No Free Lunch in Factored Phrase-Based Machine Translation *

Aleš Tamchyna and Ondřej Bojar

Institute of Formal and Applied Linguistics,
Malostranské náměstí 25, 11800 Praha
{tamchyna,bojar}@ufal.mff.cuni.cz

Abstract. Factored models have been successfully used in many language pairs to improve translation quality in various aspects. In this work, we analyze this paradigm in an attempt at automating the search for well-performing machine translation systems. We examine the space of possible factored systems, concluding that a fully automatic search for good configurations is not feasible. We demonstrate that even if results of automatic evaluation are available, guiding the search is difficult due to small differences between systems, which are further blurred by randomness in tuning. We describe a heuristic for estimating the complexity of factored models. Finally, we discuss the possibilities of a "semi-automatic" exploration of the space in several directions and evaluate the obtained systems.

1 Introduction

Phrase-based statistical machine translation [1] is probably the most popular approach to MT today. However, its models use no linguistic information for translating—words are treated as mere strings, no internal structure is considered. As such, phrase-based models suffer from certain inherent limitations that some linguistic insight might help to overcome.
Factored models are an extension of phrase-based translation. They were introduced by [2] with the aim to reduce several problems of the paradigm, centered around the inability to handle linguistic description beyond surface forms. In a factored model, the system no longer translates words. Instead, each word is represented by a *vector of factors* that can contain the surface form, but also lemma, word class, morphological characteristics or any other information relevant for translation.
Factored models can employ various types of additional information to improve translation quality on many language pairs in various aspects like morphological coherence [3–8], grammatical coherence [9], compound handling [10] or domain adaptation [11, 12].
In factored translation, decoding is decomposed into a series of mapping steps: translation steps map source factors to target factors, generation steps operate solely on the target side. There are many ways of defining

* This work was supported by the Czech Science Foundation grant P406/11/1499 and the EU project MosesCore (FP7-ICT-2011-7-288487).

a factored system. We can vary the set of source and target factors, but also the mapping steps and the order of their application.
Factored systems are mainly designed based on linguistic intuition, yet there may exist interesting configurations which lack a straightforward linguistic interpretation. The aim of this work is to analyze whether factored systems could be generated *automatically*, i.e. whether we can create an algorithm to decide, given a language pair and possible factors, which configuration will produce the best translations.

2 Factored Phrase-Based Translation

As in phrase-based translation, the main source of data for training a factored model is a parallel corpus. In this case, the corpus can be factored; each word can be annotated with arbitrary linguistic information.
In factored models, translation consists of applying *translation* and *generation* steps that gradually fill in the target-side factors and produce a final translation.
Translation steps (T) operate on phrases, they map a defined subset of source factors to a defined subset of target factors. The translation proceeds similarly as in the phrase-based scenario, it operates on phrases, i.e. contiguous sequences of words regardless of any syntactic structure.
Generation steps (G) operate on the target side, their input is a subset of factors (already generated, e.g. by a previous translation step) and they give at output another subset of target factors. Generation steps operate on single target words, so no word alignment is necessary. In fact, additional monolingual data can be used in their training.
The example in Figure 1 shows a scenario with two translation steps and one generation step. Source lemmas are translated to target lemmas, similarly for tags (translation). The joint information is then used on the target side to generate final surface forms (generation); for each word, the step generates its surface form based on lemma and tag (factors that were filled in by the previous translation steps). Note that factored models used in practice are *synchronous*—the same segmentation into phrases is used for all translation steps.

2.1 Translation Options in Factored Models

Factored models, especially the more complex setups, can dramatically increase the computational cost—the combination of translation options

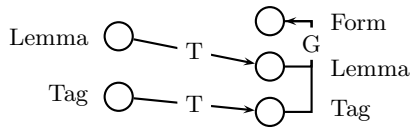

Fig. 1. Factored translation. An example of translation and generation steps.

