machine translation (MT) metrics (e.g. BLEU) when applied to morphologically rich languages such as Czech. A novel metric SemPOS based on the deep-syntactic representation of the sentence tackles the issue and retains the performance for translation to English as well.

1 Introduction

Automatic metrics of machine translation (MT) quality are vital for research progress at a fast pace. Many automatic metrics of MT quality have been proposed and evaluated in terms of correlation with human judgments while various techniques of manual judging are being examined as well, see e.g. MetricsMATR08 (Przybocki et al., 2008)[1], WMT08 and WMT09 (Callison-Burch et al., 2008; Callison-Burch et al., 2009)[2].

The contribution of this paper is twofold. Section 2 illustrates and explains severe problems of a widely used BLEU metric (Papineni et al., 2002) when applied to Czech as a representative of languages with rich morphology. We see this as an instance of the sparse data problem well known for MT itself: too much detail in the formal representation leading to low coverage of e.g. a translation dictionary. In MT evaluation, too much detail leads to the lack of comparable parts of the hypothesis and the reference.

* This work has been supported by the grants EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic), FP7-ICT-2009-4-247762 (Faust), GA201/09/H057, GAUK 1163/2010, and MSM 0021620838. We are grateful to the anonymous reviewers for further research suggestions.

[1] http://nist.gov/speech/tests/metricsmatr/2008/results/

[2] http://www.statmt.org/wmt08 and wmt09

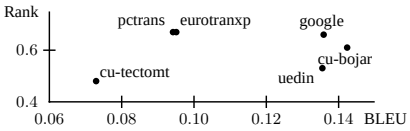


Figure 1: BLEU and human ranks of systems participating in the English-to-Czech WMT09 shared task.

Section 3 introduces and evaluates some new variations of SemPOS (Kos and Bojar, 2009), a metric based on the deep syntactic representation of the sentence performing very well for Czech as the target language. Aside from including dependency and n-gram relations in the scoring, we also apply and evaluate SemPOS for English.

2 Problems of BLEU

BLEU (Papineni et al., 2002) is an established language-independent MT metric. Its correlation to human judgments was originally deemed high (for English) but better correlating metrics (esp. for other languages) were found later, usually employing language-specific tools, see e.g. Przybocki et al. (2008) or Callison-Burch et al. (2009). The unbeaten advantage of BLEU is its simplicity.

Figure 1 illustrates a very low correlation to human judgments when translating to Czech. We plot the official BLEU score against the rank established as the percentage of sentences where a system ranked no worse than all its competitors (Callison-Burch et al., 2009). The systems developed at Charles University (cu-) are described in Bojar et al. (2009), uedin is a vanilla configuration of Moses (Koehn et al., 2007) and the remaining ones are commercial MT systems.

In a manual analysis, we identified the reasons for the low correlation: BLEU is overly sensitive to *sequences* and *forms* in the hypothesis matching

Proceedings of the ACL 2010 Conference Short Papers, pages 86–91,
Uppsala, Sweden, 11-16 July 2010.

Con-firmed	Error Flags	1-grams	2-grams	3-grams	4-grams
Yes	Yes	6.34%	1.58%	0.55%	0.29%
Yes	No	36.93%	13.68%	5.87%	2.69%
No	Yes	22.33%	41.83%	54.64%	63.88%
No	No	**34.40%**	**42.91%**	**38.94%**	**33.14%**
Total n-grams		35,531	33,891	32,251	30,611

Table 1: n-grams confirmed by the reference and containing error flags.

the reference translation. This focus goes directly against the properties of Czech: relatively free word order allows many permutations of words and rich morphology renders many valid word forms not confirmed by the reference.[3] These problems are to some extent mitigated if several reference translations are available, but this is often not the case.

Figure 2 illustrates the problem of "sparse data" in the reference. Due to the lexical and morphological variance of Czech, only a single word in each hypothesis matches a word in the reference. In the case of pctrans, the match is even a false positive, "do" (to) is a preposition that should be used for the "minus" phrase and not for the "end of the day" phrase. In terms of BLEU, both hypotheses are equally poor but 90% of their tokens were not evaluated.

Table 1 estimates the overall magnitude of this issue: For 1-grams to 4-grams in 1640 instances (different MT outputs and different annotators) of 200 sentences with manually flagged errors[4], we count how often the n-gram is confirmed by the reference and how often it contains an error flag. The suspicious cases are n-grams confirmed by the reference but still containing a flag (false positives) and n-grams not confirmed despite containing no error flag (false negatives).

Fortunately, there are relatively few false positives in n-gram based metrics: 6.3% of unigrams and far fewer higher n-grams.

The issue of false negatives is more serious and confirms the problem of sparse data if only one reference is available. 30 to 40% of n-grams do not contain any error and yet they are not confirmed by the reference. This amounts to 34% of running unigrams, giving enough space to differ in human judgments and still remain unscored.

Figure 3 documents the issue across languages: the lower the BLEU score itself (i.e. fewer confirmed n-grams), the lower the correlation to human judgments regardless of the target language (WMT09 shared task, 2025 sentences per language).

Figure 4 illustrates the overestimation of scores caused by too much attention to sequences of tokens. A phrase-based system like Moses (cubojar) can sometimes produce a long sequence of tokens exactly as required by the reference, leading to a high BLEU score. The framed words in the illustration are not confirmed by the reference, but the actual error in these words is very severe for comprehension: nouns were used twice instead of finite verbs, and a misleading translation of a preposition was chosen. The output by pctrans preserves the meaning much better despite not scoring in either of the finite verbs and producing far shorter confirmed sequences.

3 Extensions of SemPOS

SemPOS (Kos and Bojar, 2009) is inspired by metrics based on overlapping of linguistic features in the reference and in the translation (Giménez and Márquez, 2007). It operates on so-called "tectogrammatical" (deep syntactic) representation of the sentence (Sgall et al., 1986; Hajič et al., 2006), formally a dependency tree that includes only autosemantic (content-bearing) words.[5] SemPOS as defined in Kos and Bojar (2009) disregards the syntactic structure and uses the semantic part of speech of the words (noun, verb, etc.). There are 19 fine-grained parts of speech. For each semantic part of speech t, the overlapping $\mathrm{O}(t)$ is set to zero if the part of speech does not occur in the reference or the candidate set and otherwise it is computed as given in Equation 1 below.

[3] Condon et al. (2009) identify similar issues when evaluating translation to Arabic and employ rule-based normalization of MT output to improve the correlation. It is beyond the scope of this paper to describe the rather different nature of morphological richness in Czech, Arabic and also other languages, e.g. German or Finnish.

[4] The dataset with manually flagged errors is available at `http://ufal.mff.cuni.cz/euromatrixplus/`

[5] We use TectoMT (Žabokrtský and Bojar, 2008), `http://ufal.mff.cuni.cz/tectomt/`, for the linguistic pre-processing. While both our implementation of SemPOS as well as TectoMT are in principle freely available, a stable public version has yet to be released. Our plans include experiments with approximating the deep syntactic analysis with a simple tagger, which would also decrease the installation burden and computation costs, at the expense of accuracy.

SRC	Prague Stock Market falls to minus by the end of the trading day
REF	pražská burza se ke konci obchodování propadla do minusu
cu-bojar	praha stock market klesne k minus na konci obchodního dne
pctrans	praha trh cenných papírů padá minus do konce obchodního dne

Figure 2: Sparse data in BLEU evaluation: Large chunks of hypotheses are not compared at all. Only a single unigram in each hypothesis is confirmed in the reference.

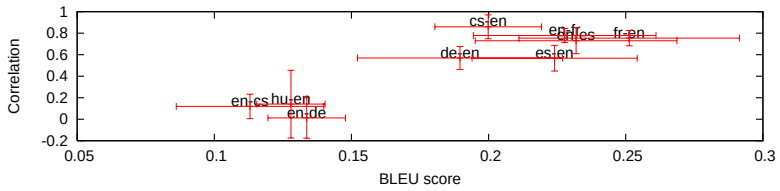


Figure 3: BLEU correlates with its correlation to human judgments. BLEU scores around 0.1 predict little about translation quality.

$$\mathrm{O}(t) = \frac{\sum_{i \in I} \sum_{w \in r_i \cap c_i} \min(\mathrm{cnt}(w,t,r_i), \mathrm{cnt}(w,t,c_i))}{\sum_{i \in I} \sum_{w \in r_i \cup c_i} \max(\mathrm{cnt}(w,t,r_i), \mathrm{cnt}(w,t,c_i))} \quad (1)$$

The semantic part of speech is denoted t; c_i and r_i are the candidate and reference translations of sentence i, and $\mathrm{cnt}(w,t,rc)$ is the number of words w with type t in rc (the reference or the candidate). The matching is performed on the level of lemmas, i.e. no morphological information is preserved in ws. See Figure 5 for an example; the sentence is the same as in Figure 4.

The final SemPOS score is obtained by macro-averaging over all parts of speech:

$$\mathrm{SemPOS} = \frac{1}{|T|} \sum_{t \in T} \mathrm{O}(t) \quad (2)$$

where T is the set of all possible semantic parts of speech types. (The degenerate case of blank candidate and reference has SemPOS zero.)

3.1 Variations of SemPOS

This section describes our modifications of SemPOS. All methods are evaluated in Section 3.2.

Different Classification of Autosemantic Words. SemPOS uses semantic parts of speech to classify autosemantic words. The tectogrammatical layer offers also a feature called *Functor* describing the relation of a word to its governor similarly as semantic roles do. There are 67 functor types in total.

Using *Functor* instead of *SemPOS* increases the number of word classes that independently require a high overlap. For a contrast we also completely remove the classification and use only one global class (*Void*).

Deep Syntactic Relations in SemPOS. In SemPOS, an autosemantic word of a class is confirmed if its lemma matches the reference. We utilize the dependency relations at the tectogrammatical layer to validate valence by refining the overlap and requiring also the lemma of 1) the parent (denoted "par"), or 2) all the children regardless of their order (denoted "sons") to match.

Combining BLEU and SemPOS. One of the major drawbacks of *SemPOS* is that it completely ignores word order. This is too coarse even for languages with relatively free word order like Czech. Another issue is that it operates on lemmas and it completely disregards correct word forms. Thus, a weighted linear combination of SemPOS and BLEU (computed on the surface representation of the sentence) should compensate for this. For the purposes of the combination, we compute BLEU *only* on unigrams up to fourgrams (denoted BLEU_1, ..., BLEU_4) but including the brevity penalty as usual. Here we try only a few weight settings in the linear combination but given a held-out dataset, one could optimize the weights for the best performance.

SRC	Congress yields: US government can pump 700 billion dollars into banks
REF	kongres ustoupil : vláda usa může do bank napumpovat 700 miliard dolarů
cu-bojar	kongres | výnosy | : vláda usa může | čerpadlo | 700 miliard dolarů | v | bankách
pctrans	kongres vynáší : us vláda může čerpat 700 miliardu dolarů do bank

Figure 4: Too much focus on sequences in BLEU: pctrans' output is better but does not score well. BLEU gave credit to cu-bojar for 1, 3, 5 and 8 fourgrams, trigrams, bigrams and unigrams, resp., but only for 0, 0, 1 and 8 n-grams produced by pctrans. Confirmed sequences of tokens are underlined and important errors (not considered by BLEU) are framed.

REF	kongres/n ustoupit/v :/n vláda/n usa/n banka/n napumpovat/v 700/n miliarda/n dolar/n
cu-bojar	kongres/n výnos/n :/n vláda/n usa/n moci/v čerpadlo/n 700/n miliarda/n dolar/n banka/n
pctrans	kongres/n vynášet/v :/n us/n vláda/n čerpat/v 700/n miliarda/n dolar/n banka/n

Figure 5: SemPOS evaluates the overlap of lemmas of autosemantic words given their semantic part of speech (n, v, ...). Underlined words are confirmed by the reference.

