

Robust Multilingual Statistical Morphological Generation Models

Ondřej Dušek and Filip Jurčiček

Charles University in Prague, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, CZ-11800 Praha, Czech Republic
{odusek, jurcicek}@ufal.mff.cuni.cz

Abstract

We present a novel method of statistical morphological generation, i.e. the prediction of inflected word forms given lemma, part-of-speech and morphological features, aimed at robustness to unseen inputs. Our system uses a trainable classifier to predict “edit scripts” that are then used to transform lemmas into inflected word forms. Suffixes of lemmas are included as features to achieve robustness. We evaluate our system on 6 languages with a varying degree of morphological richness. The results show that the system is able to learn most morphological phenomena and generalize to unseen inputs, producing significantly better results than a dictionary-based baseline.

1 Introduction

Surface realization is an integral part of all natural language generation (NLG) systems, albeit often implemented in a very simple manner, such as filling words into ready hand-written templates. More sophisticated methods use hand-written grammars (Gatt and Reiter, 2009), possibly in combination with a statistical reranker (Langkilde and Knight, 1998). Existing NLG systems are very often applied to languages with little morphology, such as English, where a small set of hand-written rules or the direct use of word forms in the symbolic representation or templates is usually sufficient, and so the main focus of these systems lies on syntax and word order.

However, this approach poses a problem in languages with a complex morphology. Avoiding inflection, i.e. ensuring that a word lemma will keep its base form at all times, often leads to very unnatural results (see Figure 1). Some generators use a hand-made morphological dictionary

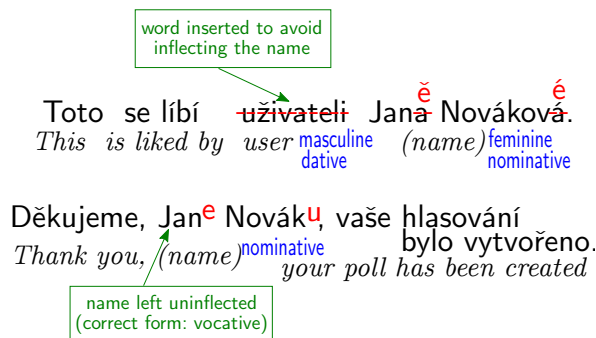


Figure 1: Unnatural language resulting from templates with no inflection.

The sentences are taken from the Czech translations of Facebook and Doodle, which use simple templates to generate personalized texts. Corrections to make the text fluent are shown in red.

for inflection (Ptáček and Žabokrtský, 2006) or a dictionary learned from automatically tagged data (Toutanova et al., 2008). That gives good results, but reaching sufficient coverage with a hand-made dictionary is a very demanding task and even using extreme amounts of automatically annotated data will not generalize beyond the word forms already encountered in the corpus. Hand-written rules can become overly complex and are not easily adaptable for a different language.

Therefore, the presented method relies on a statistical approach that learns to predict morphological inflection from annotated data. As a result, such approach is more robust, i.e. capable of generalizing to unseen inputs, and easily portable to different languages.

An attempt to implement statistical morphological generation has already been made by Bohnet et al. (2010). However, their morphology generation was only a component of a complex generation system. Therefore, no deep analysis of the capabilities of the methods has been performed. In addition, their method did not attempt to generalize beyond seen inputs. In this paper, we propose

several improvements and provide a detailed evaluation of a statistical morphological inflection system, including more languages into the evaluation and focusing on robustness to unseen inputs.

The paper is structured as follows: first, we explain the problem of morphological generation (Section 2), then give an account of our system (Section 3). Section 4 provides a detailed evaluation of the performance of our system in different languages. We then compare our system to related works in Section 5. Section 6 concludes the paper.

2 The Problem of Morphological Realization

The problem of morphological surface realization is inverse to part-of-speech tagging and lemmatization (or stemming): given a lemma/stem of a word and its part-of-speech and morphological properties, the system should output the correctly inflected form of the word. An example is given in Figure 2. This does not include generating auxiliary words (such as *be* → *will be*), which are assumed to be already generated.

word	+	NNS	→	words
Wort	+	NN Neut,Pl,Dat	→	Wörtern
be	+	VBZ	→	is
ser	+	V ^{gen=c,num=s,person=3,mood=indicative,tense=present}	→	es

Figure 2: The task of morphological generation (examples for English, German, and Spanish).