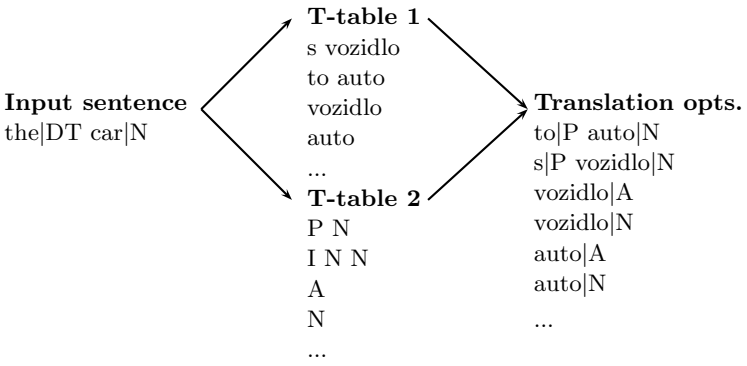

Fig. 2. Phrase expansion in factored models. Options can be used multiple times, such as "DT N"→"N", or completely discarded if they are inconsistent, such as "DT N"→"I N N".

of various steps can cause a combinatorial explosion. Generating all of them is costly in terms of computational time and memory. During decoding, pruning will likely discard good hypotheses, as stacks will be filled with too many factor combinations.

Consider the example shown in Figure 1. This particular translation system uses two translation tables (lemma→lemma, tag→tag) and one generation table (target lemma|tag→form). For each source phrase, the decoder generates all possible translations of the lemmas. Then it combines each lemma with all consistent translations of the tags (resulting in a subset of Cartesian product of the lemma/tag options). Finally, each combination generates zero, one or more (phrases of) target forms. The first two expansions are illustrated in Figure 2.

An expansion is considered *consistent* if the target side has the same length (we are filling in additional factors of a given target phrase) and if the shared factors match.

If the steps share some of the output factors, the order of application of mapping step plays a significant role. In this case, only consistent translation options can be generated during expansion. This restriction has two effects for phrase expansion. First, it limits the number of translation options generated from the existing options. Second, it discards those partial options for which no consistent expansion exists.

For example, suppose that we define two separate translation steps:

1. lemma→lemma
2. tag→lemma

If the steps are applied in this order, the decoder will first generate possible lexical translations. The second step then ensures consistency with the source morphology (e.g. disambiguate between translating English words as nouns or verbs). If we invert the order, the tags will be "translated" first, resulting in an explosion of translation options (the decoder has to produce all lemmas that the source tag can be mapped to).

2.2 Factors

We process our data with Treex,[1] a modular framework for natural language processing. We use tagging and shallow and deep parsing on both sides (English and Czech), enabling us to work with a wide range of linguistic information. Detailed documentation of the discussed factors can be found in PDT[2] and PCEDT[3].

From the morphological layer, we extract the *lemma* and *morphological tag* of each word. Czech lemmas are disambiguated. English tags come from the Penn Treebank tagset [13], Czech tags use the positional system of the Prague Dependency Treebank 2.0 [14]. This tagset is much richer than the English counterpart—about a half of the 4000 possible tags were actually seen in a corpus.

On the surface-syntactic (so-called analytical) layer, words are annotated with their *analytical function*. Examples of analytical functions include `Sb` for subject or `Pred` for predicate.

The tectogrammatical layer describes the deep syntactic structure of sentences. It contains annotation of phenomena that border on the syntax and semantics, such as semantic roles, (grammatical) coreference or valency. We draw a number of factors directly from the annotation:

t-lemma Tectogrammatical lemma, i.e. the deep-syntactic lemma.

functor Describes syntactic-semantic relation of a node to its parent node. Its possible values include ACT (actor), PAT (patient) or ADDR (addressee).

grammatemes A set of factors that describe meaning-bearing morphological properties of t-nodes. We extracted the following categories:

- **gender** Grammatical gender.
- **number** Grammatical number.
- **sempos** Semantic part of speech. This factor classifies autosemnatic words into 4 classes: nouns, adjectives, adverbs and verbs (with their respective subcategories).
- **tense** This attribute specifies the tense of verbs.
- **verbmod** This factor indicates the verb mood.
- **negation** Indicator of negation.

formeme Contains a projection of some morpho-syntactic information from the morphological and analytical layers.

2.3 Software

We use a common set of tools for statistical MT: GIZA++ [15] for computing word alignments, SRILM [16] for creating language models and the Moses toolkit [17] for decoding.