SemPOS for English. The tectogrammatical layer is being adapted for English (Cinková et al., 2004; Hajič et al., 2009) and we are able to use the available tools to obtain all SemPOS features for English sentences as well.

3.2 Evaluation of SemPOS and Friends

We measured the metric performance on data used in MetricsMATR08, WMT09 and WMT08. For the evaluation of metric correlation with human judgments at the system level, we used the Pearson correlation coefficient ρ applied to ranks. In case of a tie, the systems were assigned the average position. For example if three systems achieved the same highest score (thus occupying the positions 1, 2 and 3 when sorted by score), each of them would obtain the average rank of $2 = \frac{1+2+3}{3}$. When correlating ranks (instead of exact scores) and with this handling of ties, the Pearson coefficient is equivalent to Spearman's rank correlation coefficient.

The MetricsMATR08 human judgments include preferences for pairs of MT systems saying which one of the two systems is better, while the WMT08 and WMT09 data contain system scores (for up to 5 systems) on the scale 1 to 5 for a given sentence. We assigned a human ranking to the systems based on the percent of time that their translations were judged to be better than or equal to the translations of any other system in the manual evaluation. We converted automatic metric scores to ranks.

Metrics' performance for translation to English and Czech was measured on the following testsets (the number of human judgments for a given source language in brackets):

To English: MetricsMATR08 (cn+ar: 1652), WMT08 News Articles (de: 199, fr: 251), WMT08 Europarl (es: 190, fr: 183), WMT09 (cz: 320, de: 749, es: 484, fr: 786, hu: 287)

To Czech: WMT08 News Articles (en: 267), WMT08 Commentary (en: 243), WMT09 (en: 1425)

The MetricsMATR08 testset contained 4 reference translations for each sentence whereas the remaining testsets only one reference.

Correlation coefficients for English are shown in Table 2. The best metric is Void_{par} closely followed by $\text{Void}_{\text{sons}}$. The explanation is that Void compared to SemPOS or Functor does not lose points by an erroneous assignment of the POS or the functor, and that Void_{par} profits from checking the dependency relations between autosemantic words. The combination of BLEU and SemPOS[6] outperforms both individual metrics, but in case of SemPOS only by a minimal difference. Additionally, we confirm that 4-grams alone have little discriminative power both when used as a metric of their own (BLEU_4) as well as in a linear combination with SemPOS.

The best metric for Czech (see Table 3) is a linear combination of SemPOS and 4-gram BLEU closely followed by other SemPOS and BLEU_n combinations. We assume this is because BLEU_4 can capture correctly translated fixed phrases, which is positively reflected in human judgments. Including BLEU_1 in the combination favors translations with word forms as expected by the refer-

[6] For each $n \in \{1, 2, 3, 4\}$, we show only the best weight setting for SemPOS and BLEU_n.

Metric	Avg	Best	Worst
Void_{par}	0.75	0.89	0.60
$\text{Void}_{\text{sons}}$	0.75	0.90	0.54
Void	0.72	0.91	0.59
$\text{Functor}_{\text{sons}}$	0.72	1.00	0.43
GTM	0.71	0.90	0.54
$4{\cdot}\text{SemPOS}{+}1{\cdot}\text{BLEU}_2$	0.70	0.93	0.43
$\text{SemPOS}_{\text{par}}$	0.70	0.93	0.30
$1{\cdot}\text{SemPOS}{+}4{\cdot}\text{BLEU}_3$	0.70	0.91	0.26
$4{\cdot}\text{SemPOS}{+}1{\cdot}\text{BLEU}_1$	0.69	0.93	0.43
NIST	0.69	0.90	0.53
$\text{SemPOS}_{\text{sons}}$	0.69	0.94	0.40
SemPOS	0.69	0.95	0.30
$2{\cdot}\text{SemPOS}{+}1{\cdot}\text{BLEU}_4$	0.68	0.91	0.09
BLEU_1	0.68	0.87	0.43
BLEU_2	0.68	0.90	0.26
BLEU_3	0.66	0.90	0.14
BLEU	0.66	0.91	0.20
TER	0.63	0.87	0.29
PER	0.63	0.88	0.32
BLEU_4	0.61	0.90	-0.31
$\text{Functor}_{\text{par}}$	0.57	0.83	-0.03
Functor	0.55	0.82	-0.09

Table 2: Average, best and worst system-level correlation coefficients for translation to English from various source languages evaluated on 10 different testsets.

Metric	Avg	Best	Worst
$3{\cdot}\text{SemPOS}{+}1{\cdot}\text{BLEU}_4$	0.55	0.83	0.14
$2{\cdot}\text{SemPOS}{+}1{\cdot}\text{BLEU}_2$	0.55	0.83	0.14
$2{\cdot}\text{SemPOS}{+}1{\cdot}\text{BLEU}_1$	0.53	0.83	0.09
$4{\cdot}\text{SemPOS}{+}1{\cdot}\text{BLEU}_3$	0.53	0.83	0.09
SemPOS	0.53	0.83	0.09
BLEU_2	0.43	0.83	0.09
$\text{SemPOS}_{\text{par}}$	0.37	0.53	0.14
$\text{Functor}_{\text{sons}}$	0.36	0.53	0.14
GTM	0.35	0.53	0.14
BLEU_4	0.33	0.53	0.09
Void	0.33	0.53	0.09
NIST	0.33	0.53	0.09
$\text{Void}_{\text{sons}}$	0.33	0.53	0.09
BLEU	0.33	0.53	0.09
BLEU_3	0.33	0.53	0.09
BLEU_1	0.29	0.53	-0.03
$\text{SemPOS}_{\text{sons}}$	0.28	0.42	0.03
$\text{Functor}_{\text{par}}$	0.23	0.40	0.14
Functor	0.21	0.40	0.09
Void_{par}	0.16	0.53	-0.08
PER	0.12	0.53	-0.09
TER	0.07	0.53	-0.23

Table 3: System-level correlation coefficients for English-to-Czech translation evaluated on 3 different testsets.

ence, thus allowing to spot bad word forms. In all cases, the linear combination puts more weight on SemPOS. Given the negligible difference between SemPOS alone and the linear combinations, we see that word forms are not the major issue for humans interpreting the translation—most likely because the systems so far often make more important errors. This is also confirmed by the observation that using BLEU alone is rather unreliable for Czech and BLEU-1 (which judges unigrams only) is even worse. Surprisingly BLEU-2 performed better than any other n-grams for reasons that have yet to be examined. The error metrics PER and TER showed the lowest correlation with human judgments for translation to Czech.

4 Conclusion

This paper documented problems of single-reference BLEU when applied to morphologically rich languages such as Czech. BLEU suffers from a sparse data problem, unable to judge the quality of tokens not confirmed by the reference. This is confirmed for other languages as well: the lower the BLEU score the lower the correlation to human judgments.

We introduced a refinement of SemPOS, an automatic metric of MT quality based on deep-syntactic representation of the sentence tackling the sparse data issue. SemPOS was evaluated on translation to Czech and to English, scoring better than or comparable to many established metrics.

References

Ondřej Bojar, David Mareček, Václav Novák, Martin Popel, Jan Ptáček, Jan Rouš, and Zdeněk Žabokrtský. 2009. English-Czech MT in 2008. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, March. Association for Computational Linguistics.

Chris Callison-Burch, Cameron Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio, June. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 workshop on statistical machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece. Association for Computational Linguistics.

Silvie Cinková, Jan Hajič, Marie Mikulová, Lucie Mladová, Anja Nedolužko, Petr Pajas, Jarmila Panevová, Jiří Semecký, Jana Šindlerová, Josef Toman, Zdeňka Urešová, and Zdeněk Žabokrtský. 2004. Annotation of English on the tectogrammatical level. Technical Report TR-2006-35, ÚFAL/CKL, Prague, Czech Republic, December.

Sherri Condon, Gregory A. Sanders, Dan Parvaz, Alan Rubenstein, Christy Doran, John Aberdeen, and Beatrice Oshika. 2009. Normalization for Automated Metrics: English and Arabic Speech Translation. In *MT Summit XII*.

Jesús Giménez and Lluís Márquez. 2007. Linguistic Features for Automatic Evaluation of Heterogenous MT Systems. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 256–264, Prague, June. Association for Computational Linguistics.

Jan Hajič, Silvie Cinková, Kristýna Čermáková, Lucie Mladová, Anja Nedolužko, Petr Pajas, Jiří Semecký, Jana Šindlerová, Josef Toman, Kristýna Tomšů, Matěj Korvas, Magdaléna Rysová, Kateřina Veselovská, and Zdeněk Žabokrtský. 2009. Prague English Dependency Treebank 1.0. Institute of Formal and Applied Linguistics, Charles University in Prague, ISBN 978-80-904175-0-2, January.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, Zdeněk Žabokrtský, and Magda Ševčíková Razímová. 2006. Prague Dependency Treebank 2.0. LDC2006T01, ISBN: 1-58563-370-4.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Kamil Kos and Ondřej Bojar. 2009. Evaluation of Machine Translation Metrics for Czech as the Target Language. *Prague Bulletin of Mathematical Linguistics*, 92.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *ACL 2002, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania.

M. Przybocki, K. Peterson, and S. Bronsart. 2008. Official results of the NIST 2008 "Metrics for MAchine TRanslation" Challenge (MetricsMATR08).

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence and Its Semantic and Pragmatic Aspects*. Academia/Reidel Publishing Company, Prague, Czech Republic/Dordrecht, Netherlands.

Zdeněk Žabokrtský and Ondřej Bojar. 2008. TectoMT, Developer's Guide. Technical Report TR-2008-39, Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague, December.

A Grain of Salt for the WMT Manual Evaluation*

Ondřej Bojar, Miloš Ercegovčević, Martin Popel
Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
{bojar,popel}@ufal.mff.cuni.cz
ercegovcevic@hotmail.com

Omar F. Zaidan
Department of Computer Science
Johns Hopkins University
ozaidan@cs.jhu.edu

Abstract

The Workshop on Statistical Machine Translation (WMT) has become one of ACL's flagship workshops, held annually since 2006. In addition to soliciting papers from the research community, WMT also features a shared translation task for evaluating MT systems. This shared task is notable for having *manual evaluation* as its cornerstone. The Workshop's overview paper, playing a descriptive and administrative role, reports the main results of the evaluation without delving deep into analyzing those results. The aim of this paper is to investigate and explain some interesting idiosyncrasies in the reported results, which only become apparent when performing a more thorough analysis of the collected annotations. Our analysis sheds some light on how the reported results should (and should not) be interpreted, and also gives rise to some helpful recommendation for the organizers of WMT.

1 Introduction

The Workshop on Statistical Machine Translation (WMT) has become an annual feast for MT researchers. Of particular interest is WMT's shared translation task, featuring a component for manual evaluation of MT systems. The friendly competition is a source of inspiration for participating teams, and the yearly overview paper (Callison-Burch et al., 2010) provides a concise report of the state of the art. However, the amount of interesting data collected every year (the system outputs and, most importantly, the annotator judgments) is quite large, exceeding what the WMT overview paper can afford to analyze with much depth.

In this paper, we take a closer look at the data collected in last year's workshop, WMT10[1], and delve a bit deeper into analyzing the manual judgments. We focus mainly on the English-to-Czech task, as it included a diverse portfolio of MT systems, was a heavily judged language pair, and also illustrates interesting "contradictions" in the results. We try to explain such points of interest, and analyze what we believe to be the positive and negative aspects of the currently established evaluation procedure of WMT.

Section 2 examines the primary style of manual evaluation: system ranking. We discuss how the interpretation of collected judgments, the computation of annotator agreement, and document that annotators' individual preferences may render two systems effectively incomparable. Section 3 is devoted to the impact of embedding reference translations, while Section 4 and Section 5 discuss some idiosyncrasies of other WMT shared tasks and manual evaluation in general.