While this problem can be solved by a set of rules to a great extent for languages with little morphology such as English (Minnen et al., 2001), it becomes much more complicated in languages with a complex nominal case system or multiple synthetic verbal inflection patterns, such as German or Czech. Figure 3 shows an example of ambiguity in these languages.

This research aims to create a system that is easy to train from morphologically annotated data, yet able to infer and apply more general rules and generate forms unseen in the training corpus.

3 Our Morphological Generation Setup

Similarly to Bohnet et al. (2010), our system is based on the prediction of edit scripts (diffs) between the lemma and the target word form (see Section 3.1), which are then used to derive the target word form from the lemma. This allows the

Teller (plate)	NN	Gen=Sg,Masc	Tellers	Herr (sir)	NN	Nom,Pl,Masc	Herren
Oma (grandma)	NN	Nom,Pl,Fem	Omas	Mann (man)	NN	Nom,Pl,Masc	Männer
stroj (machine)	N	Gen=I,Cas=7,Num=S	strojem	pán (sir)	N	Gen=M,Cas=2,Num=S	pána
kost (bone)	N	Gen=F,Cas=3,Num=P	kostem	kůň (horse)	N	Gen=M,Cas=2,Num=S	kone

Figure 3: Morphological ambiguity in German and Czech.

The same inflection pattern is used to express multiple morphological properties (left) and multiple patterns may express the same property (right).

system to operate even on previously unseen lemmas. The employed classifier and features are described in Sections 3.2 and 3.3. Section 3.4 then gives an overview of the whole morphological inflection process.

3.1 Lemma-Form Edit Scripts

Our system uses lemma-form edit scripts based on the Levenshtein string distance metric (Levenshtein, 1966): the dynamic programming algorithm used to compute the distance can be adapted to produce diffs on characters, i.e. a mapping from the source string (lemma) to the target string (word form) that indicates which characters were added, replaced or removed.

We use the distance from the end of the word to indicate the position of a particular change, same as Bohnet et al. (2010). We have added several enhancements to this general scenario:

- Our system treats separately changes at the beginning of the word, since they are usually independent of the word length and always occur at the beginning, such as the prefix *ge-* for past participles in German or *ne-* for negation in Czech.
- Adjacent changes in the string are joined together to produce a total lower number of more complex changes.
- If the Levenshtein edit script indicates a removal of letters from the beginning of the word, we treat the target word form as irregular, i.e. as if the whole word changed.
- In our setup, the edit scripts need not be treated as atomic, which allows to train separate classification models for word changes that are orthogonal (cf. Section 3.4).

An example of the edit scripts generated by our system is shown in Figure 4.

do	doing	>0- <i>ing</i>
llegar	llegó	>2- <i>ó</i>
Mann	Männer	>0- <i>er,3:1-ä</i>
jenž	jež	>2:1-
mantenir	mantindran	>0- <i>an,2:1-d,4:1-i</i>
sparen	gespart	>2- <i>t,<ge</i>
vědět	nevíme	>4- <i>íme,<ne</i>
be	is	* <i>is</i>

Figure 4: Example edit scripts generated by our system.

The changes are separated by commas. “>” denotes a change at the end of the word, “N:” denotes a change at the N -th character from the end. The number of deleted characters and their replacement follows in both cases. “<” marks additions to the beginning of a word (regardless of its length). “*” marks irregular forms where the whole word is replaced.

Our diffs are case-insensitive since we believe that letter-casing and morphology are distinct phenomena and should be treated separately. Case-insensitivity, along with merging adjacent changes and the possibility to split models, causes a decrease in the number of different edit scripts, thus simplifying the task for the classifier.

In our preliminary experiments on Czech, we also explored the possibility of using different distance metrics for the edit scripts, such as various settings of the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) or the longest common subsequence¹ post-edited with regular expressions to lower the total number of changes. However, this did not have any noticeable impact on the performance of the models.