[1] `http://ufal.mff.cuni.cz/treex/`
[2] `http://ufal.mff.cuni.cz/pdt2.0/doc/pdt-guide/en/html/`
[3] `http://ufal.mff.cuni.cz/pcedt2.0/en/`

3 Space of Factored Configurations

In this section, we describe the space of possible factored configurations. A taxonomy of factored systems was proposed by [18]. From this perspective, our work considers Direct (one translation step) and Single-Step (multiple mapping steps within a single search) factored setups.

3.1 Enumeration of Possible Configurations

We can partially order factored setups by the number of mapping steps and explore them in a canonical order (T, TT, TG, TTT,...). Each of these setups can use many combinations of factors and mappings.
Even for one mapping step (this must be a translation step), there are many possible configurations: on the source side, it must use at least one of the lexical factors, but it can also include any number of additional factors, leading to an exponential number of possibilities.[4] The situation on the target side is similar. An exhaustive evaluation is thus intractable even with one translation step.
When multiple mapping steps are involved, the number of configurations explodes further. We analyzed configurations of two mapping steps and the number of factors on each side restricted to 2. Let the first factor (denoted by 0) be the surface form on both sides.
Table 1 shows the viable configurations. For each combination, we provide an example of a potentially good translation system to demonstrate that these combinations warrant exploration. The last column contains our estimate of the number of possible combinations of factored values, given our setting: 12 factors on top of the surface forms, two of which are lexically informative (lemma, tlemma).
We found 13 possible factored scenarios for two mapping steps and estimate that 1142 systems would have to be evaluated if our goal was to explore the space exhaustively. These results demonstrate that an exhaustive search is unrealistic even in this extremely restricted setting. If we hope to find good configurations in this space automatically, we have to guide our search somehow.

4 Evaluation of Factored Configurations

In order to navigate in this space, ideally, we would hope to find a heuristic that would help us predict the translation quality without much computation. But let us back off to a simpler question—can we even reliably compare two factored systems?
The simplest way of evaluating two MT systems is to translate a test set using both of them and compare the achieved BLEU scores [19]. This procedure however disregards the fact that model tuning is randomized. Factored systems can have many parameters (usually 5 for each translation step, 2 for generation steps), adding dimensions to the weight space and thus increasing the effects of randomness.

[4] The number of configurations is proportional to the size of the *power set* of the set of source factors S, i.e. $2^{|S|}$.

Table 1. Enumeration of configurations with two mapping steps.

Mapping Steps		Sample Plausible Setup		Estimated
First	Second	First Mapping Step	Second Mapping Step	Combinations
0→0	1→0	form→form	tag→form	12
0→1	1→0,1	form→POS	lemma→form|POS	48
1→0	0→0	lemma→form	form→form	2
1→0	0→0,1	lemma→form	form→form|tag	24
1→1	0→0,1	tag→tag	form→form|tag	144
1→0,1	0→0	lemma→form|POS	form→form	24
1→0,1	0→1	lemma→form|POS	form→POS	24
0→0,1	1→0	form→form|tag	lemma→form	144
0→0,1	1→1	form→form|tag	tag→tag	144
0,1→0	0→0,1	form|tag→form	form→form|tag	144
0,1→0	1→0,1	form|lemma→form	lemma→form|tag	144
0,1→1	0→0,1	form|tag→lemma	form→form|lemma	144
0,1→1	1→0,1	form|lemma→lemma	lemma→form|lemma	144

Our task also requires us to compare systems which are very close in performance. Can we distinguish the random variance in tuning from a true difference between systems?

We evaluated two algorithms for tuning, minimum error rate training [20] and pairwise-ranked optimization [21]. MERT uses random starting points to avoid reaching local optima. PRO samples its training examples randomly (pairs of translations with high differences in BLEU), but unlike MERT, it is empirically very stable.

In these experiments, we used CzEng 0.9 [22], a richly annotated parallel Czech-English corpus. We trained on a random subset of 200 thousand sentences, development a test data were random 1000-sentence samples from the respective sections of the same corpus.

We used two alternative decoding paths, one that translated form|factor → form and another that only mapped form → form (as a back-off). Each of these paths represents five weights that need to be optimized.

Table 2 shows the evaluated factors. We ran MERT for each factor three times. We can see that differences in BLEU scores in MERT runs are often as high as 0.5 absolute point, which is roughly the same as the improvement we expect from incorporating a useful factor in the system.

