Word Translation Prediction for Morphologically Rich Languages with Bilingual Neural Networks

Ke Tran Arianna Bisazza Christof Monz
Informatics Institute, University of Amsterdam
Science Park 904, 1098 XH Amsterdam, The Netherlands
{m.k.tran,a.bisazza,c.monz}@uva.nl

Abstract

Translating into morphologically rich languages is a particularly difficult problem in machine translation due to the high degree of inflectional ambiguity in the target language, often only poorly captured by existing word translation models. We present a general approach that exploits source-side contexts of foreign words to improve translation prediction accuracy. Our approach is based on a probabilistic neural network which does not require linguistic annotation nor manual feature engineering. We report significant improvements in word translation prediction accuracy for three morphologically rich target languages. In addition, preliminary results for integrating our approach into a large-scale English-Russian statistical machine translation system show small but statistically significant improvements in translation quality.

1 Introduction

The ability to make context-sensitive translation decisions is one of the major strengths of phrase-based SMT (PSMT). However, the way PSMT exploits source-language context has several limitations as pointed out, for instance, by Quirk and Menezes (2006) and Durrani et al. (2013). First, the amount of context used to translate a given input word depends on the phrase segmentation, with hypotheses resulting from different segmentations competing with one another. Another issue is that, given a phrase segmentation, each source phrase is translated independently from the others, which can be problematic especially for short phrases. As a result, the predictive translation of a source phrase does not access useful linguistic clues in the source sentence that are outside of the scope of the phrase.

Lexical weighting tackles the problem of unreliable phrase probabilities, typically associated with long phrases, but does not alleviate the problem of context segmentation. An important share of the translation selection task is then left to the language model (LM), which is certainly very effective but can only leverage *target* language context. Moreover, decisions that are taken at early decoding stages—such as the common practice of retaining only top n translation options for each source span—depend only on the translation models and on the target context available in the phrase.

Source context based translation models (Gimpel and Smith, 2008; Mauser et al., 2009; Jeong et al., 2010; Haque et al., 2011) naturally address these limitations. These models can exploit a boundless context of the input text, but they assume that target words can be predicted independently from each other, which makes them easy to integrate into state-of-the-art PSMT systems. Even though the independence assumption is made on the target side, these models have shown the benefits of utilizing source context, especially in translating into morphologically rich languages. One drawback of previous research on this topic, though, is that it relied on rich sets of manually designed features, which in turn required the availability of linguistic annotation tools like POS taggers and syntactic parsers.

In this paper, we specifically focus on improving the prediction accuracy for word translations. Achieving high levels of word translation accuracy is particularly challenging for language

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1676–1688,
October 25-29, 2014, Doha, Qatar.

pairs where the source language is morphologically poor, such as English, and the target language is morphologically rich, such as Russian, i.e., language pairs with a high degree of surface realization ambiguity (Minkov et al., 2007). To address this problem we propose a general approach based on bilingual neural networks (BNN) exploiting source-side contextual information.

This paper makes a number of contributions: Unlike previous approaches our models do not require any form of linguistic annotation (Minkov et al., 2007; Kholy and Habash, 2012; Chahuneau et al., 2013), nor do they require any feature engineering (Gimpel and Smith, 2008). Moreover, besides directly predicting fully inflected forms as Jeong et al. (2010), our approach can also model stem and suffix prediction explicitly. Prediction accuracy is evaluated with respect to three morphologically rich target languages (Bulgarian, Czech, and Russian) showing that our approach consistently yields substantial improvements over a competitive baseline. We also show that these improvements in prediction accuracy can be beneficial in an end-to-end machine translation scenario by integrating into a large-scale English-Russian PSMT system. Finally, a detailed analysis shows that our approach induces a positive bias on phrase translation probabilities leading to a better ranking of the translation options employed by the decoder.

2 Lexical coverage of SMT models

The first question we ask is whether translation can be improved by a more accurate selection of the translation options already existing in the SMT models, as opposed to generating new options. To answer this question we measure the lexical coverage of a baseline PSMT system trained on English-Russian.[1] We choose this language pair because of the morphological richness on the target side: Russian is characterized by a highly inflectional morphology with a particularly complex nominal declension (six core cases, three genders and two number categories). As suggested by Green and DeNero (2012), we compute the recall of reference tokens in the set of target tokens that the decoder could produce in a translation of the source, that is the target tokens of all phrase pairs that matched the input sentence and that were actually used for decoding.[2] We call this the decoder's *lexical search space*. Then, we compare the reference/space recall against the reference/MT-output recall: that is, the percentage of reference tokens that also appeared in the 1-best translation output by the SMT system. Results for the WMT12 benchmark are presented in Table 1. From the first two rows, we see that only a rather small part of the correct target tokens available to the decoder are actually produced in the 1-best MT output (50% against 86%). Although our word-level analysis does not directly estimate phrase-level coverage, these numbers suggest that a large potential for translation improvement lies in better lexical selection during decoding.

Token recall:	
reference/MT-search-space	86.0%
reference/MT-output	50.0%
stem-only reference/MT-output	12.3%
of which reachable	11.2%

Table 1: Lexical coverage analysis of the baseline SMT system (English-Russian wmt12).

