PBML

The Prague Bulletin of Mathematical Linguistics

NUMBER 101 APRIL 2014 29–41

Integrating a Discriminative Classifier into Phrase-based and Hierarchical Decoding

Aleš Tamchyna[a], Fabienne Braune[b], Alexander Fraser[b], Marine Carpuat[c], Hal Daumé III[d], Chris Quirk[e]

[a] Charles University in Prague
[b] Ludwig Maximilians University Munich
[c] National Research Council Canada
[d] University of Maryland
[e] Microsoft Research

Abstract

Current state-of-the-art statistical machine translation (SMT) relies on simple feature functions which make independence assumptions at the level of phrases or hierarchical rules. However, it is well-known that discriminative models can benefit from rich features extracted from the source sentence context outside of the applied phrase or hierarchical rule, which is available at decoding time. We present a framework for the open-source decoder Moses that allows discriminative models over source context to easily be trained on a large number of examples and then be included as feature functions in decoding.

1. Introduction

Phrase-based and hierarchical SMT represent the state of the art for many language pairs. Both of these methods model translation through decomposing the input into segments (phrases, or source sides of hierarchical rules) and translating each one separately. In other words, the translation units are considered independent of each other. Generally, these are scored using relative frequencies. This strong independence assumption often prevents phrase-based or hierarchical models to correctly choose between ambiguous segments. As an example, consider the polysemous French nounphrase “un rapport” where “un” is the masculine indefinite article in French, and “rapport” is a noun which can mean “report” or “relationship”. Figure 1 shows sam-

Corresponding author: tamchyna@ufal.mff.cuni.cz

Cite as: Aleš Tamchyna, Fabienne Braune, Alexander Fraser, Marine Carpuat, Hal Daumé III, Chris Quirk. Integrating a Discriminative Classifier into Phrase-based and Hierarchical Decoding. The Prague Bulletin of Mathematical Linguistics No. 101, 2014, pp. 29–41. doi: 10.2478/pralin-2014-0002.

ple training data with 2 parallel sentences (let us assume that the word alignment is correct, i.e., 1-to-1 and monotonic in both cases).

Sentence 1: il a rédigé un **rapport** .	he has written a **report** .
Sentence 2: un **rapport** entre les coûts et ...	a **relationship** between the costs and ...

Figure 1. Example parallel data for an SMT system.

For this data, the maximum likelihood estimates for "un rapport" - "a relationship" and "un rapport" - "a report" are both 0.5, since they both occur once. If we then observe a test sentence containing "rédigé" (to write), the direct phrasal translation feature function is unable to distinguish between the two translations of "un rapport", although it is clear that the "report" sense should be more probable. The only component in the translation model able to perform some disambiguation in this case is the language model, which is typically employed to ensure the coherence of MT output. However, it only has a limited scope and its estimation suffers from data sparsity when the window size is increased. Consequently, phrase-based and hierarchical models leave space for improving translation quality by conditioning the choice of translation units on contextual information. There have indeed been a number of successful attempts to exploit context on the source (input) side.

In this work, we integrate a discriminative classifier into the open-source decoder Moses in order to score translation rules using richer models of their source context. Related work on discriminatively trained word and phrase lexica will be presented and discussed in Section 5. We provide a complete framework for both phrase-based and hierarchical translation that allows the training of discriminative models over source-side context and the inclusion of classifier predictions in decoding.

The paper is organized as follows: Section 2 describes the relationship between SMT and discriminative classification and provides details of our machine learning setting. In Section 3, we describe the integration into Moses, including the interface for defining new classifier features. Section 4 discusses our experiments. Section 5 concludes the paper with a discussion.

2. Discriminative Classification

We have integrated the high-speed streaming classifier Vowpal Wabbit (VW) into Moses to act as a discriminative phrase lexicon. In this section, we first present the integration of our discriminative model into the Moses translation model. In a second step, we discuss how VW works as discriminative classifier, using a set of label-dependent features. Finally, we show how to train our model using VW.