Furthermore, if we disregard statistical significance and look simply at the BLEU scores, we might draw very different conclusions depending on which MERT run we consider. We can even entirely invert the ordering of some factors:

- tag (25.07) > functor (25.03) > sempos (25.01) > baseline (24.66)
- baseline (25.16) > sempos (25.01) > functor (24.99) > tag (24.61)

Moreover, if we use just one MERT run and do a statistical significance test, specifically the bootstrap resampling as introduced by [23], the con-

Table 2. BLEU scores achieved by multiple MERT runs and PRO.

Factor	BLEU (3 runs)	Mean	StDev	BLEU-PRO
child(0)→tlemma	24.75, 25.12, 25.43	**25.10**	0.28	24.82
functor	24.99, 25.03, 25.26	**25.09**	0.12	24.56
—	24.66, 25.15, 25.16	24.99	0.23	24.84
formeme	24.58, 25.08, 25.09	24.92	0.24	24.79
sempos	24.75, 25.00, 25.01	24.92	0.12	24.90
tag	24.61, 24.74, 25.07	24.81	0.19	24.90
lemma	24.34, 24.80, 24.88	24.67	0.24	24.81

fidence intervals are so wide that we cannot consider any two systems to be significantly different.[5]
Regarding PRO, our experiments confirmed the stability of the algorithm. However, notice that the order of factors achieved by MERT and PRO is very different. Also, even though MERT is much less stable, it often finds a better set of weights than PRO.
We therefore decided to evaluate all of our experiments by running MERT several (3) times and calculating mean and standard deviation. However we cannot rely on these scores to guide a fully automatic search.

5 Estimating Complexity of Factored Setups

We developed a tool that estimates the number of partial translation options (i.e. translation with factors partially filled in) generated by each step. This estimation is done without decoding and only uses small sample phrase tables. An automatic search for configurations can use this estimate of complexity to prevent training of unrealistic setups. The estimates for individual steps can provide further insights for analysis.
If we estimate the average number of options for a single step, we cannot use the arithmetic mean because extracted phrases obey the power law in a sense: phrases that occur only once have only one translation in the phrase table. These phrases actually make up most of the phrase table but in fact they are almost never used. On the other hand, very frequent phrases tend to have a large number of translations. We therefore use a frequency-weighted average (t_i denotes the number of translations and f_i is the source phrase frequency):

$$avg = \frac{\sum_i f_i \cdot t_i}{\sum_i f_i} \tag{1}$$

When multiple steps are used, the decoder first generates partial options according to the first step and then expands them in the following steps. Each expansion must be consistent. An example of an expansion was shown in Figure 2.

[5] Recently, pair-wise significance tests that sample from multiple runs of the optimizer have been suggested [24].

To approximate this procedure of expansion, we factor each source phrase according to the length of translations and the values of fixed target factors. So each source phrase effectively becomes several source phrases. We then count their translation options separately.
So far we have discussed how to approximate the number of translation options for translation steps. Generation steps are slightly different as generation is done word-by-word. This implies that for a phrase of length k, there will be about avg^k translation options. Instead of k we use the average phrase length according to the first translation table.
When combining the translation and generation steps to obtain an estimate of the number of full translation options, we simply multiply the individual estimates. For each step, we also account for the observed difference in the average number of translation options between tables trained on the full data set and our sample tables (this only needs to be computed once). In our case the ratio was roughly 1.3.
We did not find a way to estimate the effect of *implicit* pruning: for example, we might have a step that translates tag → tag and a following translation step form → form|tag. Some of the previously generated tags will be discarded (if the second step did not generate them) and some of the expansions as well (if their tag was not generated by the previous step). This is the primary source of errors in our estimates, especially for generation steps.

5.1 Evaluation

We evaluated the estimation accuracy for several factored systems. We modified Moses to emit the average number of translation options and compared the results obtained when translating a test set with our prediction. Table 3 shows the results ("t:" and "g:" distinguish translation and generation steps).
As we progress to more complicated setups, the results start to suffer from the deficiency of the heuristic (as discussed above). However, while the absolute values are wrong, the ordering of the setups is correct. This allows us to use the heuristic to pinpoint difficult configurations and the

Table 3. Estimation of the number of translation options per phrase.