3.2 Used Statistical Models

We use the multi-class logistic regression classifier from the LibLINEAR package² (Fan et al., 2008) for the prediction of edit scripts. We use L1-regularization since it yields models that are smaller in size and the resulting trained weights indicate the important features in a straightforward way. This direct influence on features (similar to keyword spotting) allows for a simple interpretation of the learned models. We examined various settings of the regularization cost and the termination criterion (See Section 4.1).

We have also experimented with support vector machines from the LibSVM package (Chang

¹We used the Perl implementation of this algorithm from <https://metacpan.org/module/String::Diff>.

²We use it via the Python wrapper in the Scikit-Learn library (<http://scikit-learn.org>).

and Lin, 2011), but the logistic regression classifier proved to be better suited to this task, providing a higher edit script accuracy on the development set for German and Czech (when feature concatenation is used, cf. Section 3.3), while also requiring less CPU time and RAM to train.

3.3 Features

While the complete set of features varies across languages given their specifics, most of the features are common to all languages:

- *lemma* of the word in question,
- coarse and fine-grained *part-of-speech* tag,
- *morphological features* (e.g. case, gender, tense etc., tagset-dependent), and
- *suffixes of the lemma* of up to 4 characters.

Since morphological changes usually occur near the end of the word, they mostly depend just on that part of the word and not on e.g. prefixes or previous parts of a compound. Therefore, using suffixes allows the classifier to generalize to unknown words.

In addition, as we use a linear classifier, we have found the concatenation of various morphological features, such as number, gender, and case in nouns or tense and person in verbs, to be very beneficial. We created new features by concatenating all possible subsets of morphological features, as long as all their values were non-empty (to prevent from creating duplicate values). To avoid combinatorial explosion, we resorted to concatenating only case, number, and gender for Czech and excluding the `postype` feature from concatenation for Spanish and Catalan.

We also employ the properties of adjacent words in the sentence as features in our models for the individual languages (see Section 4). These are used mainly to model congruency (*is* vs. *are* in English, different adjectival declension after definite and indefinite article in German) or article vocalization (*l'* vs. *el* in Catalan). The congruency information could be obtained more reliably from elsewhere in a complete NLG system (e.g. features from the syntactic realizer), which would probably result in a performance gain, but lies beyond the scope of this paper.

No feature pruning was needed in our setup as our classifier was able to handle the large amount of features (100,000s, language-dependent).

3.4 Overall Schema of the Predictor

After an examination of the training data, we decided to use a separate model for the changes that occur at the beginning of the word since they tend to be much simpler than and not very dependent on the changes towards the end of the word (e.g. the usages of the Czech negation prefix *ne-* or the German infinitive prefix *zu-* are quite self-contained phenomena).

The final word inflection prediction schema looks as follows:

1. Using the statistical model described in Section 3.2, predict an edit script (cf. Section 3.1) for changes at the end or in the middle of the word.³
2. Predict an edit script for the possible addition of a prefix using a separate model.
3. Apply the edit scripts predicted by the previous steps as rules to generate the final inflected word form.

4 Experimental Evaluation

We evaluate our morphological generation setup on all of the languages included in the CoNLL 2009 Shared Task data sets except Chinese (which, as an isolating language, lacks morphology almost altogether): English, German, Spanish, Catalan, Japanese, and Czech. We use the CoNLL 2009 data sets (Hajič et al., 2009) with gold-standard morphology annotation for all our experiments (see Table 1 for a detailed overview).

We give a discussion of the overall performance of our system in all the languages in Section 4.1. We focus on Czech in the detailed analysis of the generalization power of our system in Section 4.2 since Czech has the most complicated morphology of all these languages. In addition, the morphological annotation provided in the CoNLL 2009 Czech data set is more detailed than in the other languages, which eliminates the need for additional syntactic features (cf. Section 3.3). We also provide a detailed performance overview on English for comparison.

4.1 Overall Performance

The performance of our system in the best settings for the individual languages measured on the

³Completely irregular forms (see Section 3.1) are also predicted by this step.

CoNLL 2009 evaluation test sets is shown in Table 2. We used the classifier and features described in Sections 3.2 and 3.3 (additional features for the individual languages are listed in the table). We used two models as described in Section 3.4 for all languages but English, where no changes at the beginning of the word were found in the training data set and a single model was sufficient. We performed a grid search for the best parameters of the first model⁴ and used the same parameters for both models.⁵

One can see from the results in Table 2 that the system is able to predict the majority of word forms correctly and performs well even on data unseen in the training set.