To quantify the importance of morphology, we count how many reference tokens matched the MT output only at the stem level[3] and for how many of those the correct surface form existed in the search space (reachable matches). These two numbers represent the upper bound of the improvement achievable by a model only predicting suffixes given the target stems. As shown in Table 1, such a model could potentially increase the reference/MT-output recall by 12.3% with generation of new inflected forms, and by 11.2% without. Thus, also when it comes to morphology, generation seems to be of secondary importance compared to better selection in our experimental setup.

3 Predicting word translations in context

It is standard practice in PSMT to use word-to-word translation probabilities as an additional phrase score. More specifically, state-of-the-art PSMT systems employ the maximum-likelihood estimate of the context-independent probability of a target word given its aligned source word $P(t_j|s_i)$ for each word alignment link a_{ij}.

[1] Training data and SMT setup are described in Section 6.

[2] This corresponds to the top 30 phrases sorted by weighted phrase, lexical and LM probabilities, for each source span. Koehn (2004) and our own experience suggest that using more phrases has little or no impact on MT quality.

[3] Word segmentation for this analysis is performed by the Russian Snowball stemmer, see also Section 5.3.

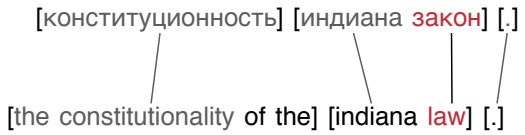


Figure 1: Fragment of English sentence and its incorrect Russian translation produced by the baseline SMT system. Square brackets indicate phrase boundaries.

The main goal of our work is to improve the estimation of such probabilities by exploiting the context of s_i, which in turn we expect will result in better phrase translation selection. Figure 1 illustrates this idea: the translation of "law" in this example has a wrong case—nominative instead of genitive. Due to the rare word "Indiana/индиана", the target LM must backoff to the bigram history and does not penalize this choice sufficiently. However, a model that has access to the word "of" in the near source context could bias the translation of "law" to the correct case.

We then model $P(t_j|\mathbf{c}_{s_i})$ with source context $\mathbf{c}_{s_i}$ defined as a fixed-length word sequence centered around s_i:

$$\mathbf{c}_{s_i} = s_{i-k}, ..., s_i, ..., s_{i+k}$$

Our definition of context is similar to the $n-1$ word history used in n-gram LMs. Similarly to previous work in source context-sensitive translation modeling (Jeong et al., 2010; Chahuneau et al., 2013), target words are predicted independently from each other, which allows for an efficient decoding integration. We are particularly interested in translating into morphologically rich languages where source context can provide useful information for the prediction of target translation, for example, the gender of the subject in a source sentence constrains the morphology of the translation of the source verb. Therefore, we integrate the notions of stem and suffix directly into the model. We assume the availability of a word segmentation function g that takes a target word t as input and returns its stem and suffix: $g(t) = (\sigma, \mu)$. Then, the conditional probability $p(t_j|\mathbf{c}_{s_i})$ can be decomposed into stem probability and suffix probability:

$$p(t_j|\mathbf{c}_{s_i}) = p(\sigma_j|\mathbf{c}_{s_i})p(\mu_j|\mathbf{c}_{s_i}, \sigma_j) \quad (1)$$

These two probabilities can be estimated separately, which yields the two subtasks:

1. predict target stem σ given source context $\mathbf{c}_s$;
2. predict target suffix μ given source context $\mathbf{c}_s$ and target stem σ.

Based on the results of our analysis, we focus on the selection of existing translation candidates. We then restrict our prediction on a set of possible target candidates depending on the task instead of considering all target words in the vocabulary. More specifically, for each source word s_i, our candidate generation function returns the set of target words $T_s = \{t_1, \ldots, t_m\}$ that were aligned to s_i in the parallel training corpus, which in turn corresponds to the set of target words that the SMT system can produce for a given source. In practice, we use a pruned version of T_s to speed up training and reduce noise (see details in Section 5).

As for the morphological models, given T_s and g, we can obtain $L_s = \{\sigma_1, \ldots, \sigma_k\}$, the set of possible target stem translations of s, and $M_\sigma = \{\mu_1, \ldots, \mu_l\}$, the set of possible suffixes for a target stem σ. The use of L_s, and M_σ is similar to stemming and inflection operations in (Toutanova et al., 2008) while the set T_s is similar to the GEN function in (Jeong et al., 2010).[4]

Our approach differs crucially from previous work (Minkov et al., 2007; Chahuneau et al., 2013) in that it does not require linguistic features such as part-of-speech and syntactic tree on the source side. The proposed models automatically learn features that are relevant for each of the modeled tasks, directly from word-aligned data. To make the approach completely language independent, the word segmentation function g can be trained with an unsupervised segmentation tool. The effects of using different word segmentation techniques are discussed in Section 5.

4 Bilingual neural networks for translation prediction

Probabilistic neural network (NN), or continuous space, language models have received increasing attention over the last few years and have been applied to several natural language processing tasks (Bengio et al., 2003; Collobert and Weston, 2008; Socher et al., 2011; Socher et al., 2012). Within statistical machine translation, they

[4]Note that our suffix generation function M_σ is restricted to the forms observed in the target monolingual data, but not to those aligned to a source word s, which opens the possibility of generating inflected forms that are missing from the translation models. We leave this possibility to future work.

have been used for monolingual target language modeling (Schwenk et al., 2006; Le et al., 2011; Duh et al., 2013; Vaswani et al., 2013), n-gram translation modeling (Son et al., 2012), phrase translation modeling (Schwenk, 2012; Zou et al., 2013; Gao et al., 2014) and minimal translation modeling (Hu et al., 2014). The recurrent neural network LMs of Auli et al. (2013) are primarily trained to predict target word sequences. However, they also experiment with an additional input layer representing source side context.