Mapping Steps	Estimation	Moses Avg.
t:form→form	$1.3 \cdot 5.38 \doteq$ **7**	**12**
t:tag→tag +	$1.3 \cdot 11.28 \cdot$	
+ t:form→form|tag	$1.3 \cdot 1.28 \doteq$ **24**	**85**
t:lemma→lemma +	$1.3 \cdot 5.23 \cdot$	
+ t:tag→tag +	$1.3 \cdot 57.25 \cdot$	**173**
+ g:lemma|tag→form	$1.3 \cdot 1.13 \doteq$ **655**	
t:lemma→lemma +	$1.3 \cdot 5.19 \cdot$	
+ t:functor→functor +	$1.3 \cdot 52.48 \cdot$	**5153**
+ g:lemma|functor→form	$1.3 \cdot 16.54 \doteq$ **9903**	

problematic steps in them. For example, the last setup (with functors) ran many times longer than the identical configuration with tags (despite the fact that there are far more tags than functors). This difference is correctly discovered by the heuristic.

6 Experiments

In this section, we describe the conducted experiments. Because of the discussed difficulties—the absence of a reliable method for evaluation, the small and insignificant differences in BLEU and the enormous number of possible configurations—we did not carry out a fully automatic exploration of the space of factored setups. Instead, we conducted several sets of experiments in a few targeted research directions; given a small set of factors, a fixed setting and the predictor of setup complexity, we were able to carry out a "semi-automatic" search.

The main source of data for our experiments is CzEng in its latest release 1.0 [25]. It is a richly annotated Czech-English parallel corpus with over 15 million parallel sentences from 7 different domains. We do not use the whole CzEng in the experiments (otherwise the duration of experiments would prohibit *any* search), we limit ourselves to the news domain as the source of both parallel data for translation model training and target-side monolingual data for language modeling.

Our development data (for system tuning) are the test set for WMT11 translation task [26]. For final evaluation of each system, we use WMT test set for 2012. The evaluation data for WMT are news articles, hence the choice of training data. Table 4 shows basic statistics of the data.

6.1 Additional Source Factor

We evaluated the usefulness of all additional factors in combination with the translation of surface forms. The setup was the following:

1. form|*extra* → form
2. (form → form)

All factors were evaluated with and without the alternative path. Results are summarized in Table 5. Baseline system is denoted by '—'. The ± sign denotes the standard deviation over 3 runs of the optimizer. MERT was used for tuning of the systems.

We still see only very little improvements over the baseline BLEU, complicated by variance that makes most of the differences insignificant.

Table 4. Statistics of the data used in experiments.

Data Set	Data Source	Sentences	En Words	Cs Words
Training	CzEng 1.0 news	197053	4641026	4193078
Development	WMT11 test set	3003	74822	65602
Test	WMT12 test set	3003	72955	65306

Table 5. BLEU scores of configurations with 1 translation step.

Factor	Single Path	+Alternative
—	9.93±0.03	—
afun	**10.08±0.08**	**10.11±0.09**
formeme	2.41±0.01	9.95±0.02
functor	9.08±0.08	**10.07±0.08**
gender	9.70±0.05	9.87±0.06
lemma	9.93±0.08	9.66±0.30
negation	**10.05±0.03**	9.99±0.02
number	10.00±0.03	9.96±0.08
person	9.92±0.03	9.79±0.18
sempos	9.93±0.06	9.95±0.16
tag	10.00±0.07	9.95±0.11
tense	**10.06±0.05**	**10.05±0.06**
tlemma	8.62±0.06	9.99±0.15
verbmod	9.56±0.04	9.94±0.10

Even so, several factors stand out in both scenarios as potentially valuable for modeling the English-Czech translation.

In the first column, factors that lead to data sparsity were penalized due to the absence of a back-off. Formeme stands out as the most prominent example, with the BLEU score 2.41 and almost no deviation; all MERT runs converged in a few iterations. Adding this factor diluted the data so much that translation became impossible. Factors that achieved high scores in this column can be (relatively) safely added to translation systems: they do not make the data much sparser and increase translation quality. The best factors are highlighted: analytical function, negation, tense. Grammatical number and tag are also potentially useful.

Analytical function provides roles of English words (subject, predicate etc.) which help disambiguate target-side morphology—in Czech, subjects are almost always in nominative case while objects frequently appear in accusative or dative case.