2.1. Integration into the Moses Translation Model

In Moses, the translation model is implemented as a so-called log-linear model, see Formula 1, which is a linear model that uses feature functions which often look like log probabilities.

$$p_\lambda(E, A, S|F) \propto \exp(\sum_i \lambda_i h_i(A, S, E, F)) \tag{1}$$

In Formula 1, h_i are feature functions and λ_i are the corresponding weights. The formula expresses the probability of sentence E, full sentence word alignment A, and source phrasal segmentation S, given source sentence F. We use lowercase e, f to denote phrases, and a to denote a phrasal word alignment. Feature functions in current (phrasal or hierarchical) SMT systems are typically dense, i.e., they output a small, fixed number of features (feature scores). Baseline features include direct and inverse phrase translation probabilities $p(e, a|f), p(f, a|e)$, lexical weights $p_{lex}(e, a|f)$, $p_{lex}(f, a|e)$, the English language model score $p(E)$ (often implemented as a 5-gram) and others. The conceptually most important feature function in Moses is the feature function modeling $p(e, a|f)$, where f is a source-language phrase, e is a target-language phrase and a is the word alignment between e and f. This is often referred to as the direct phrasal translation probability.

We augment the translation model given in Formula 1 with a new feature function conditioning not only on the source-language phrase but also on the context around it. Formally, we define $p(e, a|f, f')$, where f' represents the input source-language sentence context external to the source-phrase f being translated by this phrase pair. We use discriminative classification to compute this probability because it allows us to use arbitrary information on the source side. We also take advantage of factored machine translation by overloading f and e using factors (we use one factor of morphological tags, and one factor of lemmas). Finally, we allow the external source-side context f' to also access the same factored information (tags and lemmas) in the source sentence external to the phrase being translated. We will sometimes abuse notation to drop a.

2.2. VW as a Discriminative Phrase Lexicon Model

We train VW to choose a target-language (English) phrase e and a word alignment a given a source-language (French) phrase f and all other information available about the source-language sentence F, which we denote f'. We accomplish this by training the classifier on examples extracted from each aligned phrase-pair instance (e, f) in the automatically word-aligned parallel training data. For a new input sentence and particular source-language phrase f, we would like to choose the correct target-language phrase e based on the similarity of f and f' to examples in the training data. To accomplish this, we use label-dependent classification, as outlined below.

Label-dependent Features. In the Discriminative Phrase Lexicon scenario we wish to estimate the probability for a target-language phrase e given a source-language phrase f and the source sentence external context f'. We therefore associate one set of features used for classification with the fixed source phrase and source external context, and have another set of features which varies with the target-language phrase e (i.e., label-dependent). We define the feature space as the $S \times T$ cross-product – this is similar to simply concatenating each source feature with each target feature. We also take the features themselves without concatenation. Figure 2 shows the implemented features on a sample sentence. By taking the cross-product of the source and target features, we obtain a powerful final representation for classification. For instance, we implement the featurization of a discriminative word (rather than phrase) lexicon by taking the cross-product of source-context-bag-of-words and source-phrase-bag-of-words with target-phrase-bag-of-words.

Context

Form:	nous	ne	le	savons	pas	encore	.
Lemma:	il	ne	le	savon	pas	encore	.
Tag:	CLS	ADV	DET	NC	ADV	ADV	PONCT

Phrase Pair

Source:	ne le savons pas
Target:	do not know
Alignment:	0-1 1-2 2-2 3-1
Scores:	-7.5 -9.2 -1.6 -7.5

Features

Source indicator:	p^ne_le_savons w^pas
Target indicator:	p^do_not_know
Source internal:	w^ne w^le w^savons w^pas
Target internal:	w^do w^not w^know
Context:	c^0_-1_nous c^1_-1_il c^2_-1_CLS c^0_1_encore ...
Paired:	p^ne_not p^le_know p^savons_know p^pas_not
Scores:	sc^0_-10 sc^0_-9 sc^0_-8 sc^1_-10 sc^2_-10 ...