Our models differ from most previous work in neural language modeling in that we predict a target translation given a source context while previous models predict the next word given a target word history. Unlike previous work in phrase translation modeling with NNs, our models have the advantage of accessing source context that can fall outside the phrase boundaries.

We now describe our models in a general setting, predicting target translations given a source context, where target translations can be either words, stems or suffixes.[5]

4.1 Neural network architecture

Following a common approach in deep learning for NLP (Bengio et al., 2003; Collobert and Weston, 2008), we represent each source word s_i by a column vector $\mathbf{r}_{s_i} \in \mathbb{R}^d$. Given a source context $\mathbf{c}_{s_i} = s_{i-k}, ..., s_i, ..., s_{i+k}$ of k words on the left and k words on the right of s_i, the context representation $\mathbf{r}_{\mathbf{c}_{s_i}} \in \mathbb{R}^{(2k+1)\times d}$ is obtained by concatenating the vector representations of all words in $\mathbf{c}_{s_i}$:

$$\mathbf{r}_{\mathbf{c}_{s_i}} = \mathbf{r}_{s_{i-k}} \odot ... \odot \mathbf{r}_{s_{i+k}}$$

Our main BNN architecture for word or stem prediction (Figure 2a) is a feed-forward neural network (FFNN) with one hidden layer, a matrix $\mathbf{W}_1 \in \mathbb{R}^{n\times(2k+1)d}$ connecting the input layer to the hidden layer, a matrix $\mathbf{W}_2 \in \mathbb{R}^{|V_t|\times n}$ connecting the hidden layer to the output layer, and a bias vector $\mathbf{b}_2 \in \mathbb{R}^{|V_t|}$ where $|V_t|$ is the size of target translations vocabulary. The target translation distribution $P_{\text{BNN}}(t|\mathbf{c}_{s_i})$ for a given source context $\mathbf{c}_{s_i}$ is computed by a forward pass:

$$\text{softmax}\left(\mathbf{W}_2\, \phi(\mathbf{W}_1 \mathbf{r}_{\mathbf{c}_{s_i}}) + \mathbf{b}_2\right) \qquad (2)$$

where ϕ is a nonlinearity (tanh, sigmoid or rectified linear units). The parameters of the neural network are $\boldsymbol{\theta} = \{\mathbf{r}_{s_i}, \mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_2\}$.

The suffix prediction BNN is obtained by adding the target stem representation $\mathbf{r}_\sigma$ to the input layer (see Figure 2b).

4.2 Model variants

We encounter two major issues with FFNNs: (i) They do not provide a natural mechanism to compute word surface conditional probability $p(t|\mathbf{c}_s)$ given individual stem probability $p(\sigma|\mathbf{c}_s)$ and suffix probability $p(\mu|\mathbf{c}_s, \sigma)$, and (ii) FFNNs do not provide the flexibility to capture long dependencies among words if they lie outside the source context window. Hence, we consider two BNN variants: a log-bilinear model (LBL) and a convolutional neural network model (ConvNet). LBL could potentially address (i) by factorizing target representations into target stem and suffix representations whereas ConvNets offer the advantage of modeling variable input length (ii) (Kalchbrenner et al., 2014).

Log-bilinear model. The FFNN models stem and suffix probabilities separately. A log-bilinear model instead could directly model word prediction through a factored representation of target words, similarly to Botha and Blunsom (2014). Thus, no probability mass would be wasted over stem-suffix combinations that are not in the candidate generation function. The LBL model specifies the conditional distribution for the word translation $t_j \in T_{s_i}$ given a source context $\mathbf{c}_{s_i}$:

$$P_{\boldsymbol{\theta}}(t_j|\mathbf{c}_{s_i}) = \frac{\exp(s_{\boldsymbol{\theta}}(t_j, \mathbf{c}_{s_i}))}{\sum_{t'_j \in T_{s_i}} \exp(s_{\boldsymbol{\theta}}(t'_j, \mathbf{c}_{s_i}))} \qquad (3)$$

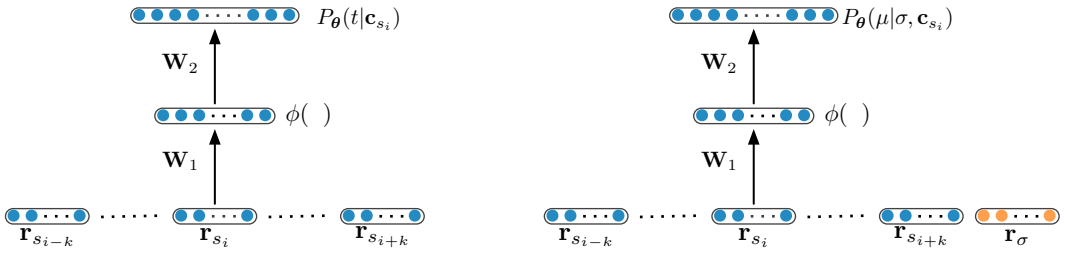
We use an additional set of word representations $\mathbf{q}_{t_j} \in \mathbb{R}^n$ for target translations t_j. The LBL model computes a predictive representation $\hat{\mathbf{q}}$ of a source context $\mathbf{c}_{s_i}$ by taking a linear combination of the source word representations $\mathbf{r}_{s_{i+m}}$ with the position-dependent weight matrices $\mathbf{C}_m \in \mathbb{R}^{n\times d}$:

$$\hat{\mathbf{q}} = \sum_{m=-k}^{k} \mathbf{C}_m \mathbf{r}_{s_{i+m}} \qquad (4)$$