When manually inspecting the errors produced by the system, we observed that in some cases the system in fact assigned a form synonymous to the one actually occurring in the test set, such as *not* instead of *n't* in English or *také* instead of *taky* (both meaning *also*) in Czech. However, most errors are caused by the selection of a more frequent rule, even if incorrect given the actual morphological features. We believe that this could possibly be mitigated by using features combining lemma suffixes and morphological categories, or features from the syntactic context.

The lower score for German is caused partly by the lack of syntactic features for the highly ambiguous adjective inflection and partly by a somewhat problematic lemmatization of punctuation (all punctuation has the lemma “_” and the part-of-speech tag only distinguishes terminal, comma-like and other characters).

4.2 Generalization Power

To measure the ability of our system to generalize to previously unseen inputs, we compare it against a baseline that uses a dictionary collected from the same data and leaves unseen forms intact. The performance of our system on unseen forms is shown in Table 2 for all languages. A comparison with the dictionary baseline for varying training data sizes in English and Czech is given in Table 3.

It is visible from Table 3 that our approach

⁴We always used L1-norm and primal form and modified the termination criterion *tol* and regularization strength *C*. The best values found on the development data sets for the individual languages are listed in Table 2.

⁵As the changes at the beginning of words are much simpler, changing parameters did not have a significant influence on the performance of the second model.

Language	Data set sizes			In Eval (%)		
	Train	Dev	Eval	-Punct	InflF	UnkF
English	958,167	33,368	57,676	85.93	15.14	1.80
German	648,677	32,033	31,622	87.24	45.12	8.69
Spanish	427,442	50,368	50,630	85.42	29.96	6.16
Catalan	390,302	53,015	53,355	86.75	31.89	6.28
Japanese	112,555	6,589	13,615	87.34	10.73	6.43
Czech	652,544	87,988	92,663	85.50	42.98	7.68

Table 1: The CoNLL 2009 data sets: Sizes and properties

The data set sizes give the number of words (tokens) in the individual sets. The right column shows the percentage of data in the evaluation set: *-Punct* = excluding punctuation tokens, *InflF* = only forms that differ from the lemma (i.e. have a non-empty edit script), *UnkF* = forms unseen in the training set.

Language	Additional features	Best parameters	Rule (%) accuracy	Form accuracy (%)			
				Total	-Punct	InflF	UnkF
English	W-1/LT	C=10, tol=1e-3	99.56	99.56	99.49	97.76	98.26
German	W-1/LT, MC	C=10, tol=1e-3	96.66 / 99.91	96.46	98.01	92.64	89.63
Spanish	MC	C=100, tol=1e-3	99.05 / 99.98	99.01	98.86	97.10	91.11
Catalan	W+1/C1, MC	C=10, tol=1e-3	98.91 / 99.86	98.72	98.53	96.49	94.24
Japanese	MC	C=100, tol=1e-3	99.94 / 100.0	99.94	99.93	99.59	99.54
Czech	MC	C=100, tol=1e-3	99.45 / 99.99	99.45	99.35	98.81	95.93

Table 2: The overall performance of our system in different languages.

The *additional features* include: *MC* = concatenation of morphological features (see Section 3.3), *W-1/LT* = lemma and part-of-speech tag of the previous word, *W+1/C1* = first character of the following word.

Rule (edit script) accuracy is given for the prediction of changes at the end or in the middle and at the beginning of the word, respectively.

The *form accuracy* field shows the percentage of correctly predicted (lowercased) target word forms: *Total* = on the whole evaluation set; *-Punct*, *InflF*, *UnkF* = on subsets as defined in Table 1.

maintains a significantly⁶ higher accuracy when compared to the baseline for all training data sizes. It is capable of reaching high performance even with relatively small amounts of training instances. The overall performance difference becomes smaller as the training data grow; however, performance on unseen inputs and relative error reduction show a different trend: the improvement stays stable. The relative error reduction decreases slightly for English where unknown word forms are more likely to be base forms of unknown lemmas, but keeps increasing for Czech where unknown word forms are more likely to require inflection (the accuracy reached by the baseline method on unknown forms equals the percentage of base forms among the unknown forms).