Tense helped disambiguate verb forms mainly when the predicate contained an auxilliary verb specifying future or past tense. Our annotation assigns this tense also to the main verb (e.g. "will|post go|post") making its translation easier even when it is translated independently (as a one-word phrase).

We suspect that the benefit of the negation attribute is more due to the annotation rules—nouns are (almost always) assigned an empty value, while verbs, adjectives and adverbs are assigned either "neg0" or "neg1". Thus the negation attribute provides a coarse-level PoS tagging useful for modelling the overall sentence structure.

In the second column, even factors that introduce some degree of data sparsity can achieve high scores—they may help in modeling some rare but difficult phenomena. In the situations where the additional information is not helpful, the alternative path maintains good quality of

translation. Functor, analytical function and tense appear to be the most promising factors according to this column.
We used the results to create a combination of factors that we then evaluated separately. As it is not clear which back-offs should be used when multiple factors are combined, we evaluated several approaches; the results are summarized in Table 6 and demonstrate quite clearly that the simplest back-off (just translating surface forms) works best—the overall BLEU score is the highest and this setup was also the most stable one.

6.2 Multiple Mapping Steps

In this section, we evaluated a typical factored scenario with several factors. The scenario consists of two consecutive translation steps: lemma → lemma and one additional factor to its counterpart. This is followed by a generation step that takes the lemma and the additional factor and generates surface form on the target side. All of the factors have a language model on the target side. An alternative path maps surface form directly to all three target factors.
This setup has been used with tags in the past and improvements have been reported on similarly small datasets. Our results are shown in Table 7. Systems without a score ran for too long (one MERT iteration took over a day); this was correctly predicted by our complexity heuristic.
We achieved a large gain in BLEU (roughly 1.1 point absolute) when we used morphological tag as the additional factor, which confirms previous findings. However, no other factor was beneficial in this scenario.

7 Discussion

7.1 Experimental Results

We were able to improve translation performance (0.3 BLEU absolute) when using a single translation step by combining well-performing factors on the source side. We showed that analytical function, tense and

Table 6. Back-off strategies and achieved BLEU scores.

Translation Steps	BLEU
form|afun → form : : form|functor → form : : form|tense → form	10.00±0.29
form|afun|functor|tense → form : : form|afun → form	10.08±0.10
form|afun|functor → form : : form|tense → form	10.10±0.08
form|afun|functor|tense → form : : form → form	**10.24±0.02**

Table 7. BLEU scores of systems with 2 translation and 1 generation steps.

Factor	BLEU	Prediction of Complexity
formeme	9.91±0.05	4573
—	9.93±0.03	7
tag	**11.05±0.03**	655
functor	—	9903
sempos	—	38412
tense	—	13607

functors as used in the PCEDT annotation are the most useful from a wide range of attributes for modeling factored phrase-based transfer of English into Czech.

We also evaluated a scenario that consists of multiple mapping steps. Unfortunately, similarly to the previous set of experiments, we were unable to identify any new useful factors, so even though our improvement in BLEU score is quite large (over 1.1 points), our findings are not new.

7.2 Search for Factored Configurations

It seems that finding the correct combination of steps and factors is not a task that an algorithm can solve, especially not by brute force—the number of possibilities explodes no matter which direction of exploration we take. A clever automatic search in the space of configurations does not seem feasible due to the low reliability of automatic MT evaluation and frequent large variance in scores across different optimization runs.

We believe it is possible to search for factored configurations semi-automatically given a particular research goal—the methods and tools that we developed can assist in selecting the most suitable factored setup from a limited number of possibilities.

8 Conclusion

We provided an analysis of the paradigm of factored machine translation. We described the complexity of the space of configurations. We proposed a heuristic that can successfully predict which factored setups are too complex to be feasible. We carried out a "semi-automatic" search for factored configurations in several directions and evaluated the results.

In the future, we would like to apply the developed machinery to more complex setups and richer sets of factors but obviously with a manual guidance. We would also like to improve the precision of the heuristic for complexity estimation.