The score function $s_{\boldsymbol{\theta}}(t_j, \mathbf{c}_{s_i})$ measures the similarity between the predictive representation $\hat{\mathbf{q}}$ and the target representation $\mathbf{q}_{t_j}$:

$$s_{\boldsymbol{\theta}}(t_j, \mathbf{c}_{s_i}) = \hat{\mathbf{q}}^\top \mathbf{q}_{t_j} + \mathbf{b}_h^\top \mathbf{q}_{t_j} + b_{t_j} \qquad (5)$$

[5] The source code of our models is available at `https://bitbucket.org/ketran`



(a) BNN for word prediction.

(b) BNN for suffix prediction.

Figure 2: Feed-forward BNN architectures for predicting target translations: (a) word model (similar to stem model), and (b) suffix model with an additional vector representation $\mathbf{r}_\sigma$ for target stems σ.

Here b_{t_j} is the bias term associated with target word t_j. $\mathbf{b}_h \in \mathbb{R}^n$ are the representation biases. $s_{\boldsymbol{\theta}}(t_j, \mathbf{c}_{s_i})$ can be seen as the negative energy function of the target translation t_j and its context $\mathbf{c}_{s_i}$. The parameters of the model thus are $\boldsymbol{\theta} = \{\mathbf{r}_{s_i}, \mathbf{C}_m, \mathbf{q}_{t_j}, \mathbf{b}_h, b_{t_j}\}$. Our log-bilinear model is a modification of the log-bilinear model proposed for n-gram language modeling in (Mnih and Hinton, 2007).

Convolutional neural network model. This model (Figure 3) computes the predictive representation $\hat{\mathbf{q}}$ by applying a sequence of $2k$ convolutional layers $\{\mathbf{L}_1, \ldots, \mathbf{L}_{2k}\}$. The source context $\mathbf{c}_{s_i}$ is represented as a matrix $\mathbf{m}_{c_{s_i}} \in \mathbb{R}^{d\times(2k+1)}$:

$$\mathbf{m}_{c_{s_i}} = \left[\mathbf{r}_{s_{i-k}}; \ldots; \mathbf{r}_{s_{i+k}}\right] \tag{6}$$

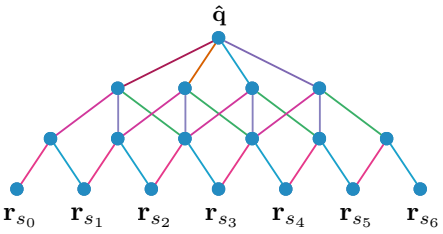


Figure 3: Convolutional neural network model. Edges with the same color indicate the same kernel weight matrix.

Each convolutional layer $\mathbf{L}_i$ consists of a one-dimensional filter $\mathbf{m}_i \in \mathbb{R}^{d\times 2}$. Each row of $\mathbf{m}_i$ is convolved with the corresponding row in the previous layer resulting in a weight matrix whose number of columns decreases by one. Thus after $2k$ convolutional layers, the network transforms the source context matrix $\mathbf{m}_{c_{s_i}}$ to a feature vector $\hat{\mathbf{q}} \in \mathbb{R}^d$. A fully connected layer with weight matrix $\mathbf{W}$ followed by a softmax layer are placed after the last convolutional layer $\mathbf{L}_{2k}$ to perform classification. The parameters of the convolutional neural network model are $\boldsymbol{\theta} = \{\mathbf{r}_{s_i}, \mathbf{m}_j, \mathbf{W}\}$. Here, we focus on a fixed length input, however convolutional neural networks may be used to model variable length input (Kalchbrenner et al., 2014; Kalchbrenner and Blunsom, 2013).

4.3 Training

In training, for each example $(t, \mathbf{c}_s)$, we maximize the conditional probability $P_{\boldsymbol{\theta}}(t|\mathbf{c}_s)$ of a correct target label t. The contribution of the training example $(t, \mathbf{c}_s)$ to the gradient of the log conditional probability is given by:

$$\frac{\partial}{\partial\boldsymbol{\theta}} \log P_{\boldsymbol{\theta}}(t|\mathbf{c}_s) = \frac{\partial}{\partial\boldsymbol{\theta}} s_{\boldsymbol{\theta}}(t|\mathbf{c}_s) - \sum_{t'\in T_s} P_{\boldsymbol{\theta}}(t'|\mathbf{c}_s) \frac{\partial}{\partial\boldsymbol{\theta}} s_{\boldsymbol{\theta}}(t', \mathbf{c}_s)$$

Note that in the gradient, we do not sum over all target translations T but a set of possible candidates T_s of a source word s. In practice $|T_s| \leq 200$ with our pruning settings (see Section 5.1), thus training time for one example does not depend on the vocabulary size. Our training criterion can be seen as a form of contrastive estimation (Smith and Eisner, 2005), however we explicitly move the probability mass from *competing* candidates to the correct translation candidate, thus obtaining more reliable estimates of the conditional probabilities.