2 The *System Ranking* Task

At the core of the WMT manual evaluation is the system ranking task. In this task, the annotator is presented with a source sentence, a reference translation, and the outputs of five systems over that source sentence. The instructions are kept minimal: the annotator is to rank the presented translations from best to worst. Ties are allowed, but the scale provides five rank labels, allowing the annotator to give a total order if desired.

The five assigned rank labels are submitted at once, making the 5-tuple a unit of annotation. In the following, we will call this unit a *block*. The blocks differ from each other in the choice of the

* This work has been supported by the grants EuroMatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic), P406/10/P259, MSM 0021620838, and DARPA GALE program under Contract No. HR0011-06-2-0001. We are grateful to our students, colleagues, and the three reviewers for various observations and suggestions.

[1] http://www.statmt.org/wmt10

Proceedings of the 6th Workshop on Statistical Machine Translation, pages 1–11,
Edinburgh, Scotland, UK, July 30–31, 2011.

Language Pair	Systems	Blocks	Labels	Comparisons	Ref $\geq$ others	Intra-annot. κ	Inter-annot. κ
German-English	26	1,050	5,231	10,424	0.965	0.607	0.492
English-German	19	1,407	6,866	13,694	0.976	0.560	0.512
Spanish-English	15	1,140	5,665	11,307	0.989	0.693	0.508
English-Spanish	17	519	2,591	5,174	0.935	0.696	0.594
French-English	25	837	4,156	8,294	0.981	0.722	0.452
English-French	20	801	3,993	7,962	0.917	0.636	0.449
Czech-English	13	543	2,691	5,375	0.976	0.700	0.504
English-Czech	18	1,395	6,803	13,538	0.959	0.620	0.444
Average	19	962	4,750	9,471	0.962	0.654	0.494

Table 1: Statistics on the collected rankings, quality of references and kappas across language pairs. In general, a block yields a set of five rank labels, which yields a set of $\binom{5}{2} = 10$ pairwise comparisons. Due to occasional omitted labels, the Comparisons/Blocks ratio is not exactly 10.

source sentence and the choice of the five systems being compared. A couple of tricks are introduced in the sampling of the source sentences, to ensure that a large enough number of judgments is repeated across different screens for meaningful computation of inter- and intra-annotator agreement. As for the sampling of systems, it is done uniformly – no effort is made to oversample or undersample a particular system (or a particular pair of systems together) at any point in time.

In terms of the interface, the evaluation utilizes the infrastructure of Amazon's Mechanical Turk (MTurk)[2], with each MTurk HIT[3] containing three blocks, corresponding to three consecutive source sentences.

Table 1 provides a brief comparison of the various language pairs in terms of number of MT systems compared (including the reference), number of blocks ranked, the number of pairwise comparisons extracted from the rankings (one block with 5 systems ranked gives 10 pairwise comparisons, but occasional unranked systems are excluded), the quality of the reference (the percentage of comparisons where the reference was better or equal than another system), and the κ statistic, which is a measure of agreement (see Section 2.2 for more details).[4]

We see that English-to-Czech, the language pair on which we focus, is not far from the average in all those characteristics except for the number of collected comparisons (and blocks), making it the second most evaluated language pair.

[2] `http://www.mturk.com/`

[3] "HIT" is an acronym for *human intelligence task*, which is the MTurk term for a single screen presented to the annotator.

[4] We only use the "expert" annotations of WMT10, ignoring the data collected from paid annotators on MTurk, since they were not part of the official evaluation.

2.1 Interpreting the Rank Labels

The description in the WMT overview paper says: "Relative ranking is our official evaluation metric. [Systems] are ranked based on how frequently they were judged to be **better than or equal to any other system**." (Emphasis added.) The WMT overview paper refers to this measure as "$\geq$ others", with a variant of it called "$>$ others" that does not reward ties.

We first note that this description is somewhat ambiguous, and an uninformed reader might interpret it in one of two different ways. For some system A, each block in which A appears includes four implicit pairwise comparisons (against the other presented systems). How is A's score computed from those comparisons?

The correct interpretation is that A is rewarded once for **each** of the four comparisons in which A wins (or ties).[5] In other words, A's score is the number of pairwise comparisons in which A wins (or ties), divided by the total number of pairwise comparisons involving A. We will use "$\geq$ others" (resp. "$>$ others") to refer to this interpretation, in keeping with the terminology of the overview paper.

The other interpretation is that A is rewarded only if A wins (or ties) **all** four comparisons. In other words, A's score is the number of *blocks* in which A wins (or ties) all comparisons, divided by the number of *blocks* in which A appears. We will use "$\geq$ all in block" (resp. "$>$ all in block") to refer to this interpretation.[6]

[5] Personal communication with WMT organizers.

[6] There is yet a third interpretation, due to a literal reading of the description, where A is rewarded at most once per block if it wins (or ties) *any one* of its four comparisons. This is probably less useful: it might be good at identifying the bottom tier of systems, but would fail to distinguish between all other systems.

	REF	CU-BOJAR	CU-TECTO	EUROTRANS	ONLINEB	PC-TRANS	UEDIN
$\geq$ others	95.9	65.6	60.1	54.0	**70.4**	62.1	62.2
$>$ others	90.5	45.0	44.1	39.3	49.1	**49.4**	39.6
$\geq$ all in block	93.1	32.3	30.7	23.4	**37.5**	32.5	28.1
$>$ all in block	81.3	13.6	**19.0**	13.3	15.6	18.7	10.6

Table 2: Sentence-level ranking scores for the WMT10 English-Czech language pair. The "$\geq$ others" and "$>$ others" scores reproduced here exactly match numbers published in the WMT10 overview paper. A boldfaced score marks the best system in a given row (besides the reference).

For quality control purposes, the WMT organizers embed the reference translations as a 'system' alongside the actual entries (the idea being that an annotator clicking randomly would be easy to detect, since they would not consistently rank the reference 'system' highly). This means that the reference is as likely as any other system to appear in a block, and when the score for a system A is computed, pairwise comparisons with the reference *are* included.

We use the publicly released human judgments[7] to compute the scores of systems participating in the English-Czech subtask, under both interpretations. Table 2 reports the scores, with our "$\geq$ others" (resp. "$>$ others") scores reproduced exactly matching those reported in Table 21 of the WMT overview paper. (For clarity, Table 2 is abbreviated to include only the top six systems of twelve.)

Our first suggestion is that **both** measures could be reported in future evaluations, since each tells us something different. The first interpretation gives partial credit for an MT system, hence distinguishing systems from each other at a finer level. This is especially important for a language pair with relatively few annotations, since "$\geq$ others" would produce a larger number of data points (four per system per block) than "$\geq$ all in block" (one per system per block). Another advantage of the official "$\geq$ others" is greater robustness towards various factors like the number of systems in the competition, the number of systems in one block or the presence of the reference in the block (however, see Section 3).

As for the second interpretation, it helps identify whether or not a single system (or a small group of systems) is strongly dominant over the other systems. For the systems listed in Table 2,

[7] http://statmt.org/wmt10/results.html

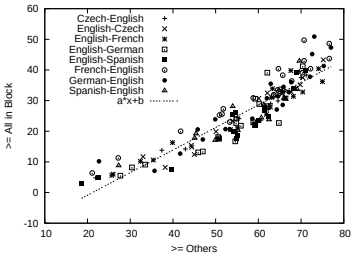


Figure 1: "$\geq$ all in block" and "$\geq$ others" provide very similar ordering of systems.

"$>$ all in block" suggests its potential in the context of system combination: CU-TECTO and PC-TRANS win almost one fifth of the blocks in which they appear, despite the fact that either a reference translation or a combination system already appears alongside them. (See also Table 4 below.)

Also, note that if the ranking task were designed specifically to cater to the "$\geq$ all in block" interpretation, it would only have **two** 'rank' labels (basically, "top" and "non-top"). In that case, annotators would spend *considerably* less time per block than they do now, since all they need to do is identify the top system(s) per block, without distinguishing non-top systems from each other.

Even for those interested in distinguishing non-state-of-the-art systems from each other, we point out that the "$\geq$ all in block" interpretation ultimately gives a system ordering that is very similar to that of the official "$\geq$ others" interpretation, **even** for the lower-tier systems (Figure 1).

2.2 Annotator Agreement

The WMT10 overview paper reports inter- and intra-annotator agreement over the pairwise comparisons, to show the validity of the evaluation setup and the "$\geq$ others" metric. Agreement is quantified using the following formula:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \quad (1)$$

where $P(A)$ is the proportion of times two annotators are observed to agree, and $P(E)$ is the expected proportion of times two annotators would agree by chance. Note that κ has a value of at most 1, with higher κ values indicating higher rates of agreement. The κ measure is more meaningful

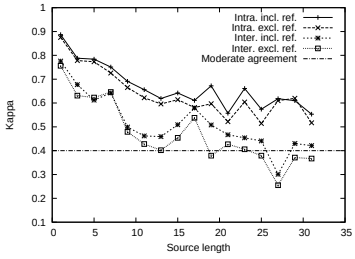


Figure 2: Intra-/inter-annotator agreement with/without references, across various source sentence lengths (lengths of n and $n+1$ are used to plot the point at $x = n$). This figure is based on all language pairs.

than reporting $P(A)$ as is, since it takes into account, via $P(E)$, how 'surprising' it is for annotators to agree in the first place.

In the context of pairwise comparisons, an agreement between two annotators occurs when they compare the same pair of systems (S_1,S_2), and both agree on their relative ranking: either $S_1 > S_2$, $S_1 = S_2$, or $S_1 < S_2$. $P(E)$ is then:

$$P(E) = P^2(s_1>s_2) + P^2(s_1=s_2) + P^2(s_1<s_2) \quad (2)$$

In the WMT overview paper, all three categories are assumed equally likely, giving $P(E) = \frac{1}{9} + \frac{1}{9} + \frac{1}{9} = \frac{1}{3}$. For consistency with the WMT overview paper, and unless otherwise noted, we also use $P(E) = \frac{1}{3}$ whenever a κ value is reported. (Though see Section 2.2.2 for a discussion about $P(E)$.)

2.2.1 Observed Agreement for Different Sentence Lengths

In Figure 2 we plot the κ values across different source sentence lengths. We see that the inter-annotator agreement (when excluding references) is reasonably high only for sentences up to 10 words in length – according to Landis and Koch (1977), and as cited by the WMT overview paper, not even 'moderate' agreement can be assumed if κ is less than 0.4. Another popular (and controversial) rule of thumb (Krippendorff, 1980) is more strict and says that $\kappa < 0.67$ is not suitable even for tentative conclusions.

For this reason, and given that a majority of sentences are indeed more than 10 words in length (the median is 20 words), we suggest that future evaluations either include fewer outputs per block, or divide longer sentences into shorter segments (e.g. on clause boundaries), so these segments are more easily and reliably comparable. The latter suggestions assumes word alignment as a preprocessing and presenting the annotators the context of the judged segment.

2.2.2 Estimating $P(E)$, the Expected Agreement by Chance

Several agreement measures (usually called kappas) were designed based on the Equation 1 (see Artstein and Poesio (2008) and Eugenio and Glass (2004) for an overview and a discussion). Those measures differ from each other in how to define the individual components of Equation 2, and hence differ in what the expected agreement by chance ($P(E)$) would be:[8]

- The S measure (Bennett et al., 1954) assumes a uniform distribution over the categories.
- Scott's π (Scott, 1955) estimates the distribution empirically from *actual annotation*.
- Cohen's κ (Cohen, 1960) estimates the distribution empirically as well, and further assumes *a separate distribution for each annotator*.

Given that the WMT10 overview paper assumes that the three categories ($S_1 > S_2$, $S_1 = S_2$, and $S_1 < S_2$) are equally likely, it is using the S measure version of Equation 1, though it does not explicitly say so – it simply calls it "the kappa coefficient" (K).