Figure 2. Implemented Features

Vowpal Wabbit. To implement label-dependent classification, we chose to use Vowpal Wabbit (VW),[1] implemented by John Langford. VW has a fast implementation of stochastic gradient descent and L-BFGS for many different loss functions. VW is

[1] http://hunch.net/~vw/

widely used for machine learning tasks. VW was built into a library for the work reported here. VW has built-in support for:

- Feature hashing (scaling to billions of features)
- Caching (no need to re-parse text)
- Different losses and regularizers
- Reductions framework to binary classification/regression
- Multithreading/multicore processing.

2.3. Training the discriminative Model

For each (e, f) instance in the file with extracted phrase pairs, we create a training example. We first perform significance testing (Johnson et al., 2007) to reduce the total number of (e, f) types in the phrase table. We then extract one training example per (e, f) instance extracted. We generate a line for each possible translation of f in the reduced phrase table. The correct translation is assigned a *loss* of zero, all other translations of f get a loss of 1.

Example. As an example, consider the polysemous French noun-phrase "un rapport" introduced in Section 1, which can either be translated as "report" or "relationship". We have shown that although the translation model cannot adequately choose between these translations, it is clear that in a test sentence containing "rédigé" (to write), the "report" translation should be more probable. We can operationalize this with label-dependent features as shown in Figure 3.

	Source Namespace	Target Namespace	Loss
Sentence 1:	p^un_rapport c^il c^a **c^rédigé**	p^a_report	0
		p^a_relationship	1
Sentence 2:	p^un_rapport c^entre c^les ...	p^a_report	1
		p^a_relationship	0

Figure 3. Training examples with label-dependent features extracted from sample parallel data.

The features prefixed with "p^" are the phrases being modeled. The features prefixed with "c^" implement the bag-of-words feature. Implementation details of the training procedure are given in the next section. In our example, during training the model can learn from Sentence 1 that the $S \times T$ cross-product feature c^rédigé.p^a_report should push the loss towards zero. At testing, this allows "a report" to be chosen.

We train VW using the cost-sensitive one-against-all reduction and label-dependent features. We use the dev set (the same dev set as is used for MERT) to perform early stopping.

3. Integration of VW into Moses

In this section, we present the engineering details of the integration of VW into the phrase-based and hierarchical components of Moses. As an overall illustration, we compare the standard Moses pipeline and a pipeline with integrated classifier. To this aim, consider again the polysemous French noun-phrase “un rapport” presented in Section 1. In the standard pipeline, shown in Figure 4 (left), phrases or hierarchical rules are extracted from the word-aligned parallel data and scored using maximum likelihood estimation.

During decoding, the scored units are applied. As noted in sections 1 and 2, this pipeline does not allow to choose the correct translation of “un rapport” given the source sentence context. In a pipeline where VW is integrated to Moses, shown in figure 4 (right), the training procedure is augmented with an additional step to train the discriminative model using VW. The details of classifier training have been presented in Section 2.3. During decoding, the trained model is queried and the obtained prediction is added to the log-linear model, as shown in Section 2.1. This additional score allows the system to choose the correct translation of “un rapport” given the source context.