The BNN parameters are initialized randomly according to a zero-mean Gaussian. We regularize the models with L_2. As an alternative to the L_2 regularizer, we also experiment with dropout (Hinton et al., 2012), where the neurons are randomly zeroed out with dropout rate p. This technique is known to be useful in computer vision tasks but has been rarely used in NLP tasks. In FFNN, we use dropout after the hidden layer, while in ConvNet, dropout applies after the last convolutional layer. The dropout rate p is set to 0.3 in our exper-

iments. We use rectified nonlinearities[6] in FFNN and after each convolutional layer in ConvNet. We train our BNN models with the standard stochastic gradient descent.

5 Evaluating word translation prediction

In this section, we assess the ability of our BNN models to predict the correct translation of a word in context. In addition to English-Russian, we also consider translation prediction for Czech and Bulgarian. As members of the Slavic language family, Czech and Bulgarian are also characterized by highly inflectional morphology. Czech, like Russian, displays a very rich nominal inflection with as many as 14 declension paradigms. Bulgarian, unlike Russian, is not affected by case distinctions but is characterized by a definiteness suffix.

5.1 Experimental setup

The following parallel corpora are used to train the BNN models:

- English-Russian: WMT13 data (News Commentary and Yandex corpora);
- English-Czech: CzEng 1.0 corpus (Bojar et al., 2012) (Web Pages and News sections);
- English-Bulgarian: a mix of crawled news data, TED talks and Europarl proceedings.

Detailed corpus statistics are given in Table 2. For each language pair, accuracies are measured on a held-out set of 10K parallel sentences.

To prepare the candidate generation function, each dataset is first word-aligned with GIZA++, then a bilingual lexicon with maximum-likelihood probabilities (P_{mle}) is built from the symmetrized alignment. After some frequency and significance pruning,[7] the top 200 translations sorted by $P_{\text{mle}}(t|s) \cdot P_{\text{mle}}(s|t)$ are kept as candidate word translations for each source word in the vocabulary. Word alignments are also used to train the BNN models: each alignment link constitutes a training sample, with no special treatment of unaligned words and 1-to-many alignments.

The context window size k is set to 3 (corresponding to 7-gram) and the dimensionality of source word representations to 100 in all experiments. The number of hidden units in our feedforward neural networks and the target translation embedding size in LBL models are set to 200. All models are trained for 10 iterations with learning rate set to 0.001.

	En-Ru	**En-Cs**	**En-Bg**
Sentences	1M	1M	0.8M
Src. tokens	26.5M	19.2M	19.3M
Trg. tokens	24.7M	16.7M	18.9M
Src. T/T	.0109	.0105	.0051
Trg. T/T	.0247	.0163	.0104

Table 2: BNN training corpora statistics: number of sentences, tokens, and type/token ratio (T/T).

5.2 Word, stem and suffix prediction accuracy

We measure accuracy at top-n, i. e. the number of times the correct translation was in the top n candidates sorted by a model. For each subtask—word, stem and suffix prediction—the BNN model is compared to the context-independent maximum-likelihood baseline $P_{\text{mle}}(t|s)$ on which the PSMT lexical weighting score is based. Note that this is a more realistic baseline than the uniform models sometimes reported in the literature. The oracle corresponds to the percentage of aligned source-target word pairs in the held-out set that are covered by the candidate generation function. Out of the missing links, about 4% is due to lexicon pruning. Results for all three language pairs are presented in Table 3. In this series of experiments, the morphological BNNs utilize unsupervised segmentation models trained on each target language following Lee et al. (2011).[8]

As shown in Table 3, the BNN models outperform the baseline by a large margin in all tasks and languages. In particular, word prediction accuracy at top-1 increases by +6.4%, +24.6% and +9.0% absolute in English-Russian, English-Czech and English-Bulgarian respectively, without the use of any features based on linguistic annotation. While the baseline and oracle differences among languages can be explained by different levels of overlap between training and held-out set, we cannot easily explain why the Czech BNN performance is so much higher. When comparing the

[6] We find that using rectified linear units gives better results than sigmoid and tanh.

[7] Each lexicon is pruned with minimum word frequency 5, minimum source-target word pair frequency 2, minimum log odds ratio 10.

[8] We use the C++ implementation available at `http://groups.csail.mit.edu/rbg/code/morphsyn`

Model	En-Ru	En-Cs	En-Bg
Word prediction (%):			
Baseline	33.0 / 50.1	42.0 / 59.9	47.9 / 66.0
Word BNN	39.4 / 56.6	66.6 / 81.4	56.9 / 74.0
	+6.4 / +6.5	+24.6/+21.5	+9.0 / +8.0
Oracle	79.5	90.2	86.9
Stem prediction (%):			
Baseline	40.7 / 58.2	46.1 / 64.3	51.9 / 70.1
Stem BNN	45.1 / 62.5	66.1 / 81.6	56.7 / 74.4
	+4.4 / +4.3	+20.0/+17.3	+4.8 / +4.3
Suffix prediction (%):			
Baseline	71.2 / 85.6	78.8 / 93.2	81.5 / 92.4
Suffix BNN	77.0 / 89.7	91.9 / 97.4	87.7 / 94.9
	+5.8 / +4.1	+13.1 /+4.2	+6.2 / +2.5

Table 3: BNN prediction accuracy (top-1/top-3) compared to a context-independent maximum-likelihood baseline.

three prediction subtasks, we find that word prediction is the hardest task as expected. Stem prediction accuracies are considerably higher than word prediction accuracies in Russian, but almost equal in the other two languages. Finally, baseline accuracies for suffix prediction are by far the highest, ranging between 71.2% and 81.5%, which is primarily explained by a smaller number of candidates to choose from. Also on this task, the BNN model achieves considerable gains of +5.8%, +13.1% and +6.2% at top-1, without the need of manual feature engineering.