Regardless of what the measure should be called, we believe that the uniform distribution itself is not appropriate, even though it seems to model a "random clicker" adequately. In particular, and given the design of the ranking interface, $\frac{1}{3}$ is an overestimate of $P(S_1 = S_2)$ for a random clicker, and should in fact be $\frac{1}{5}$: each system receives one of five rank labels, and for two systems to receive the same rank label, there are only five (out of 25) label pairs that satisfy $S_1 = S_2$. Therefore, with $P(S_1 = S_2) = \frac{1}{5}$,

[8] These three measures were later generalized to more than two annotators (Fleiss, 1971; Bartko and Carpenter, 1976), Thus, without loss of generality, our examples involve two annotators.

"≥ Others"		S	π
Inter	incl. ref.	0.487	0.454
	excl. ref.	0.439	0.403
Intra	incl. ref.	0.633	0.609
	excl. ref.	0.601	0.575

Table 3: Summary of two variants of kappa: S (or K as it is reported in the WMT10 paper) and our proposed Scott's π. We report inter- vs. intra-annotator agreement and collected from all comparisons ("incl. ref.") vs. collected only from comparisons without the reference ("excl. ref.") because it is generally easier to agree that the reference is better than the other systems. This table is based on all language pairs.

we have $P(S_1 > S_2) = P(S_1 < S_2) = \frac{2}{5}$, and therefore $P(E) = 0.36$ rather than 0.333.

Taking the discussion a step further, we actually advocate following the idea of Scott's π, whereby the distribution of each category is estimated *empirically from the actual annotation*, rather than assuming a random annotator – these frequencies are easy to compute, and reflect a more meaningful $P(E)$.[9]

Under this interpretation, $P(S_1 = S_2)$ is calculated to be 0.168, reflecting the fraction of pairwise comparisons that correspond to a tie. (Note that this further supports the claim that setting $P(S_1 = S_2) = \frac{1}{3}$ for a random clicker, as used in the WMT overview paper, is an overestimate.) This results in $P(E) = 0.374$, yielding, for instance, $\pi = 0.454$ for "$\geq$ others" inter-annotator agreement, somewhat lower than $\kappa = 0.487$ (reported in Table 3).

We do note that the difference is rather small, and that our aim is to be mathematically sound above all. Carefully defining $P(E)$ would be important when comparing kappas across different tasks with different $P(E)$, or when attempting to satisfy certain thresholds (as the cited 0.4 and 0.67). Furthermore, if one is interested in measuring agreement for individual annotators, such as identifying those who have unacceptably low intra-annotator agreement, the question of $P(E)$ is quite important, since annotation behavior varies noticeably from one annotator to another. A 'conservative' annotator who prefers to rank systems as being tied most of the time would have a high $P(E)$, whereas an annotator using ties moderately would have a low $P(E)$. Hence, two annotators with equal agreement rates ($P(A)$) are not necessarily equally proficient, since their $P(E)$ might differ considerably.[10]

[9] We believe that $P(E)$ should not reflect the chance that two *random* annotators would agree, but the chance that two **actual** annotators would agree *randomly*. The two sound subtly related but are actually quite different.

2.3 The $\geq$ variant vs. the $>$ variant

Even within the same interpretation of how systems could be scored, there is a question of whether or not to reward ties. The overview paper reports both variants of its measure, but does not note that there are non-trivial differences between the two orderings. Compare for example the "$\geq$ others" ordering vs. the "$>$ others" ordering of CU-BOJAR and PC-TRANS (Table 2), showing an unexpected swing of 7.9%:

	$\geq$ others	$>$ others
CU-BOJAR	**65.6**	45.0
PC-TRANS	62.1	**49.4**

CU-BOJAR seems better under the $\geq$ variant, but loses out when only strict wins are rewarded. Theoretically, this could be purely due to chance, but the total number of pairwise comparisons in "$\geq$ others" is relatively large (about 1,500 pairwise comparisons for each system), and ought to cancel such effects.

A similar pattern could be seen under the "all in block" interpretation as well (e.g. for CU-TECTO and ONLINEB). Table 4 documents this effect by looking at how often a system is the sole winner of a block. Comparing PC-TRANS and CU-BOJAR again, we see that PC-TRANS is up there with CU-TECTO and DCU-COMBO as the most frequent sole winners, winning 71 blocks, whereas CU-BOJAR is the sole winner of only 53 blocks. This is in spite of the fact that PC-TRANS actually appeared in slightly fewer blocks than CU-BOJAR (385 vs. 401).

One possible explanation is that the two variants ("$\geq$" and "$>$") measure two subtly different things about MT systems. Digging deeper into Table 2's values, we find that CU-BOJAR is tied with another system $65.6 - 45.0 = 20.4\%$ of the time, while PC-TRANS is tied with another system only $62.1 - 49.4 = 12.7\%$ of the time. So it seems that PC-TRANS's output is *noticeably different* from another system more frequently than CU-BOJAR, which reduces the number of times that annotators

[10] Who's more impressive: a psychic who correctly predicts the result of a coin toss 50% of the time, or a psychic who correctly predicts the result of a *die roll* 50% of the time?

Blocks	Sole Winner
305	Reference
73	CU-TECTO
71	PC-TRANS
70	DCU-COMBO
57	RWTH-COMBO
54	ONLINEB
53	CU-BOJAR
46	EUROTRANS
41	UEDIN
41	UPV-COMBO
175	One of eight other systems
409	No sole winner
1395	Total English-to-Czech Blocks

Table 4: A breakdown of the 1,395 blocks for the English-Czech task, according to which system (if any) is the sole winner. On average, a system appears in 388 blocks.

mark PC-TRANS as tied with another system.[11] In that sense, the "≥" ranking is hurting PC-TRANS, since it does not benefit from its small number of ties. On the other hand, the ">" variant would not reward CU-BOJAR for its large number of ties.

The "≥ others" score may be artificially boosted if several very similar systems (and therefore likely to be "tied") take part in the evaluation.[12] One possible solution is to completely disregard ties and calculate the final score as $\frac{\text{wins}}{\text{wins+losses}}$. We recommend to use this score instead of "≥ others" ($\frac{\text{wins+ties}}{\text{wins+ties+losses}}$) which is biased toward often tied systems, and "> others" ($\frac{\text{wins}}{\text{wins+ties+losses}}$) which is biased toward systems with few ties.

2.4 Surprise? Does the Number of Evaluations Affect a System's Score?

When examining the system scores for the English-Czech task, we noticed a surprising pattern: it seemed that the more times a system is sampled to be judged, the lower its "≥ others" score ("≥ all in block" behaving similarly). A scatter plot of a system's score vs. the number of blocks in which it appears (Figure 3) makes the pattern obvious.

We immediately wondered if the pattern holds in other language pairs. We measured Pearson's correlation coefficient within each language pair, reported in Table 5. As it turns out, English-Czech happened to be the one language pair where the 'correlation' is strongest, with only English-Spanish also having a somewhat strong correlation. Overall, though, there is a consistent trend that can be seen across the language pairs. Could it really be the case that the more often a system is judged, the worse its score gets?

[11] Indeed, PC-TRANS is a commercial system (manually) tuned over a long period of time and based on resources very different from what other participants in WMT use.

[12] In the preliminary WMT11 results, this seems to happen to four Moses-like systems (UEDIN, CU-BOJAR, CU-MARECEK and CU-TAMCHYNA) which have better "≥ others" score but worse "> others" score than CU-TECTO.

Source	Target	Correlation of Block Count vs. "≥ Others"
English	Czech	-0.558
English	Spanish	-0.434
Czech	English	-0.290
Spanish	English	-0.240
English	French	-0.227
English	German	-0.161
French	English	-0.024
German	English	0.146
Overall		-0.092

Table 5: Pearson's correlation between the number of blocks where a system was ranked and the system's "≥ others" score. (The reference itself is not included among the considered systems).

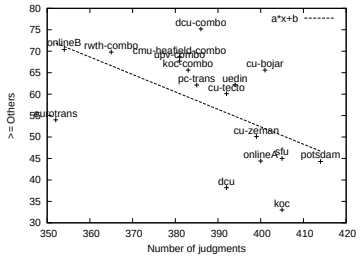


Figure 3: A plot of "≥ others" system score vs. times judged, for English-Czech.

Examining plots for the other language pairs makes things a bit clearer. Consider for example the plot for English-Spanish (Figure 4). As one would hope, the data points actually come together to form a cloud, **indicating a lack of correlation**. The reason that a hint of a correlation exists is the presence of two outliers in the bottom right corner. In other words, the **very** worst systems are, indeed, the ones judged quite often. We observed this pattern in several other language pairs as well.

The correlation naturally does not imply causation. We are still not sure how to explain the artifact. A subtle possibility lies in the MTurk interface: annotators have the choice to accept a HIT or skip it before actually providing their la-

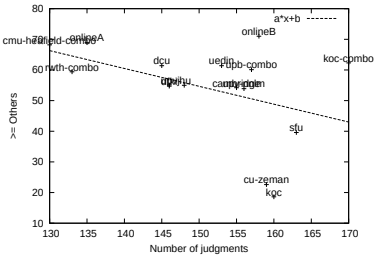


Figure 4: A plot of "≥ others" system score vs. times judged, for English-Spanish.

	REF	CU-BOJAR	CU-TECTO	EUROTRANS	ONLINEB	PC-TRANS	UEDIN
REF	-	4.3	4.3	5.1	3.8	3.6	2.3
CU-BOJAR	**87.1**	-	**45.7**	28.3	**44.4**	39.5	**41.1**
CU-TECTO	**88.2**	35.8	-	38.0	**55.8**	**44.0**	36.0
EUROTRANS	**88.5**	**60.9**	**46.8**	-	**50.7**	**53.8**	**48.6**
ONLINEB	**91.2**	31.1	29.1	32.8	-	43.8	**39.3**
PC-TRANS	**88.0**	**45.3**	42.9	28.6	**49.3**	-	36.6
UEDIN	**94.3**	39.3	**44.2**	31.9	32.1	**49.5**	-

Table 6: Pairwise comparisons extracted from sentence-level rankings of the WMT10 English-Czech News Task. Re-evaluated to reproduce the numbers published in WMT10 overview paper. Bold in column A and row B means that system A is pairwise better than system B.

bels. It might be the case that some annotators are more willing to accept HITs when there is an obviously poor system (since that would make their task somewhat easier), and who are more prone to skipping HITs where the systems seem hard to distinguish from each other. So there might be a causation effect after all, but in the reverse order: a system gets judged more often if it is a bad system.[13] A suggestion from the reviewers is to run a pilot annotation with deliberate inclusion of a poor system among the ranked ones.

2.5 Issues of Pairwise Judgments

The WMT overview paper also provides pairwise system comparisons: each cell in Table 6 indicates the percentage of pairwise comparisons between the two systems where the system in the column was ranked better (>) than the system in the row. For instance, there are 81 ranking responses where both CU-TECTO and CU-BOJAR were present and indeed ranked[14] among the 5 systems in the block. In 37 (45.7%) of the cases, CU-TECTO was ranked better, in 29 (35.8%), CU-BOJAR was ranked better and there was a tie in the remaining 15 (18.5%) cases. The ties are not explicitly shown in Table 6 but they are implied by the total of 100%. The cell is in bold where there was a win in the pairwise comparison, so 45.7 is bold in our example.

An interesting "discrepancy" in Table 6 is that CU-TECTO wins pairwise comparisons with CU-BOJAR and UEDIN but it scores worse than them in the official "≥ others", cf. Table 2. Similarly, UEDIN outperformed ONLINEB in the pairwise comparisons but it was ranked worse in both > and ≥ official comparison.

In the following, we focus on the CU-BOJAR (B) and CU-TECTO (T) pair because they are interesting competitors on their own. They both use the same parallel corpus for lexical mapping but operate very differently: CU-BOJAR is based on Moses while CU-TECTO transfers at a deep syntactic layer and generates target text which is more or less grammatically correct but suffers in lexical choice.