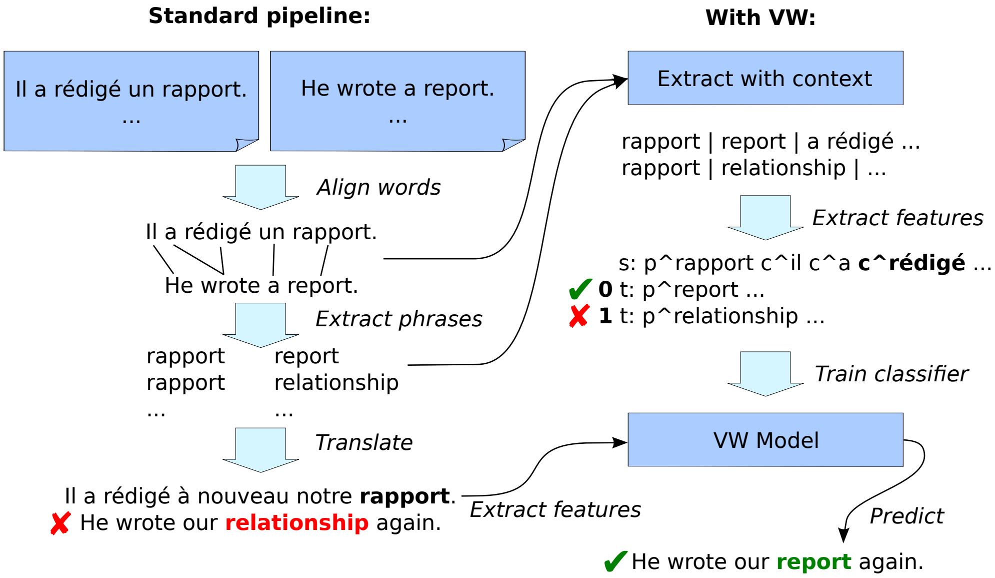

Figure 4. Classifier Pipeline. Training examples are extracted along with context which helps disambiguate phrasal translations.

We integrate VW in training and decoding. For both tasks an interface between the MT system (Moses) and the learner (VW) has to be created.

We begin by presenting the overall architecture of the interface between Moses and VW in Section 3.1. Then we explain how to train such a model in Section 3.2 and query it during decoding in Section 3.3.

3.1. Overall Architecture

We integrate the learner (VW) into the MT system (Moses) by defining a single library used for training the discriminative model as well as getting model predictions during decoding. Using this library avoids code duplication but even more importantly assures consistent definition and configuration of features for model training and prediction. Also, it makes the overall architecture simple and extensible. The design of the library is inspired by the Producer/Consumer pattern: we decouple (i) feature extraction from training and input data and (ii) generation of features provided to the learner. (i) is implemented by the *Feature Extractor* interface and (ii) by the *Feature Consumer* interface. The design chosen also allows us to clearly separate the logic of feature generation from the specificities of the learner. The only requirement needed to add different learners is the implementation of the *FeatureConsumer* interface. For instance, it would be easy to add such an interface for MegaM,[2] which we leave for future work.

The *FeatureExtractor* interface supports extraction of various types of features from the source sentence context as well as from the source and target side of phrases (for phrase-based SMT) or hierarchical rules (for hierarchical SMT). Feature extraction is controlled through the specification of a configuration file. The current implementation supports the features presented in Section 2.2.

We provide 3 implementations of the *FeatureConsumer*. The two first are used to train the discriminative model and the third to get model predictions. More precisely, the *VWFileTrainConsumer* extracts features for training VW in text format and stores them in an output file. We use this class in training to visually inspect the generated features before training the VW model. Otherwise, it is possible to feed the features directly into VW and train the model using *VWLibraryTrainConsumer*. In decoding, *VWLibraryPredictConsumer* is used to get VW predictions.

The same library is implemented in the phrase-based and in the hierarchical components of Moses.

3.2. Training the Model

For training our discriminative model, we first modify the phrase and hierarchical rule extraction algorithms provided in the phrase-based and hierarchical components

[2] `http://www.umiacs.umd.edu/~hal/megam/`

of Moses. The modified routines output an additional file indicating in which source sentence and at which position each phrase and rule have been extracted. These annotations allow us to extract ”source sentence context” features for each phrase and hierarchical rule. In order to provide a richer context, the source side training data is augmented with a factored annotation containing morphological and POS tags. Using these annotations as well as the parallel corpus, training examples for VW are extracted using the feature library described in Section 3.1. The *FeatureExtractor*, controlled by the configuration file, specifies which features are extracted while the *FeatureConsumer* generates the training examples for VW.

3.3. Getting Predictions during Decoding

Model predictions queried during decoding are implemented as a feature function which is added to the log-linear model and consequently tuned on a held-out data set. In its phrase-based and hierarchical components, Moses offers an interface for defining feature functions. Our model predictions are integrated into the decoder by implementing this interface (with some tricks). The feature function for the phrase-based component is located in the class *ClassifierFeature* while the feature function for the hierarchical component is in *ContextFeature*.