References

1. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: HLT/NAACL. (2003)
2. Koehn, P., Hoang, H.: Factored translation models. In: EMNLP-CoNLL, ACL (2007) 868–876
3. Bojar, O.: English-to-Czech Factored Machine Translation. In: Proc. of ACL WMT, Prague, Czech Republic, ACL (2007) 232–239
4. Avramidis, E., Koehn, P.: Enriching morphologically poor languages for statistical machine translation. In: Proc. of ACL/HLT, Columbus, Ohio, ACL (2008) 763–770
5. Badr, I., Zbib, R., Glass, J.: Segmentation for English-to-Arabic statistical machine translation. In: Proc. of ACL/HLT Short Papers, Columbus, Ohio, ACL (2008) 153–156
6. Ramanathan, A., Choudhary, H., Ghosh, A., Bhattacharyya, P.: Case markers and morphology: addressing the crux of the fluency problem in English-Hindi SMT. In: Proc. of ACL/IJCNLP: Volume 2, Suntec, Singapore, ACL (2009) 800–808
7. Koehn, P., Haddow, B., Williams, P., Hoang, H.: More linguistic annotation for statistical machine translation. In: Proc. of WMT and MetricsMATR, Uppsala, Sweden, ACL (2010) 115–120
8. Yeniterzi, R., Oflazer, K.: Syntax-to-Morphology Mapping in Factored Phrase-Based Statistical Machine Translation from English to Turkish. In: Proc. of ACL, Uppsala, Sweden, ACL (2010) 454–464
9. Birch, A., Osborne, M., Koehn, P.: CCG Supertags in Factored Statistical Machine Translation. In: Proc. of ACL WMT, Prague, Czech Republic, ACL (2007) 9–16
10. Stymne, S.: German Compounds in Factored Statistical Machine Translation. In Nordström, B., Ranta, A., eds.: Advances in Natural Language Processing. Volume 5221 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2008) 464–475
11. Koehn, P., Schroeder, J.: Experiments in domain adaptation for statistical machine translation. In: Proc. of ACL WMT, Prague, Czech Republic, ACL (2007) 224–227
12. Niehues, J., Waibel, A.: Domain adaptation in statistical machine translation using factored translation models. In: EAMT. (2010)
13. Santorini, B.: Part-of-Speech Tagging Guidelines for the Penn Treebank Project. University of Pennsylvania, School of Engineering and Applied Science, Dept. of Computer and Information Science, Philadelphia (1990)
14. Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Z., Ševčíková Razímová, M.: Prague Dependency Treebank 2.0. LDC2006T01, ISBN: 1-58563-370-4 (2006)
15. Och, F.J., Ney, H.: Improved statistical alignment models. In: ACL, ACL (2000)
16. Stolcke, A.: SRILM - an extensible language modeling toolkit. In: Proc. of ICSLP2002 - INTERSPEECH, Denver, Colorado, USA, ISCA (2002)

17. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Proc. of ACL: Demo and Poster Sessions, Prague, Czech Republic, ACL (June 2007) 177–180
18. Bojar, O., Jawaid, B., Kamran, A.: Probes in a Taxonomy of Factored Phrase-Based Models. In: Proc. of ACL WMT, Montréal, Canada, ACL (2012) 253–260
19. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: A method for automatic evaluation of machine translation. In: ACL, ACL (2002) 311–318
20. Och, F.J.: Minimum error rate training in statistical machine translation. In: Proc. of ACL, Sapporo, Japan, ACL (2003) 160–167
21. Hopkins, M., May, J.: Tuning as ranking. In: EMNLP, ACL (2011) 1352–1362
22. Bojar, O., Žabokrtský, Z.: CzEng0.9: Large Parallel Treebank with Rich Annotation. Prague Bulletin of Mathematical Linguistics **92** (2009)
23. Koehn, P.: Statistical Significance Tests for Machine Translation Evaluation. In: Proc. of EMNLP, Barcelona, Spain (2004)
24. Clark, J.H., Dyer, C., Lavie, A., Smith, N.A.: Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In: Proc. of ACL (Short Papers), ACL (2011) 176–181
25. Bojar, O., Žabokrtský, Z., Dušek, O., Galuščáková, P., Majliš, M., Mareček, D., Maršík, J., Novák, M., Popel, M., Tamchyna, A.: The Joy of Parallelism with CzEng 1.0. In: Proc. of LREC, İstanbul, Turkey, ELRA (2012) 3921–3928
26. Callison-Burch, C., Koehn, P., Monz, C., Zaidan, O.: Findings of the 2011 Workshop on Statistical Machine Translation. In: Proc. of ACL WMT, Edinburgh, Scotland, ACL (2011) 22–64