2.5.1 Different Set of Sentences

The mismatch in the outcomes of "≥ others" and pairwise comparisons could be caused by different set of sentences. The pairwise ranking is collected from the set of blocks where both CU-BOJAR and CU-TECTO appeared (and were indeed ranked). Each of the systems however competes in other blocks as well, which contributes to the official "≥ others".

The set of sentences underlying the comparison is very different and more importantly that the basis for pairwise comparisons is much smaller than the basis of the official "≥ others" interpretation. The outcome of the official interpretation however depends on the random set of systems your system was compared to. In our case, it is impossible to distinguish, whether CU-TECTO had just bad luck on sentences and systems it was compared to when CU-BOJAR was not in the block and/or whether the 81 blocks do not provide a reliable picture.

2.5.2 Pairwise Judgments Unreliable

To complement WMT10 rankings for the two systems and avoid the possible lower reliability due to 5-fold ranking instead of a targeted compari-

[13] No pun intended!

[14] The users sometimes did not fill any rank for a system. Such cases are ignored.

		Author of B says:				
		B>T	T>B	both fine	both wrong	Total
T says:	B>T	9	-	1	1	11
	T>B	2	13	-	3	18
	both fine	2	-	2	3	7
	both wrong	10	5	1	11	27
	Total	23	18	4	18	63

Table 7: Additional annotation of 63 CU-BOJAR (B) vs. CU-TECTO (T) sentences by two annotators.

	Better		Both	
Annotator	B	T	fine	wrong
A	**24**	23	5	11
C	10	**12**	5	36
D	**32**	20	2	9
M	11	**18**	7	27
O	**23**	18	4	18
Z	25	**27**	2	9
Total	**125**	118	25	110

Table 8: Blurry picture of pairwise rankings of CU-BOJAR vs. CU-TECTO. Wins in bold.

son, we asked the main authors of both CU-BOJAR and CU-TECTO to carry out a *blind* pairwise comparison on the exact set of 63 sentences appearing across the 81 blocks in which both systems were ranked. As the totals in Table 7 would suggest, each author unwittingly recognized his system and slightly preferred it. The details however reveal a subtler reason for the low agreement: one of the annotators was less picky about MT quality and accepted 10+5 sentences completely rejected by the other annotator. In total, these two annotators agreed on $9 + 13 + 2 + 11 = 35$ (56%) of cases and their pairwise κ is 0.387.

A further annotation of these 63 sentences by four more people completes the blurry picture: the pairwise κ for each pair of our five annotators ranges from 0.242 to 0.615 with the average 0.407±0.106. The multi-annotator κ (Fleiss, 1971) is 0.394 and all six annotators agree on a single label only in 24% of cases. The agreement is not better even if we merge the categories "Both fine" and "Both wrong" into a single one: The pairwise κ ranges from 0.212 to 0.620 with the average 0.405±0.116, the multi-annotator κ is 0.391. Individual annotations are given in Table 8.

Naturally, the set of these 63 sentences is not a representative sample. Even if one of the systems actually won, such an observation could not have been generalized to other test sets. The purpose of the exercise was to check whether we are *at all* able to agree which of the systems translates this specific set of sentences better. As it turns out, even a simple pairwise ranking can fail to provide an answer because different annotators simply have different preferences.

SRC	It's not completely ideal.		
REF	Není to úplně ideální.	Ranks	
PC-TRANS	To není úplně ideální.	2	5
CU-BOJAR	To není úplně ideální.	5	4

Table 9: Two rankings by the same annotator.

SRC	FCC awarded a tunnel in Slovenia for 64 million
REF	FCC byl přidělen tunel ve Slovinsku za 64 milionů
Gloss	FCC **was** awarded a tunnel in Slovenia for 64 million
HYP1	FCC přidělil tunel ve Slovinsku za 64 miliónů
HYP2	FCC přidělila tunel ve Slovinsku za 64 milionů
Gloss	FCC awarded$^{\text{masc}}_{\text{fem}}$ a tunnel in Slovenia for 64 million

Figure 5: A poor reference translation confuses human judges. The SRC and REF differ in the active/passive form, attributing completely different roles to "FCC".

Finally, Table 9 illustrates how poor the WMT10 rankings can be. The exact same string produced by two systems was ranked differently each time – by the same annotator. (The hypothesis is a plausible translation, only the information structure of the sentence is slightly distorted so the translation may not fit well it the surrounding context.)

3 The Impact of the Reference Translation

3.1 Bad Reference Translations

Figure 5 illustrates the impact of poor reference translation on manual ranking as carried out in Section 2.5.2. Of our six independent annotations, three annotators marked the hypotheses as "both fine" given the match with the source and three annotators marked them as "both wrong" due to the mismatch with the reference. Given the construction of the WMT test set, this particular sentence comes from a Spanish original and it was most likely translated directly to both English and Czech.

Source	Target	Correlation of Reference vs. "≥ others"
Spanish	English	0.341
English	French	0.164
French	English	0.098
German	English	0.088
Czech	English	-0.041
English	Czech	-0.145
English	Spanish	-0.411
English	German	-0.433
Overall		-0.107

Table 10: Pearson's correlation of the relative percentage of blocks where the reference was included in the ranking and the final "≥ others" of the system (the reference itself is not included among the considered systems).

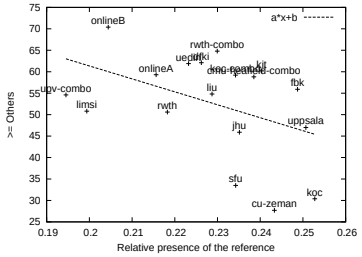


Figure 6: Correlation of the presence of the reference and the official "≥ others" for English-German evaluation.

3.2 Reference Can Skew Pairwise Comparisons

The exact set of competing systems in each 5-fold ranking in WMT10 evaluation is random. The "≥ others" however is affected by this: a system may suffer more losses if often compared to the reference, and similarly it may benefit from being compared to a poor competitor.

To check this, we calculate the correlation between the relative presence of the reference among the blocks where a system was judged and the system's official "≥ others" score. Across language, there is almost no correlation (Pearson's coefficient: -0.107). However, for some language pairs, the correlation is apparent, as listed in Table 10. Negative correlation means: the more often the system was compared along with the reference, the worse the score of the system.

Figure 6 plots the extreme case of English-German evaluation.

Source	Target	Min	Avg±StdDev	Max
English	Czech	40	65±19	115
English	French	40	66±17	110
English	German	10	40±16	80
English	Spanish	30	54±15	85
Czech	English	5	38±13	60
French	English	5	37±15	70
German	English	10	32±12	65
Spanish	English	35	56±11	70

Table 11: The number of post-edits per system for each language pair to complement Figure 3 (page 12) of the WMT10 overview paper.

4 Other WMT10 Tasks

4.1 Blind Post-Editing Unreliable

WMT often carries out one more type of manual evaluation: "Editing the output of systems without displaying the source or a reference translation, and then later judging whether edited translations were correct." (Callison-Burch et al., 2010). We call the evaluation "blind post-editing" for short.

We feel that blind post-editing is more informative than system ranking. First, it constitutes a unique comprehensibility test, and after all, MT should aim at comprehensible output in the first place. Second, blind post-editing can be further analyzed to search for specific errors in system output, see Bojar (2011) for a preliminary study.

Unfortunately, the amount of post-edits collected in WMT10 varied a lot across systems and language pairs. Table 11 provides the minimum, average and maximum number of post-edits of outputs of a particular MT system. We see that e.g. while English-to-Czech has many judgments of this kind per system, Czech-to-English is one of the worst supported directions.

It is not surprising that conclusions based on 5 observations can be extremely deceiving. For instance CU-BOJAR seems to produce 60% of outputs comprehensible (and thus wins in Figure 3 on page 12 in the WMT overview paper), far better than CMU. This is not in line with the ranking results where both rank equally (Table 5 on page 10 in the WMT overview paper). In fact, CU-BOJAR was post-edited 5 times and 3 of these post-edits were acceptable while CMU was post-edited 30 times and 5 of these post-edits were acceptable.

4.2 A Remark on System Combination Task

One results of WMT10 not observed in previous years was that system combinations indeed performed better than individual systems. Previous

	Dev Set	Test Set	
Sententes	455	2034	Diff
GOOGLE	17.32±1.25	16.76±0.60	↘
BOJAR	16.00±1.15	16.90±0.61	↗
TECTOMT	11.48±1.04	13.19±0.58	↗
PC-TRANS	10.24±0.92	10.84±0.46	↗
EUROTRAN	9.64±0.92	11.04±0.48	↗

Table 12: BLEU scores of sample five systems in English-to-Czech combination task.

years failed to show this clearly, because Google Translate used to be included among the combined systems, making it hard to improve. In WMT10, Google Translate was excluded from system combination task (except for translations involving Czech, where it was accidentally included).

Our Table 12 provides an additional explanation why the presence of Google among combined systems leads to inconclusive results. While the test set was easier (based on BLEU) than the development set for most systems, it was much harder for Google. All system combinations were thus likely to overfit and select Google n-grams most often. Without access to Google powerful language models, the combination systems were likely to underperform Google in final fluency of the output.

5 Further Issues of Manual Evaluation

We have already seen that the comprehensibility test by blind post-editing provides a different picture of the systems than the official ranking. Berka et al. (2011) introduced a third "quiz-based evaluation". The quiz-like evaluation used the English-to-Czech WMT10 systems, applied to different texts: short text snippets were translated and annotators were asked to answer three yes/no questions complementing each snippet. The order of the systems was rather different from the official WMT10 results: CU-TECTO won the quiz-based evaluation despite being the fourth in WMT10.

Because the texts were different in WMT10 and the quiz-based evaluation, we asked a small group of annotators to apply the ranking technique on the text snippets. While not exactly comparable to the WMT10 ranking, the WMT10 ranking was confirmed: CU-TECTO was again among the lowest-scoring systems and Google won the ranking.

Bojar (2011) applies the error-flagging manual evaluation by Vilar et al. (2006) to four systems of WMT09 English-to-Czech task. Again, the overall order of the systems is somewhat different when ranked by the number of errors flagged. Mireia Farrús and Fonollosa (2010) use a coarser but linguistically motivated error classification for Catalan-Spanish and suggest that differences in ranking are caused by annotators treating some types of errors as more serious.

In short, different types of manual evaluations lead to different results even when identical systems and texts are evaluated.

6 Conclusion

We took a deeper look at the results of the WMT10 manual evaluation, and based on our observations, we have some recommendations for future evaluations:

- We propose to use a score which ignores ties instead of the official "$\geq$ others" metric which rewards ties and "$>$ others" which penalizes ties. Another score, "$\geq$ all in block", could help identify which systems are more dominant.
- Inter-annotator agreement decreases dramatically with sentence length; we recommend including fewer sentences per block, at least for longer sentences.
- We suggest agreement be measured based on an empirical estimate of $P(E)$, or at least using a more correct random clicking $P(E) = 0.36$.
- There is evidence of a negative correlation between the number of times a system is judged and its score; we recommend a deeper analysis of this issue.
- We recommend the reference be sampled at a lower rate than other systems, so as to play a smaller role in the evaluation. We also recommend better quality control over the production of the references.

And to the readers of the WMT overview paper, we point out:

- Pairwise comparisons derived from 5-fold rankings are sometimes unreliable. Even a targeted pairwise comparison of two systems can shed little light as to which is superior.
- The acceptability of post-edits is sometimes very unreliable due to the low number of observations.

References

R. Artstein and M. Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

John J. Bartko and William T. Carpenter. 1976. On the methods and theory of reliability. *Journal of Nervous and Mental Disease*, 163(5):307–317.

E. M. Bennett, R. Alpert, and A. C. Goldstein. 1954. Communications through limited questioning. *Public Opinion Quarterly*, 18(3):303–308.

Jan Berka, Martin Černý, and Ondřej Bojar. 2011. Quiz-Based Evaluation of Machine Translation. *Prague Bulletin of Mathematical Linguistics*, 95:77–86, March.