In the phrase-based as well as the hierarchical component of Moses, the task of the feature function is to re-evaluate each translation option by querying VW according to the source sentence context in which they occur. More precisely, for each translation option applying to a given span, source context features are extracted using the library described in Section 3.1.

Using the extracted features, the *VWLibraryPredictConsumer* is used to query the VW model. The obtained predictions are then normalized to transform model scores (losses) into probabilities. The feature function is evaluated prior to phrase- or rule-table pruning and decoding. This allows us to save computation and to avoid discarding options which our feature considers good.

Note that in order to implement this behavior, we needed to deviate a little from the standard feature interface: our feature function is called immediately after translation options are collected and evaluates all translation options for a given source span at the same time (to allow normalization).

3.4. Integration into the Hierarchical Component

Integration of a discriminative model into a hierarchical system is generally more challenging than the integration into a phrase-based system. The main reasons are that (i) many left-hand-sides of hierarchical rules can apply to the same source sentence span and (ii) a single left-hand-side of a hierarchical rule can apply to many source sentence spans. As a consequence, many more rules have to be collected to generate training examples and many more translation options have to be re-evaluated during decoding.

As an illustration, consider the French segment “patiente diabétique et enceinte” (an English word-by-word gloss is “patient diabetic and pregnant”, it means “diabetic and pregnant patient”). Consider that the following hierarchical rules have been extracted from the training data.

r1 $X/X \rightarrow <X_0$ enceinte , pregnant $X_0>$
r2 $X/X \rightarrow <X_0$ enceinte , X_0 enclosure$>$
r3 $X/X \rightarrow <$patiente X_0 , X_0 patient$>$
r4 $X/X \rightarrow <$patiente X_0 et X_1, X_0 and X_1 patient$>$

Each rule consists of lexical items and aligned non-terminal symbols. All rules presented above match the source segment “patiente diabétique et enceinte” although rules r1, r3 and r4 have different source language sides. On the other hand, rules with source side “X enceinte” can apply to the complete segment “patiente diabétique et enceinte” or to the last three or two words.

As shown in Section 2.2, features extracted to train and query the integrated model include source context features. The extraction of these features requires information about the source sentence context surrounding the place where a rule has been extracted (training) or applied (decoding). In a phrase-based system, the source context is the context surrounding a single source phrase. In a hierarchical system, the source context cannot be attached to a single source side of rule because several rules can match the same source language segment. Hence computing a discriminative score for hierarchical rules according to their context of occurrence in the source language sentence requires to collect, for a given span, all hierarchical rules applying to this span. For instance when considering the context surrounding the linguistic segment ”patiente diabétique et enceinte” all rules applying to the span beginning at ”patiente” and ending at ”diabetique” have to be collected. Such rules include rules r1,r2 and r3.

4. Experiments

Performance. In terms of performance, the phrase-based component of Moses with the discriminative model takes 80% relative longer than the Moses baseline without VW. The hierarchical component is slower (300% relative longer) due to the additional complexity described in Section 3.4.

We made queries to VW thread-safe and tested all of our code in a parallel setting.

The classifier feature is also fully integrated in Moses’ Experiment Management System (EMS, experiment.perl) which allows potential new users to quickly create experiments with our feature.

Phrase-based Experiments. We used our feature in a setting similar to phrase-sense disambiguation (PSD, Carpuat and Wu, 2007), utilizing all of the classifier features described in Section 3.1. We utilized science domain training data consisting of

113291 French and English parallel sentences, as well as dev and test sets distributed with this data.[3] We computed GIZA++ word alignments by using a much larger parallel French/English corpus of over 2 million parallel sentences (this was only used to improve the word alignment). We trained a 5-gram language model using SRILM and decoded using KenLM. Table 1 shows the results of our experiment. Our feature receives a moderately high weight (the weight of the direct phrasal translation probability is 0.10). Our integrated system beats the baseline by a difference of 0.60 BLEU.