From these figures, it is hard to predict whether word BNNs or morphological BNNs will have a better effect on SMT performance. On one hand, the word-level BNN achieves the highest gain over the MLE baseline. On the other, the stem- and suffix-level BNNs provide two separate scoring functions, whose weights can be directly tuned for translation quality. A preliminary answer to this question is given by the SMT experiments presented in Section 6.

5.3 Effect of word segmentation

This section analyzes the effect of using different segmentation techniques. We consider two supervised tagging methods that produce lemma and inflection tag for each token in a context-sensitive manner: TreeTagger (Sharoff et al., 2008) for Russian and the Morce tagger (Spoustová et al., 2007) for Czech.[9] Finally, we employ the Russian Snowball rule-based stemmer as a light-weight context-insensitive segmentation technique.[10]

[9] Annotation included in the CzEng 1.0 corpus release.

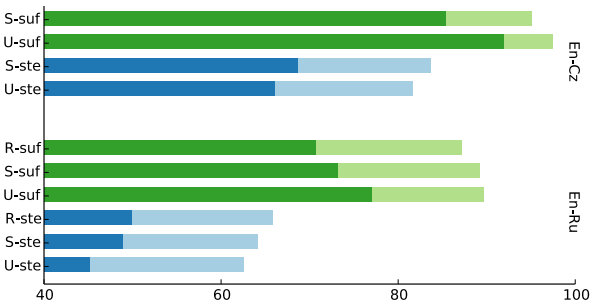


Figure 4: Effect of different word segmentation techniques (U: unsupervised, S: supervised, R: rule-based stemmer) on stem and suffix prediction accuracy. The dark part of each bar stands for top-1, the light one for top-3 accuracy.

As shown in Figure 4, accuracies for both stem and suffix prediction vary noticeably with the segmentation used. However, higher stem accuracies corresponds to lower suffix accuracies and vice versa, which can be mainly due to a general preference of a tool to segment more or less than another. In summary, the unsupervised segmentation methods and the light-weight stemmer appear to perform comparably to the supervised methods.

5.4 Effect of training data size

We examine the predictive power of our models with respect to the size of training data. Table 4 shows the accuracies of stem and suffix models trained on 200K and 1M English-Russian sentence pairs with unsupervised word segmentation. Surprisingly, we observe only a minor loss when we decrease the training data size, which suggests that our models are robust even on a small data set.

# Train sent.	Stem Acc.	Suffix Acc.
1M	45.1 / 62.5	77.0 / 89.7
200K	44.6 / 61.8	75.7 / 88.6

Table 4: Accuracy at top-1/top-3 (%) of stem and suffix BNNs with different training data sizes.

5.5 Fine-grained evaluation

We evaluate the suffix BNN model at the part-of-speech (POS) level. Table 5 provides suffix prediction accuracy per POS for En-Ru. For this analysis, Russian data is segmented by TreeTag-

[10] http://snowball.tartarus.org/algorithms/russian/stemmer.html

ger. Additionally, we report the average number of suffixes per stem given the part-of-speech.

Our results are consistent with the findings of Chahuneau et al. (2013):[11] the prediction of adjectives is more difficult than that of other POS while Russian verb prediction is relatively easier in spite of the higher number of suffixes per stem. These differences reflect the importance of source versus target context features in the prediction of the target inflection: For instance, adjectives agree in gender with the nouns they modify, but this may be only inferred from the target context.

POS	A	V	N	M	P
Acc. (%)	49.6	61.9	62.8	84.5	64.4
$\lvert M_\sigma \rvert$	18.2	18.4	9.2	7.1	13.3

Table 5: Suffix prediction accuracy at top-1 (%), breakdown by category (A: adjectives, V: verbs, N: nouns, M: numerals and P: pronouns). $|M_\sigma|$ denotes the average number of suffixes per stem.

5.6 Neural Network variants

Table 6 shows the stem and suffix accuracies of BNN variants on English-Czech. Although none of the variants outperform our main FFNN architecture, we observe similar performances by the LBL on stem prediction, and by the ConvNet on suffix prediction. This suggests that future work could exploit their additional flexibilities (see Section 4.2) to improve the BNN predictive power. As for the low suffix accuracy by the LBL, it can be explained by the absence of nonlinearity transformation. Nonlinearity is important for the suffix model where the prediction of target suffix μ_j often does not depend linearly on s_i and σ_j. The predictive representation of target stem in the LBL stem model, however, mainly depends on the source representation $\mathbf{r}_{s_i}$ through a position dependent weight matrix $\mathbf{C}_0$. Thus, we observe a smaller accuracy drop in the stem model than in the suffix model. Conversely, the ConvNet performs poorly on stem prediction because it captures the meaning of the whole source context instead of emphasizing the importance of the source word s_i as the main predictor of the target translation t_j.

[11] Chahuneau et al. (2013) report an average accuracy of 63.1% for the prediction of A, V, N, M suffixes. When we train our model on the same dataset (news-commentary) we obtain a comparable result (64.7% vs 63.1%).

Unexpectedly, no improvement is obtained by the use of dropout regularizer (see Section 4.3).