Though the number of unseen word forms is declining with increasing amounts of training data, which plays in favor of the dictionary method, unseen inputs will still occur and may become very frequent for out-of-domain data. Our system is therefore beneficial – at least as a back-off for unseen forms – even if a large-coverage morpholog-

ical dictionary is available.

We observed upon manual inspection that the suffix features were among the most prominent for the prediction of many edit scripts, which indicates their usefulness; e.g. `LemmaSuffix1=e` is a strong feature (along with `POS_Tag=VBD`) for the edit script `>0d` in English.

5 Related Work

Statistical morphological realizers are very rare since most NLG systems are either fully based on hand-written grammars, including morphological rules (Bateman et al., 2005; Gatt and Reiter, 2009; Lavoie and Rambow, 1997), or employ statistical methods only as a post-processing step to select the best one of several variants generated by a rule-based system (Langkilde and Knight, 1998; Langkilde-Geary, 2002) or to guide the decision among the rules during the generation process (Belz, 2008). While there are fully statistical surface realizers (Angeli et al., 2010; Mairesse et al., 2010), they operate in a phrase-based fashion on word forms with no treatment of morphology. Morphological generation in machine translation tends to use dictionaries – hand-written (Žabokrt-

⁶Significance at the 99% level has been assessed using paired bootstrap resampling (Koehn, 2004).

Train data part	Czech						English					
	Unseen forms	Dict. acc.		Our sys. acc.		Error reduct.	Unseen forms	Dict acc.		Our sys. acc.		Error reduct.
		Total	UnkF	Total	UnkF			Total	UnkF			
0.1	63.94	62.00	41.54	76.92	64.43	39.27	27.77	89.18	78.73	95.02	93.14	53.91
0.5	51.38	66.78	38.65	88.73	78.83	66.08	19.96	91.34	76.33	97.89	95.56	75.64
1	45.36	69.43	36.97	92.23	83.60	74.60	14.69	92.76	73.95	98.28	95.27	76.19
5	31.11	77.29	35.56	96.63	90.36	85.17	6.82	96.21	75.73	99.05	97.13	74.96
10	24.72	80.97	33.88	97.83	92.45	88.61	4.66	97.31	77.13	99.34	97.76	75.44
20	17.35	85.69	32.47	98.72	94.28	91.02	3.10	98.09	78.52	99.46	97.57	71.65
30	14.17	87.92	31.85	98.95	94.56	91.34	2.46	98.40	79.79	99.48	97.63	67.75
50	11.06	90.34	31.62	99.20	95.25	91.69	1.76	98.69	80.53	99.54	98.04	64.81
75	9.01	91.91	31.54	99.34	95.60	91.89	1.35	98.86	82.23	99.55	98.17	60.61
100	7.68	92.88	30.38	99.45	95.93	92.21	1.12	98.94	82.53	99.56	98.26	58.85

Table 3: Comparison of our system with a dictionary baseline on different training data sizes. All numbers are percentages. The accuracy of both methods is given for the whole evaluation set (*Total*) and for word forms unseen in the training set (*UnkF*). *Error reduct.* shows the relative error reduction of our method in comparison to the baseline on the whole evaluation set.

ský et al., 2008), learnt from data (Toutanova et al., 2008), or a combination thereof (Popel and Žabokrtský, 2009).

The only statistical morphological generator known to us is that of Bohnet et al. (2010), employed as a part of a support-vector-machines-based surface realizer from semantic structures. They apply their system to a subset of CoNLL 2009 data sets and their results (morphological accuracy of 97.8% for English, 97.49% for German and 98.48% for Spanish) seem to indicate that our system performs better for English, slightly better for Spanish and slightly worse for German, but the numbers may not be directly comparable to our results as it is unclear whether the authors use the original data set or the output of the previous steps of their system for evaluation and whether they include punctuation and/or capitalization.