Source	Target	BLEU		Feature
		Baseline	+Classifier	Weight
French	English	32.62	33.22	0.05

Table 1. Results of experiment with phrase-based translation

Hierarchical Experiments. The same experiment has been conducted using the hierarchical component of the system. Table 2 shows the results. Our feature receives a high weight and the integrated system beats the baseline by a difference of 0.53 BLEU. Overall, the hierarchical system performs worse than phrase-based for this experiment.

Source	Target	BLEU		Feature
		Baseline	+Classifier	Weight
French	English	31.08	31.61	0.14

Table 2. Results of experiment with hierarchical translation

5. Discussion and Conclusion

In most SMT architectures, translation rules are scored based on their relative frequency in the parallel training corpus. However, integrating richer information into translation decisions is an active area of research.

We integrated a discriminative classifier into Moses in order to score translation rules using richer models of their source context. This contrasts with the feature-rich approaches already available in Moses. For instance, factored translation models (Koehn and Hoang, 2007) can be used to define translation rules based on lemma, POS, or other representations of phrases, but these rules are still scored using relative

[3] http://www.umiacs.umd.edu/~hal/damt/

frequency. Source context features, such as words and part-of-speech tags surrounding a given source phrase, can also be directly made available to the decoder as features in the log-linear model (Gimpel and Smith, 2008). While this approach presents the advantage of directly optimizing feature weights for BLEU or other metrics of translation quality, it suffers from current limitations with large-scale discriminative training of SMT systems, as discussed in Section 2.

Our approach is inspired by context-dependent *phrase* lexicons for phrase-based SMT models (Giménez and Màrquez, 2007; Stroppa et al., 2007; Carpuat and Wu, 2007; Weller, 2010; Haque et al., 2011). These models generalize early discriminative word translation models (Berger et al., 1996) to current phrase-based SMT. Unlike in Berger et al. (1996), our phrasal translations are conditioned on the observed source sentence context, rather than on the hypothesized target language context, which facilitates the integration of context on the source side, such as the local and long-distance clues used in word sense disambiguation.

Other work has focused on *word*-level discriminative lexicons (Bangalore et al., 2007; Mauser and Ney, 2009; Venkatapathy and Bangalore, 2009), which predict which words should occur in the output sentence based on a bag-of-words representation of the source sentence. Extensions include Niehues and Waibel (2013) who enrich the bag-of-word context with n-grams and syntactic features, and Jeong et al. (2010), who focus on translation into morphologically richer languages.

Context-dependent scoring of translation rules can make decoding significantly more expensive, as context-dependent translation probabilities cannot be pre-computed once and for all at training time. Previous experimental implementations expanded phrase-tables to represent phrase instances rather than types (Giménez and Màrquez, 2007). In contrast, we integrate a fast online classifier designed for large data directly into the decoder. Our implementation uses a single global model for disambiguating all source phrases, rather than one model per phrase type as in Carpuat and Wu (2007); Giménez and Màrquez (2007) and in the Moses implementation of Mauser and Ney (2009).[4]n addition to making the implementation and training easier and more efficient, this approach can also potentially capture generalizations across phrase types as observed for word lexicons in Jeong et al. (2010).

In hierarchical phrase-based SMT, classifiers have been used to disambiguate between translations of words and short phrases (Chan et al., 2007), and to model reordering decisions (Xiong et al., 2006). In contrast, our implementation lets us directly score each translation rule, and can simultaneously be used to model soft syntactic constraints and lexical disambiguation clues.

Our work is also somewhat related to so-called sparse features (implemented in Moses by Hasler et al., 2011). The weights of these features are trained on the development set using MIRA. This is advantageous, as the weights of features associated with the development set are optimized to maximize the final performance criterion

[4] `http://www.statmt.org/moses/?n=Moses.AdvancedFeatures#ntoc32`

(BLEU) directly. However, the development set is typically very small, which limits the coverage of the features which can be effectively trained. We avoid the problem of small dev sets by training our VW model on the training set. Our feature is then just one score in the log-linear model.