Model	Stem Acc	Suffix Acc
FFNN	66.1 / 81.6	91.9 / 97.4
FFNN+do	64.6 / 81.1	91.5 / 97.5
LBL	63.6 / 79.6	86.4 / 96.4
ConvNet+do	58.6 / 75.6	90.3 / 96.9

Table 6: Accuracies at top-1/top-3 (%) of stem and suffix models. +do indicates dropout instead of L_2 regularizer. FFNN is our main architecture.

6 SMT experiments

While the main objective of this paper is to improve prediction accuracy of word translations, see Section 5, we are also interested in knowing to which extent these improvements carry over within an end-to-end machine translation task. To this end, we integrate our translation prediction models described in Section 4 into our existing English-Russian SMT system.

For each phrase pair matching the input, the phrase BNN score $P_{\text{BNN-p}}$ is computed as follows:

$$P_{\text{BNN-p}}(\tilde{s}, \tilde{t}, a) = \prod_{i=1}^{|\tilde{s}|} \begin{cases} \frac{1}{|\{a_i\}|} \sum_{j \in \{a_i\}} P_{\text{BNN}}(t_j | \mathbf{c}_{s_i}) & \text{if } |\{a_i\}| > 0 \\ P_{\text{mle}}(\text{NULL} | s_i) & \text{otherwise} \end{cases}$$

where a is the word-level alignment of the phrase pair $(\tilde{s}, \tilde{t})$ and $\{a_i\}$ is the set of target positions aligned to s_i. If a source-target link cannot be scored by the BNN model, we give it a P_{BNN} probability of 1 and increment a separate count feature ε. Note that the same phrase pair can get different BNN scores if used in different source side contexts.

Our baseline is an in-house phrase-based (Koehn et al., 2003) statistical machine translation system very similar to Moses (Koehn et al., 2007). All system runs use hierarchical lexicalized reordering (Galley and Manning, 2008; Cherry et al., 2012), distinguishing between monotone, swap, and discontinuous reordering, all with respect to left-to-right and right-to-left decoding. Other features include linear distortion, bidirectional lexical weighting (Koehn et al., 2003), word and phrase penalties, and finally a word-level 5-gram target LM trained on all available monolingual data with modified Kneser-Ney smoothing (Chen and Goodman, 1999). The distortion

Corpus	Lang.	#Sent.	#Tok.
paral.train	EN	1.9M	48.9M
	RU		45.9M
Wiki dict.	EN/RU	508K	–
mono.train	RU	21.0M	390M
WMT2012	EN	3K	64K
WMT2013		3K	56K

Table 7: SMT training and test data statistics. All numbers refer to tokenized, lowercased data.

limit is set to 6 and for each source phrase the top 30 translation candidates are considered. When translating into a morphologically rich language, data sparsity issues in the target language become particularly apparent. To compensate for this we also experiment with a 5-gram suffix-based LM in addition to the surface-based LM (Müller et al., 2012; Bisazza and Monz, 2014).

The BNN models are integrated as additional log-probability feature functions ($\log P_{\text{BNN-p}}$): one feature for the word prediction model or two features for the stem and suffix models respectively, plus the penalty feature ε.

Table 7 shows the data used to train our English-Russian SMT system. The feature weights for all approaches were tuned by using pairwise ranking optimization (Hopkins and May, 2011) on the wmt12 benchmark (Callison-Burch et al., 2012). During tuning, 14 PRO parameter estimation runs are performed in parallel on different samples of the n-best list after each decoder iteration. The weights of the individual PRO runs are then averaged and passed on to the next decoding iteration. Performing weight estimation independently for a number of samples corrects for some of the instability that can be caused by individual samples. The wmt13 set (Bojar et al., 2013) was used for testing. We use approximate randomization (Noreen, 1989) to test for statistically significant differences between runs (Riezler and Maxwell, 2005).

Translation quality is measured with case-insensitive BLEU[%] using one reference translation. As shown in Table 8, statistically significant improvements over the respective baseline (Baseline and Base+suffLM) are marked ▲ at the $p < .01$ level. Integrating our bilingual neural network approach into our SMT system yields small but statistically significant improvements of 0.4 BLEU over a competitive baseline. We can also

SMT system	wmt12 (dev)	wmt13 (test)
Baseline	24.7	18.9
+ stem/suff. BNN	25.1	19.3▲
Base+suffLM	24.5	19.2
+ word BNN	24.5	19.3
+ stem/suff. BNN	24.7	19.6▲

Table 8: Effect of our BNN models on English-Russian translation quality (BLEU[%]).

see that it is beneficial to add a suffix-based language model to the baseline system. The biggest improvement is obtained by combining the suffix-based language model and our BNN approach, yielding 0.7 BLEU over a competitive, state-of-the-art baseline, of which 0.4 BLEU are due to our BNNs. Finally, one can see that the BNNs modeling stems and suffixes separately perform better than a BNN directly predicting fully inflected forms.