Ondřej Bojar. 2011. Analyzing Error Types in English-Czech Machine Translation. *Prague Bulletin of Mathematical Linguistics*, 95:63–76, March.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden, July. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Barbara Di Eugenio and Michael Glass. 2004. The kappa statistic: A second look. *Computational linguistics*, 30(1):95–101.

J. L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Beverly Hills, CA. Chapter 12.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174.

José B. Mariño Mireia Farrús, Marta R. Costa-jussà and José A. R. Fonollosa. 2010. Linguistic-based evaluation criteria to identify statistical machine translation errors. In *Proceedings of the 14th Annual Conference of the Euoropean Association for Machine Translation (EAMT'10)*, pages 167–173, May.

William A. Scott. 1955. Reliability of content analysis: The case of nominal scale coding. *Public Opinion Quarterly*, 19(3):321–325.

David Vilar, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error Analysis of Machine Translation Output. In *International Conference on Language Resources and Evaluation*, pages 697–702, Genoa, Italy, May.

Ten Years of WMT Evaluation Campaigns: Lessons Learnt

Ondřej Bojar, Christian Federmann, Barry Haddow, Philipp Koehn, Matt Post, Lucia Specia
Charles University in Prague, Microsoft Research, University of Edinburgh, University of Edinburgh/ JHU, JHU, University of Sheffield
bojar@ufal.mff.cuni.cz, chrife@microsoft.com, bhaddow@inf.ed.ac.uk, phi@jhu.edu
post@cs.jhu.edu, l.specia@sheffield.ac.uk

Abstract

The WMT evaluation campaign (`http://www.statmt.org/wmt16`) has been run annually since 2006. It is a collection of shared tasks related to machine translation, in which researchers compare their techniques against those of others in the field. The longest running task in the campaign is the translation task, where participants translate a common test set with their MT systems. In addition to the translation task, we have also included shared tasks on evaluation: both on automatic metrics (since 2008), which compare the reference to the MT system output, and on quality estimation (since 2012), where system output is evaluated without a reference. An important component of WMT has always been the manual evaluation, wherein human annotators are used to produce the official ranking of the systems in each translation task. This reflects the belief of the WMT organizers that human judgement should be the ultimate arbiter of MT quality. Over the years, we have experimented with different methods of improving the reliability, efficiency and discriminatory power of these judgements. In this paper we report on our experiences in running this evaluation campaign, the current state of the art in MT evaluation (both human and automatic), and our plans for future editions of WMT.

Keywords: Machine Translation, Evaluation, Shared Tasks

1. Introduction

The First Workshop in Statistical Machine Translation was held in 2006, and it has been held annually since then, becoming the First WMT Conference in Machine Translation (WMT 2016) this year. In the first year of WMT there was a shared translation task which attracted 12 task description papers. In 2015 there were 5 different tasks and 46 task description papers, whilst in 2016 there will be 10 different tasks, covering translation of text and images, handling of pronouns in translation, MT evaluation, system tuning, automatic post-editing and document alignment.

The core component of WMT has been the main translation task (which in most years is the only translation task). The first translation task used Europarl (Koehn, 2005) for the test set; since then, we have constructed the test set from news text, with the complex structure and broad topic coverage providing a significant challenge to MT systems. Since 2009 the news test sets have been created specifically for the shared task, by crawling news articles in various languages and translating to the other task languages, providing the MT research community with valuable resources for future research. We have also varied the language pairs from year to year to present different challenges to researchers, although there has always been an emphasis on European languages. The language pairs included in each year's evaluation are shown in Table 1.

A central theme in the WMT shared tasks has been the evaluation of MT. We have explored this extensively, focusing on both human and automatic evaluation. The main translation task has always employed large-scale human evaluation to determine the quality and ranking of the systems; how precisely this is done has varied over the years (Section 2.). The human ranking has enabled the development of automatic metrics by providing a gold standard against which metrics can be compared. Since 2008, the metrics task has asked participants to develop tools to evaluate MT output against one or more references (Section 3.). In 2012, we introduced the quality estimation task, which takes metrics a step further, attempting to evaluate the quality of MT output without use of a reference (Section 4.).

2. Manual Evaluation

Since the very beginning, WMT organizers have taken the position that machine translation performance should be evaluated from time to time against human opinion:

> *While automatic measures are an invaluable tool for the day-to-day development of machine translation systems, they are only a imperfect substitute for human assessment of translation quality* ... (Koehn and Monz, 2006)

This is not to disparage automatic metrics, which have played a crucial role in the progress of the field and the improvement of MT quality over time. It is only to say that they are at best a proxy for what we really care about, and must be regularly anchored to human opinion. The WMT therefore produces an annual *human ranking of systems* for each task, from best to worst. In addition to helping direct researchers to the systems whose features they might wish to copy, this gold-standard system ranking is used to evaluate automatic metrics (a metric metric).

Of course, the question of which system is the best or worst is a fraught one. There are any number of answers, and subsequent questions. The first is best *for what purpose*? For a person trying to understand a foreign-language news article, an MT system that can convey the gist of an article is necessary, but quality might need to be sacrificed for speed. On the other hand, a student trying to learn how to translate an article may require a system that can also correctly generate grammatical and natural-sounding sentences. Evaluations are often broken down along these concepts of *adequacy* and *fluency*.

In fact, in the first two editions of the WMT shared translation task we used adequacy/fluency judgements on a 5-point scale as our main evaluation measure. Not satisfied with the results though, we started experimenting with

Proceedings of the LREC 2016 Workshop "Translation Evaluation – From Fragmented Tools and Data Sets to an Integrated Ecosystem", Georg Rehm, Aljoscha Burchardt et al. (eds.)

Language Pair	'06	'07	'08	'09	'10	'11	'12	'13	'14	'15	'16
Czech ↔ English		•	•	•	•	•	•	•	•	•	•
Finnish ↔ English										•	•
French ↔ English	•	•	•	•	•	•	•	•	•	•	
German ↔ English	•	•	•	•	•	•	•	•	•	•	•
German ↔ Spanish			•								
Haitian Creole → English						•					
Hindi ↔ English									•		
Hungarian ↔ English			•	•							
Romanian ↔ English											•
Russian ↔ English								•	•	•	•
Spanish ↔ English	•	•	•	•	•	•	•	•			
Turkish ↔ English											•

Table 1: Language pairs in the main translation task.

Metric	'06	'07	'08	'09	'10	'11	'12	'13	'14	'15	'16
Adequacy / Fluency	•	•									
Sentence Ranking		•	•	•	•	•	•	•	•	•	•
Constituent Ranking		•	•								
Constituent Judgement (Y/N)			•								
Sentence Comprehension				•	•			○			
Direct Assessment											•
Used MTurk					•		•	•			•

Table 2: Metrics used in the human evaluation over the years for all languages pair (•) or only English → Czech (○).

other methods and over the years, WMT has tried several different ones, encoded in different evaluations, summarized in Table 2. Brief explanations of the approaches follow:

- *Fluency / Adequacy*. Annotators were presented with a sentence, and were asked to rank it separately for both fluency and adequacy, on five-point scales.
- *Sentence Ranking*. Annotators are presented with the outputs of multiple systems, along with the source and reference sentence, and asked to rank them, from best to worst.
- *Constituent Ranking*. Annotators were asked to rank the quality of the translations of automatically-identified constituents, instead of the complete sentences.
- *Constituent Judgement (Y/N)*. Annotators were asked to provide a binary judgement on the suitability of the translation of a constituent.
- *Sentence Comprehension*. Annotators were asked to edit MT output for fluency (without providing the reference), and then (separately) to determine via binary judgement whether those edits resulted in good translations.
- *Direct Assessment (DA)*. Annotators are asked to provide a direct assessment of the quality of a single MT output compared to a single reference, using an analog scale.

The adequacy/fluency judgements were abandoned as the 5-point measurements proved to be quite inconsistent and hard to normalize, and they were not popular with the annotators. Viewing the distributions of scores provided by individual annotators showed them to be very different in shape, often skewed in different directions, so there was no clear way to combine judgements from multiple annotators. There was also complaints from annotators about the extreme difficulty in annotating long sentences of, frequently scrambled, MT output.

Two early measures of quality focused only on noun phrase constituents that were automatically identified in the reference and then extracted from system outputs via projections across automatic alignments. Constituent ranking (2007–2008) asked annotators to compare and rank these constituents, while binary constituent judgements (2008) asked them only whether a constituent (provided in context and approximately highlighted) were "acceptable" compared to the reference. An advantage of these binary judgements was very high annotator agreement rates; this is likely due in part to their relatively short length.

Another means of directly assessing output quality (and thereby inferring a system ranking) is Sentence Comprehension, used in 2009 and 2010. In this task, one set of judges was asked to edit a sentence's fluency (without access to the source or reference); these edited sentences were then later evaluated to see whether they "represent[ed] fully fluent and meaning-equivalent alternatives to the reference sentence". This mode of evaluation did not correlate well with relative ranking, however, and was abandoned in 2011 in order to focus annotators' efforts on that method.

In an effort to find a better evaluation method, we introduced Sentence Ranking in 2007. One big advantage of Sentence Ranking is that it is conceptually very simple: of-

fer the annotator two samples of MT output (and a reference) and ask them which they prefer. In practice, in order to gather judgements more efficiently, we present the annotator with 5 different MT outputs at a time, which then yields ten pairwise comparisons. We have experimented with presenting more or fewer sentences at a time, but 5 seems to be a good compromise between efficiency and reliability. We have also experimented with collecting judgements on Amazon's Mechanical Turk (2012 and 2013), in an effort to reduce the effort required from researchers. While relatively effective, the effort required to ensure that the work was completed faithfully, and the even lower annotator agreement rates, caused us to abandon it.

Since 2011, Sentence Ranking has been the only method of human evaluation we have used, but during that time the details have evolved in response to criticism. In particular, Bojar et al. (2011) pointed out various problems with the way the comparisons were collected and interpreted which led to changes in the procedure. A particular problem with Sentence Ranking is that the method involves collecting *relative* judgements of MT performance, but attempts to combine these to give an *absolute* measure of translation performance. Unless a sufficient number of carefully chosen comparisons are made, then systems can be treated unfairly by being compared too often to a very bad, or very good system (or the reference, which may be in there for control). Furthermore, systems were getting credit for ties, so systems which were very similar to others were doing better than they should. Finally, Bojar et al. (2011) showed that the agreement on the Sentence Ranking task falls off rapidly as sentence length increased.

Further analysis of the Sentence Ranking approach was provided by Lopez (2012) who pointed out the difficulties in obtaining a reliable total ordering of systems from the pairwise judgements. Further work (Koehn, 2012) suggested that we really needed to collect more judgements in order to display significant differences between the systems, and also established a means of clustering systems into equivalence classes of mutually indistinguishable systems, based on bootstrap resampling. Thus, since 2013, the system rankings have been presented as a partial ordering over systems, instead of a total ordering, where systems in the same group are considered to be tied. (However, the total ordering is still used when evaluating metrics).

One important point has not been addressed. Over the years, WMT has experimented with many different means of producing a system ranking. These rankings are then used as a gold standard for metrics tasks, and are also published as an official ranking, which researchers make use of in determining which system description papers to plumb for ideas to improve their own systems. Each year, different methods have been evaluated and then kept or discarded according to a number of criteria, such as annotator agreement numbers, or time spent. However, how can we really know which of these is the best? This point was raised by Hopkins and May (2013), who then provided a Bayesian model formulation of the human ranking problem, which allowed them to use perplexity to compare different system rankings. Influenced by this idea, in 2014, we compared the ability of three different models trained on a large set of pairwise rankings, using accuracy on held-out comparisons instead of perplexity. The method that won was a new approach that based on the TrueSkill algorithm (Sakaguchi et al., 2014). This has been in use since.