Since the morphological generator of Bohnet et al. (2010) is only a part of a larger system, they do not provide a thorough analysis of the results. While their system also predicts edit scripts derived from Levenshtein distance, their edit script representation seems less efficient than ours. They report using about 1500 and 2500 different scripts for English and German, respectively, disregarding scripts occurring only once in the training data. However, our representation only yields 154 English and 1785 German⁷ edit scripts with no pruning. Along with the independent models for the beginning of the word, this simplifies the task for the classifier. In addition to features used by

⁷We get this number when counting the edit scripts as atomic; they divide into 1735 changes at the end or in the middle of the words and 18 changes at the beginning.

Bohnet et al. (2010), our system includes the suffix features to generalize to unseen inputs.

6 Conclusions and Further Work

We have presented a fully trainable morphological generation system aimed at robustness to previously unseen inputs, based on logistic regression and Levenshtein distance edit scripts between the lemma and the target word form. The results from the evaluation on six different languages from the CoNLL 2009 data sets indicate that the system is able to learn most morphological rules correctly and is able to cope with previously unseen input, performing significantly better than a dictionary learned from the same amount of data. The system is freely available for download at:

<http://ufal.mff.cuni.cz/~odusek/flect>

In future, we plan to integrate our generator into a semantic NLG scenario, as well as a simpler template-based system, and evaluate it on further languages. We also consider employing transformation-based learning (Brill, 1995) for prediction to make better use of the possibility of splitting the edit scripts and applying the morphological changes one-by-one.

Acknowledgments

This research was partly funded by the Ministry of Education, Youth and Sports of the Czech Republic under the grant agreement LK11221 and core research funding of Charles University in Prague. The authors would like to thank Matěj Korvas and Martin Popel for helpful comments on the draft and David Marek, Ondřej Plátek and Lukáš Žilka for discussions.

References

- G. Angeli, P. Liang, and D. Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, page 502–512.
- J. A. Bateman, I. Kruijff-Korbayová, and G.-J. Kruijff. 2005. Multilingual resource sharing across both related and unrelated languages: An implemented, open-source framework for practical natural language generation. *Research on Language and Computation*, 3(2-3):191–219.
- A. Belz. 2008. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455.
- B. Bohnet, L. Wanner, S. Mille, and A. Burga. 2010. Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. In *Proceedings of the 23rd International Conference on Computational Linguistics*, page 98–106.
- E. Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational linguistics*, 21(4):543–565.
- C. C. Chang and C. J. Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.
- R. E Fan, K. W Chang, C. J Hsieh, X. R Wang, and C. J Lin. 2008. LIBLINEAR: a library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874.
- A. Gatt and E. Reiter. 2009. SimpleNLG: a realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, page 90–93.
- J. Hajič, M. Ciaramita, R. Johansson, D. Kawahara, M. A Martí, L. Màrquez, A. Meyers, J. Nivre, S. Padó, J. Štěpánek, et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, page 1–18.
- P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, page 388–395.
- I. Langkilde and K. Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, page 704–710.
- I. Langkilde-Geary. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings of the 12th International Natural Language Generation Workshop*, page 17–24.
- B. Lavoie and O. Rambow. 1997. A fast and portable realizer for text generation systems. In *Proceedings of the fifth conference on Applied natural language processing*, page 265–268.
- V. I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707.
- F. Mairesse, M. Gašić, F. Jurčiček, S. Keizer, B. Thomson, K. Yu, and S. Young. 2010. Phrase-based statistical language generation using graphical models and active learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, page 1552–1561.
- G. Minnen, J. Carroll, and D. Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.
- S. B. Needleman and C. D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.
- M. Popel and Z. Žabokrtský. 2009. Improving English-Czech tectogrammatical MT. *The Prague Bulletin of Mathematical Linguistics*, 92(1):115–134.
- J. Ptáček and Z. Žabokrtský. 2006. Synthesis of Czech sentences from tectogrammatical trees. In *Text, Speech and Dialogue*.
- K. Toutanova, H. Suzuki, and A. Ruopp. 2008. Applying morphology generation models to machine translation. In *Proc. of ACL*, volume 8.
- Z. Žabokrtský, J. Ptáček, and P. Pajas. 2008. TectoMT: highly modular MT system with tectogrammatcs used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, page 167–170. Association for Computational Linguistics.