We have integrated VW into Moses and provided a proof-of-concept implementation of a discriminative phrase lexicon. In the future we plan to implement other feature functions and integrate other classifiers. All of our code is publicly available in the Moses repository in the branches *damt_phrase* and *syntaxContext*.

Acknowledgement

This work was partially supported by Deutsche Forschungsgemeinschaft grant Models of Morphosyntax for Statistical Machine Translation (Phase 2), by NSF Grant No 1005411, by DARPA CSSG Grant D11AP00279, by grant FP7-ICT-2011-7-288487 (MosesCore) of the European Union, by project LH12093 of the Ministry of Education, Youth and Sports of the Czech Republic, and by SVV project number 260104.

A. Installation

1. Download and compile VW:

```
git clone https://github.com/JohnLangford/vowpal_wabbit.git
cd vowpal_wabbit
./autogen.sh --prefix=`pwd`
make && make install
```

2. Download and compile Moses:

```
git clone https://github.com/moses-smt/mosesdecoder.git
cd mosesdecoder
git checkout damt_phrase # or syntaxContext for hiero
./bjam --with-vw=<path-to-vowpal-wabbit>
```

3. See the provided sample EMS configuration file and INI file for the VW feature function:

```
mosesdecoder/scripts/ems/example/config.psd
mosesdecoder/scripts/ems/example/data/psd-features.ini
```

Bibliography

Bangalore, Srinivas, Patrick Haffner, and Stephan Kanthak. Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction. In *ACL*, 2007.

Berger, Adam L., Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), 1996.

Carpuat, Marine and Dekai Wu. Improving statistical machine translation using word sense disambiguation. In *EMNLP*, 2007.

Chan, Yee Seng, Hwee Tou Ng, and David Chiang. Word Sense Disambiguation Improves Statistical Machine Translation. In *ACL*, 2007.

Giménez, Jesús and Lluís Màrquez. Context-aware Discriminative Phrase Selection for Statistical Machine Translation. In *WMT*, 2007.

Gimpel, Kevin and Noah A. Smith. Rich source-side context for statistical machine translation. In *WMT*, 2008.

Haque, Rejwanul, Sudip Kumar Naskar, Antal Bosch, and Andy Way. Integrating source-language context into phrase-based SMT. *Machine Translation*, 25(3), 2011.

Hasler, Eva, Barry Haddow, and Philipp Koehn. Margin infused relaxed algorithm for Moses. *Prague Bull. Math. Linguistics*, 96, 2011.

Jeong, Minwoo, Kristina Toutanova, Hisami Suzuki, and Chris Quirk. A discriminative lexicon model for complex morphology. In *AMTA*, 2010.

Johnson, Howard, Joel Martin, George Foster, and Roland Kuhn. Improving Translation Quality by Discarding Most of the Phrasetable. In *Proc. of EMNLP-CoNLL 2007*, 2007.

Koehn, Philipp and Hieu Hoang. Factored translation models. In *EMNLP*, 2007.

Mauser, Arne and Hermann Ney. Extending statistical machine translation with discriminative and trigger-based lexicon models. In *EMNLP*, 2009.

Niehues, Jan and Alex Waibel. An MT error-driven discriminative word lexicon using sentence structure features. In *WMT*, 2013.

Stroppa, Nicolas, Antal van den Bosch, and Andy Way. Exploiting source similarity for SMT using context-informed features. In *TMI 2007*, 2007.

Venkatapathy, Sriram and Srinivas Bangalore. Discriminative machine translation using global lexical selection. *TALIP*, 8(2), May 2009.

Weller, Marion. An empirical analysis of source context features for phrase-based statistical machine translation. Diploma thesis, Universität Stuttgart, 2010.

Xiong, Deyi, Qun Liu, and Shouxun Lin. Maximum entropy based phrase reordering model for statistical machine translation. In *ACL*, 2006.

Address for correspondence:
Aleš Tamchyna
tamchyna@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