To conclude, the WMT manual evaluation has engaged in a deep and extensive experimentation over the years. The Sentence Ranking task has formed the core of our evaluation approach, and has seen many variations from year to year. We have made progress on many of the problems with evaluation. However, many problems remain: the relatively low annotator agreement rates, the immense amount of annotator time required, and the difficulty of scaling the sentence ranking task to many systems. In 2016, we plan to run a pilot investigation based on Direct Assessment of machine translation quality, which we hope will further alleviate some of these issues.

3. Automatic Evaluation

Since the second year of the WMT campaigns, targeted effort was also devoted to evaluation of automatic metrics[1] of MT quality, or **metrics task** for short. This meta-evaluation is an important complement to the shared translation task, because automatic metrics are used throughout the development of MT systems and also in automatic system optimization (Neubig and Watanabe, 2016). The utility of some of the metrics in system optimization has been tested in the sister **tuning task** in 2011 and 2015 and also planned for 2016.

Metrics of MT quality are evaluated at two levels:

System-level evaluation tests, how well a metric can replicate the human judgement about the overall quality of MT systems on the given complete set of test set sentences.

Segment-level evaluation tests how well a metric can predict the human judgement for each input sentence.

In both cases, participants of the metrics task are given input sentences, outputs of MT systems and one reference translation. Note that the reliance on a single reference is not ideal. It is well known that the reliability of automatic MT evaluation methods is limited if only one reference is available (see the WMT 2013 overview paper for an empirical evaluation of BLEU with up to 12 references for translation into Czech). The quality estimation task (Section 4.) focuses on the setup where no reference is available at all.

Table 3 summarizes the participation and methods used to evaluate the system-level and segment-level parts of the task. The task had always received a good number of participating teams. The number of evaluated metrics varies considerably across the years, because in some years, multiple variations of some metrics were evaluated.

Starting from 2013, we distinguish "baseline metrics". These metrics are run by the organizer in addition to the submitted ones. Baseline metrics include the `mteval` scoring script and all the metrics available in Moses. We report the exact configuration flags for them, so they should be reliably reproducible.

Throughout the years, the metrics task has always relied on the manual evaluation (Section 2.), so the gold standard

[1] Despite the term "metrics", none of the measures or methods is a metric in the mathematical sense.

	'07	'08	'09	'10	'11	'12	'13	'14	'15	'16
Participating Teams	-	6	8	14	9	8	12	12	11	
Evaluated Metrics	11	16	38	26	21	12	16	23	46	
Baseline Metrics							5	6	7	
System-level evaluation methods										
Spearman Rank Correlation	•	•	•	•	•	•	•	○		
Pearson Correlation Coefficient							○	•	•	•
Segment-level evaluation methods										
Ratio of Concordant Pairs		•	•							
Kendall's τ				•	•	•	$*$	$\star$	$\star$	$\star$
Tuning Task					•				•	•

• main and ○ secondary score reported for the system-level evaluation.
•, $*$ and $\star$ are slightly different variants regarding ties.

Table 3: Summary of metrics tasks over the years.

human judgements do come from different styles of evaluation. A major move from Sentence Ranking to Direct Assessment is considered in 2016, which would particularly affect the segment-level metric evaluation. In Direct Assessment, the judgements have to be sampled differently from the system-level and segment-level evaluation, and there is a concern whether we will be able to find enough distinct speakers for each of the language pairs. Preliminary experiments are now under way.

3.1. How Metrics are Evaluated

As indicated in Table 3, the metrics task has seen a few changes of the exact evaluation method.

Evaluating System-Level Evaluation System-level methods were first evaluated using Spearman rank correlation, comparing the list of systems for a particular language pair as ordered by the metric (given the test set of sentences are reference translations) and as ordered by humans (on the sample of sentences from the test set that actually receive some human judgements). Spearman rank correlation was selected in the first year, because it is applicable also to the ordinal scales of adequacy and fluency which were used in 2006 and 2007. Since 2007, Pearson correlation coefficient could have been also used (as the system scores were on continuous scales), but the switch happened only in 2013. The benefit of Pearson over Spearman is that it considers the distances between the systems, so it should be more stable for systems of similar quality.

Evaluating Segment-Level Evaluation Segment-level evaluation has so far relied on pairwise judgements of translation quality. Given two candidate translations of an input sentence, the segment-level metric gets a credit if it agrees with the human judgement, i.e. the two pairwise judgements are "concordant". The exact calculation of the final score changed throughout the years: in 2008 and 2009, a simple ratio ranging from 0 to 1 was used: the number of concordant pairs out of the total number of pairs evaluated. Starting from 2010, the score was modified to penalize discordant pairs, falling under the general definition of Kendall rank correlation coefficient, or Kendall's τ for short, with $[-1, 1]$ as the range of possible values:

$$\tau = \frac{|Concordant| - |Discordant|}{|Concordant| + |Discordant|} \quad (1)$$

There has always been a question of how to handle tied comparisons, either the humans or the metric (or both) assigning the same rank/score to the two candidates. Each type of tied pairs can be included in the denominator and if it is, it may be also included in the numerator (bonified or penalized). After the discussion available in Macháček and Bojar (2013) and Macháček and Bojar (2014), the current method:

- ignores pairs where humans tied altogether,
- does not give any credit or bonus to pairs where the metric predicted a tie,
- but includes these metric-tied pairs in the denominator.

Moving to the Direct Assessment or some other absolute scale in the human evaluation would allow use to use Pearson correlation coefficient instead of Kendall's τ.

Significance From the beginning, it was not quite clear how to establish significance of the observed differences in metric evaluation, especially at the system level where the number of participating systems is less than 20, providing a low sample size.

Starting from 2013, system-level scores for each given language pair were reported with empirical confidence bounds constructed by resampling the "golden truth": given the complete set of human judgements, 1000 variations are constructed by resampling with repetition, leading to 1000 different scorings of the systems.[2] Each participating metric provides a single scoring of the systems and this scoring is correlated with the 1000 golden truths, giving us 1000 results reflecting the variance due to the set of sentences and annotators included in the golden truth.

As noticed by Graham and Liu (2016), confidence intervals obtained from this sampling cannot be used to infer whether one metric significantly outperforms another one, because the number of "significant" pairs would be overestimated. Instead, Graham and Liu (2016) proposes a novel method, artificially generating a large number of MT systems (by

[2]Many of these scorings share the same order of the systems. Unlike Spearman rank correlation, the Pearson correlation coefficient used since 2013 however appreciates also differences in the scores.

mixing the outputs of the real MT systems participating in the translation task) and asking metrics task participants to score e.g. not 5 but 10000 MT systems on the given test set. We will try to adopt this approach in 2016, testing in practice, how many metrics task participants can cope with these enlarged sets of MT systems.

3.2. Observations in Metrics Task

While metrics tasks across the years cannot be directly compared because a whole range of conditions keeps changing, the overall setting remains stable and some general observations can be made:

- BLEU has been surpassed by far by many diverse metrics. On the other hand, we acknowledge that it remains the most widely used and also scores on average well among the baseline metrics, with CDER (Leusch and Ney, 2008) being a competitor.
- The level of 0.9 of system-level correlation into English was reached by the best metrics in 2009, rising up to 0.98 in 2011. These levels were achieved by **aggregate** or **combination metrics** that include many features and standard metrics; sometimes the combination is **trained** on a past dataset. IQmt-ULCh, SVMrank (2010) and MTeRater-Plus (2011) are the early examples, followed by a row of other combination metrics in recent years (e.g. BEER, DPMFcomb, RATATOUILLE in 2014 or 2015). MTeRater is an interesting outlier in that its main component is based on many features from automatic essay scoring (preposition choice, collocations typical for native use, inflection errors, article errors).
- Benefits were confirmed many times from **including paraphrases or synonyms** incl. Wordnet (e.g. Meteor, Tesla in 2010 and 2011), refining the metric to consider the coverage of individual **parts of speech** (e.g. PosBLEU 2008, SemPOS 2009, 2012), focusing on **content words** (Tesla, SemPOS), **dependency relations** (already 2008) or **semantic roles** (already 2007), evaluating at the level of **character sequences** (i-letter-BLEU 2010, chrF 2015, BEER).
- In 2012, we saw a drop in into-English evaluation mainly due to a different set of participating metrics. Such a "**loss of wisdom**" is unfortunate and the baseline metrics run by the organizers are one of possible means to avoiding it. In an ideal world, the authors of the top performing metrics every year would incorporate their metrics to Moses, to ensure that the metric gets evaluated in the coming years. Achieving this state is obviously complicated by the reliance of some of the metrics on diverse language-dependent resources which are not always publicly available. Meteor remains the only such maintained metric throughout the years. Hopefully, some of the trivial but well-performing metrics based on characters (chrF, i-letter-BLEU) will join the baselines soon.

4. Quality Estimation

Quality Estimation (QE) offers an alternative way of assessing translation quality. QE metrics are fully automated and, unlike common evaluation metrics (Section 3.), do not rely on comparisons against human translations. QE metrics aim to provide predictions on translation quality for MT systems in use, for any number of unseen translations. They are trained metrics, built using supervised machine learning algorithms with examples of translations labelled for quality (ideally, by humans). Predictions can be provided at different granularity levels: word, phrase, sentence, paragraph or document. Different levels require different features, label types and algorithms to build prediction models.

While work on QE started back in the early 2000's (Blatz et al., 2004), the use of MT was substantially less widespread back then, and thus the need for this type of metric was less evident. A new surge of interest appeared later (Specia et al., 2009; Soricut and Echihabi, 2010), particularly motivated by the popularisation of MT in commercial settings.

QE was first organised as a shared task (and a track at WMT) in 2012 (Callison-Burch et al., 2012). The main goals were to provide a baseline approach, devise evaluation metrics, benchmark existing approaches (features and algorithms), and establish the state-of-the-art performance in the area. The task focused on quality prediction at sentence level. Only one dataset was provided, for a single language pair (English-Spanish), on the News domain, translated by one MT system. For training and evaluation, translations were manually annotated by professional translators for quality in terms of "perceived" post-editing effort (1-5 scores) . A system to extract baseline QE features and resources to extract additional features were also provided. The baseline system used a Support Vector Machine regression algorithm trained on the features provided. This was found to be a strong baseline (both features and algorithm) and has been used in all subsequent editions of the task.

As we continued running the task in subsequent years (Bojar et al., 2013; Bojar et al., 2014; Bojar et al., 2015), our main goals have been to provide, each year, new subtasks (while keeping the popular ones), additional language pairs, and larger and more reliably labelled datasets. For most subtasks, the evaluation metrics have also been redefined over the years. Table 4 summarises the main components of the shared task over the years.

More specifically, we introduced variants of post-editing effort prediction – edit distance (a.k.a. HTER) and post-editing time – for sentence level (2013), and other subtasks at new granularity levels: (i) a system selection subtask to learn how to rank alternative MTs for the same source sentence, precisely the same goal as the metrics task (Section 3.), but without reference translations (2013); (ii) a word-level subtask concerned with predicting a binary (good/bad) or 3-way (keep, delete, replace) tag for each word in a target sentence (2013), as well as more fine-grained error categories annotated by humans (omission, word order, word form, etc., in 2014); (iii) a paragraph-level subtask to predict a Meteor score for an entire paragraph (2015); (iv) a document-level subtask to predict a task-based human-targeted score for the entire document (2016); and (v) a phrase-level subtask, where binary labels (good/bad) are to be predicted for entire "phrases", as segmented by the MT system (2016). Baseline systems and resources were provided for all these subtasks.

The main language pair has remained English-Spanish

(en→es), the only constant language over all editions for the sentence and word-level subtasks. This was mostly due to the availability of (labelled) data for this pair. However, other language pairs have been explored over the years for most subtasks. English-German (en→de) was used on various occasions, including all subtasks in 2014 and the paragraph-level subtask in 2015. German-English (de→en) was also used in the latter subtask, in all subtask in 2014, and in the MT system selection task in 2013.

The sizes of the datasets varies over the years. A good indicator is the sentence-level subtask. The figures in the last row of Table 4 refer to the largest number of sentences for any score prediction subtask in a given year.

The number of participating teams has remained considerably stable over the years (10–14), but teams tend to submit systems for various subtasks, as well as for the same subtask when multiple languages are available. The submission figures in Table 4 include only submissions for different subtasks and language pairs.

The evaluation of participating systems varies across subtasks. For sentence, paragraph and document levels, systems can be submitted for two variants of each task: scoring (for various labels, e.g. 1-5, 1-3, HTER, time, Meteor) and ranking, where only a relative ranking of test instances is required. Scoring is evaluated using standard error metrics (e.g. Mean Absolute Error) against the true scores and, since 2015, using Pearson's correlation. Ranking is evaluated using Spearman's correlation, as well as a ranking metric proposed for the task in 2012: DeltaAvg, which compares the ranking of instances given by the system against the human ranking for different quality quantiles of the test set. For the word and phrase-level tasks, per-class precision, recall and F-measure metrics are computed, with F-measure for the "bad" class used as main metric in the binary variant.

Overall, the shared tasks have led to many findings and highlighted various open problems in the field of QE. Here we summarise the most important ones:

- **Training data**: The size of the training data is important for all prediction levels, but is even more critical for word and phrase levels. For sentence level, it does not seem to be the case that having more than 2K sentences makes a significant difference in performance. The quality of the data has proved a more important concern. The dataset used for the sentence and word level subtasks in 2015, for example, although large, was of questionable quality (spurious or missing post-editings) and had a very skewed label distribution, which made model learning harder.

- **Algorithms**: There is no consensus on the best algorithm for each subtask. Various popular regression algorithms have ranked best for sentence (and paragraph) level in different years, including SVM, Multilayer Perceptron, and Gaussian Process. For word (and phrase) level, sequence labelling algorithms such as Conditional Random Fields perform best.

- **Tuning**: Feature selection and hyperparameter optimisation proved essential. The winning submissions in most years performed careful (or even exhaustive) search for both features and hyperparameter values.

- **Features**: While a range of features has been used over the years, shallow, often language-independent features, tend to contribute the most. The majority of submissions built on the set of baseline features provided. Recently, word embeddings and other neural inspired features have been sucessfully explored. While features for sentence and word/phrase-level prediction are clearly very distinct from one another, for paragraph level, most systems used virtually sentence level features. We hope that more interesting discourse features will be exploited in 2016 given the much longer documents provided as instances. A critically important feature for all levels is the *pseudo-reference* score, i.e., comparisons between the MT system output and a translation produced by another MT system for the same input sentence.

- **Labels**: Prediction of objective scores, such as post-editing distance and time, has led to better models (in terms of improvements over the baseline system and correlation with human scores) than prediction of subjective scores such as 1-5 labels. Post-editing time seems to be the most effective label. However, given the natural variance across post-editors, this is only the case when data is collected by and a model is built for a single post-editor.

- **Granularity**: The word-level subtask has proved much more challenging than the sentence-level one, often obtaining very marginal improvements over naive baselines. In the tasks we have run so far, this could have been due to: little training data, limited number of examples of words with errors (class unbalance), and potentially noisy automatic word labelling. We attempted to solve some of these limitations by providing data annotated manually for errors (2014), but for cost reasons the largest dataset we could collect has just over 2K segments. A larger dataset (14K segments) was collected based on post-editions in 2015, but the post-editing, and hence the labelling generated from it, are of questionable quality. In 2016, we are providing an even larger dataset (15K segments) post-edited by professional translators. The new phrase-level subtask in 2016 should also help overcome some of the limitations of the word-level one, by providing more natural ways in which to segment the text for errors. The paragraph-level subtask in 2015 did not attract much attention, perhaps due to the use of an automatic metric as quality label (Meteor). In 2016 we provide actual (much longer) documents labelled by humans.

- **Progress over time**: As with any other shared task, measuring progress over time is a challenge since we have new datasets (and often new training sets) every year. Progress in the QE task can however be speculated in relative terms, more specifically, with respect to the improvement of submitted systems over

	'12	'13	'14	'15	'16
Participating Teams	11	14	10	10	-
Evaluated QE Systems	20	55	57	34	-
Subtasks	1	4			
Sentence Level	•	•	•	•	•
Word Level		•	•	•	•
Paragraph Level				•	
Document Level					•
Phrase Level					•
Language Pairs	en→es	en→es, de→en	en↔de, en↔es	en→es, en↔de	en→es
Largest Dataset (snt)	2,254	2,754	4,416	14,088	15,000

Table 4: Details on different editions of the QE task over the years.

the baseline system. This is possible for the sentence-level subtask, since the language pair and baseline system have remained constant over the years. We have observed, year after year, that more systems are able to beat the baseline, and by a larger margin.

5. Plans for Future Editions

In recent years, we have used Sentence Ranking as the sole method of automatic evaluation (refining it according to certain criticisms (Bojar et al., 2011; Lopez, 2012; Koehn, 2012)), but ongoing problems with reliability, interpretability and poor scalability with increasing numbers of systems have driven the search for alternatives. In 2016, we will pilot a new technique for manual evaluation of MT output. This is based on recent work demonstrating an effective means for collecting adequacy and fluency judgements using crowd-sourcing (Graham et al., 2016). This *Direct Assessment* of machine translation quality is similar to our early attempts to judge quality with adequacy and fluency judgements (Koehn and Monz, 2006; Callison-Burch et al., 2007), but improves upon it in critical ways. Crucially, an analog scale is presented to the user in the form of a slider bar, which underneath maps to a 100-point scale, instead of the 5-point Lickert scale we used in the past, which gave us inconsistent results that were difficult to interpret. Annotators are required to do large batches of assessments in a single sitting, which allows their scores to be normalized more reliably. By embedding deformed outputs and comparing their scores to those of their uncorrupted counterpart, inconsistent, unreliable, and untrustworthy annotators can be identified, and their outputs discarded.

The potential advantages of Direct Assessment are:

- It offers good reliability, as measured by inter-annotator agreement;
- the cost of assessment scales linearly in the number of systems assessed (instead of quadratically, as with Sentence Ranking);
- it provides absolute measures which can be compared year-over-year; and
- the concepts of adequacy and fluency are readily interpretable, in a way that the scores derived from Sentence Ranking are not.

Sentence Ranking will remain our primary evaluation for this year, but the results of this evaluation will be compared to those of the DA evaluation in order to help is assess its suitability for future evaluations.

One of the big issues we face in MT evaluation is the question of *for what purpose?* In other words, the way we evaluate our MT system may depend quite strongly on what we want to use it for, whether for gisting, post-editing, direct publication, language learning, automated information extraction, or something else. The Sentence Ranking method is particularly weak in this regard, since we do not give the raters any guidance as to how they should judge the translations. In some sense, we have punted on the difficult question of purpose, allowing each annotator to be guided by his or her own intuitions. This likely explains some of the low annotator agreement rates. Using adequacy and fluency separately is an improvement as the terms have meaningful interpretation, although they are still intrinsic rather than extrinsic measures. In the end, we believe that the work of the WMT manual evaluation has improved our knowledge for how to assess human quality of MT, providing a rich well from which to draw for those wishing to focus on more targeted and specific applications.

For QE, after the 2016 edition we will have covered all possible granularity levels. The plan is to keep the most popular and the most challenging ones, with a particular emphasis on word and phrase-level prediction. Instead of more language pairs, we will prioritise larger and better datasets for fewer language pairs. Another direction we aim to pursue is better integration with other WMT evaluation tasks, e.g. using the test sets and system translations from the translation task, and reusing the manual evaluations as training data. In the past this has proved difficult logistically because of the tasks' timeframe or unsuccessful because the manual evaluations (esp. rankings) were not adequate for QE. The planned changes in the manual evaluation procedure should make this integration possible.

6. Conclusions

The WMT shared tasks have given us a platform to explore all forms of Machine Translation (MT) evaluation; human evaluation, automatic evaluation with a reference, and quality estimation. Not only that, but WMT has helped to drive research in MT evaluation, firstly by having high profile shared tasks to engage the community; and secondly by the extensive data sets that we provide. Each year, we prepare new translation test sets, and annotated data sets for quality estimation. During the tasks, we collect and release all

translation system submissions, all the human judgements, all the submissions to metrics, and all the quality estimation data. These are made available from the WMT website (for this year it is `www.statmt.org/wmt16`) and are used frequently in subsequent research.

MT evaluation is a hard problem, and is capable of generating significant controversy in the MT community, as we have observed when evaluation results were presented. This difficulty is indicated by the number of changes, experiments, and refinements we have introduced over the years. This year, with the piloting of Direct Assessment, we return to a direct measure of the quality of a system output that we abandoned a number of years ago, and are hopeful that the reformulation of the problem will make DA more successful than our earlier experiments. If so, one option for the QE task in subsequent years is for it to model the prediction of DA scores.

Acknowledgements

This work received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements 645442 (QT21) and 645357 (Cracker).

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., and Ueffing, N. (2004). Confidence Estimation for Machine Translation. In *Proc. of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland.

Bojar, O., Ercegovčević, M., Popel, M., and Zaidan, O. (2011). A Grain of Salt for the WMT Manual Evaluation. In *Proc. of the Sixth Workshop on Statistical Machine Translation*, pages 1–11, Edinburgh, Scotland.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Proc. of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. (2014). Findings of the 2014 Workshop on Statistical Machine Translation. In *Proc. of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 Workshop on Statistical Machine Translation. In *Proc. of the Tenth Workshop on Statistical Machine Translation*, pages 1–46.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (Meta-) Evaluation of Machine Translation. In *Proc. of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic.

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2012). Findings of the 2012 Workshop on Statistical Machine Translation. In *Proc. of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada.

Graham, Y. and Liu, Q. (2016). Achieving Accurate Conclusions in Evaluation of Automatic Machine Translation Metrics. In *Proc. of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (to appear)*, San Diego, CA.

Graham, Y., Baldwin, T., Moffat, A., and Zobel, J. (2016). Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, FirstView:1–28, 1.

Hopkins, M. and May, J. (2013). Models of Translation Competitions. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1416–1424, Sofia, Bulgaria.

Koehn, P. and Monz, C. (2006). Manual and Automatic Evaluation of Machine Translation between European Languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121, New York City.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of MT Summit*.

Koehn, P. (2012). Simulating Human Judgment in Machine Translation Evaluation Campaigns. In *Proc. of IWSLT*, pages 179–184.

Leusch, G. and Ney, H. (2008). BLEUSP, INVWER, CDER: Three improved MT evaluation measures. In *NIST Metrics for Machine Translation Challenge*, Waikiki, Honolulu, Hawaii, October.

Lopez, A. (2012). Putting Human Assessments of Machine Translation Systems in Order. In *Proc. of the Seventh Workshop on Statistical Machine Translation*, pages 1–9, Montréal, Canada.

Macháček, M. and Bojar, O. (2014). Results of the WMT14 Metrics Shared Task. In *Proc. of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, Baltimore, MD, USA.

Macháček, M. and Bojar, O. (2013). Results of the WMT13 Metrics Shared Task. In *Proc. of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria.

Neubig, G. and Watanabe, T. (2016). Optimization for Statistical Machine Translation: A Survey. *Computational Linguistics*, To appear.

Sakaguchi, K., Post, M., and Van Durme, B. (2014). Efficient elicitation of annotations for human evaluation of machine translation. In *Proc. of the Ninth Workshop on Statistical Machine Translation*, pages 1–11, Baltimore, Maryland, USA.

Soricut, R. and Echihabi, A. (2010). TrustRank: Inducing Trust in Automatic Translations via Ranking. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 612–621, Uppsala, Sweden.

Specia, L., Turchi, M., Cancedda, N., Dymetman, M., and Cristianini, N. (2009). Estimating the Sentence-Level Quality of Machine Translation Systems. In *Proc. of the 13th Conference of the European Association for Machine Translation*, pages 28–37, Barcelona, Spain.