To better understand the BNN effect on the SMT system, we analyze the set of phrase pairs that are employed by the decoder to translate each sentence. This set is ranked by the weighted combination of phrase translation and lexical weighting scores, target language model score and, if available, phrase BNN scores. As shown in Table 9, the morphological BNN models have a positive effect on the decoder's lexical search space increasing the recall of reference tokens among the top 1 and 3 phrase translation candidates. The mean reciprocal rank (MRR) also improves from 0.655 to 0.662. Looking at the 1-best SMT output, we observe a slight increase of reference/output recall (50.0% to 50.7%), which is less than the increase we observe for the top 1 translation candidates (57.6% to 59.0%). One possible explanation is that the new, more accurate translation distributions are overruled by other SMT model scores,

Token recall (wmt12):	Baseline	+BNN
reference/MT-search-space [top-1]	57.6%	59.0%
reference/MT-search-space [top-3]	70.7%	70.9%
reference/MT-search-space [top-30]	86.0%	85.0%
reference/MT-search-space [MRR]	0.655	0.662
reference/MT-output	50.0%	50.7%
stem-only reference/MT-output	12.3%	11.5%
of which reachable	11.2%	10.3%

Table 9: Target word coverage analysis of the English-Russian SMT system before and after adding the morphological BNN models.

like the target LM, that are based on traditional maximum-likelihood estimates. While the suffix-based LMs proved beneficial in our experiments, we speculate that higher gains could be obtained by coupling our approach with a morphology-aware neural LM like the one recently presented by Botha and Blunsom (2014).

7 Related work

While most relevant literature has been discussed in earlier sections, the following approaches are particularly related to ours: Minkov et al. (2007) and Toutanova et al. (2008) address target inflection prediction with a log-linear model based on rich morphological and syntactic features. Their model exploits target context and is applied to inflect the output of a stem-based SMT system, whereas our models predict target words (or pairs of stem-suffix) independently and are integrated into decoding. Chahuneau et al. (2013) address the same problem with another feature-rich discriminative model that can be integrated in decoding, like ours, but they also use it to inflect on-the-fly stemmed phrases. It is not clear what part of their SMT improvements is due to the generation of new phrases or to better scoring. Jeong et al. (2010) predict surface word forms in context, similarly to our word BNN, and integrate the scores into the SMT system. Unlike us, they rely on linguistic feature-rich log-linear models to do that. Gimpel and Smith (2008) propose a similar approach to directly predict phrases in context, instead of words.

All those approaches employed features that capture the global structure of source sentences, like dependency relations. By contrast, our models access only local context in the source sentence but they achieve accuracy gains comparably to models that also use global sentence structure.

8 Conclusions

We have proposed a general approach to predict word translations in context using bilingual neural network architectures. Unlike previous NN approaches, we model word, stem and suffix distributions in the target language given context in the source language. Instead of relying on manually engineered features, our models automatically learn abstract word representations and features that are relevant for the modeled task directly from word-aligned parallel data. Our preliminary results with LBL and ConvNet architectures suggest that potential improvement may be achieved by factorizing target representations or by dynamically modeling source context size. Evaluated on three morphologically rich languages, our approach achieves considerable gains in word, stem and suffix accuracy over a context-independent maximum-likelihood baseline. Finally, we have shown that the proposed BNN models can be tightly integrated into a phrase-based SMT system, resulting in small but statistically significant BLEU improvement over a competitive, large-scale English-Russian baseline.

Our analysis shows that the number of correct target words occurring in highly scored phrase translation candidates increases after integrating the morphological BNNs. However, only few of these end up in the 1-best translation output. Future work will investigate the benefits of coupling our BNN models with target language models that also exploit abstract word representations, such as Botha and Blunsom (2014) and Auli et al. (2013).

Acknowledgments

This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project numbers 639.022.213 and 612.001.218. We would like to thank Ekaterina Garmash for helping with the error analysis of the English-Russian translations.

References

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. 2013. Joint language and translation modeling with recurrent neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1044–1054, Seattle, Washington, USA, October.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

Arianna Bisazza and Christof Monz. 2014. Class-based language modeling for translating into morphologically rich languages. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*, pages 1918–1927, Dublin, Ireland.

Ondřej Bojar, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. 2012. The joy of parallelism with czeng

1.0. In *Proceedings of LREC2012*, Istanbul, Turkey, May. ELRA, European Language Resources Association.

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

Jan A. Botha and Phil Blunsom. 2014. Compositional Morphology for Word Representations and Language Modelling. In *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, June.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–51, Montréal, Canada, June. Association for Computational Linguistics.

Victor Chahuneau, Eva Schlinger, Noah A. Smith, and Chris Dyer. 2013. Translating into morphologically rich languages with synthetic phrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1677–1687, Seattle, USA, October.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 4(13):359–393.

Colin Cherry, Robert C. Moore, and Chris Quirk. 2012. On hierarchical re-ordering and permutation parsing for phrase-based decoding. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 200–209, Montréal, Canada, June. Association for Computational Linguistics.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the 25th Annual International Conference on Machine Learning*, volume 12, pages 2493–2537.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 678–683, Sofia, Bulgaria, August.

Nadir Durrani, Alexander Fraser, and Helmut Schmid. 2013. Model with minimal translation units, but decode with phrases. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1–11, Atlanta, Georgia, USA, June.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Morristown, NJ, USA. Association for Computational Linguistics.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2014. Learning continuous phrase representations for translation modeling. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 699–709. Association for Computational Linguistics.

Kevin Gimpel and Noah A. Smith. 2008. Rich source-side context for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 9–17, Columbus, Ohio, USA.

Spence Green and John DeNero. 2012. A class-based agreement model for generating accurately inflected translations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, ACL '12, pages 146–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rejwanul Haque, Sudip Kumar Naskar, Antal Bosch, and Andy Way. 2011. Integrating source-language context into phrase-based statistical machine translation. *Machine Translation*, 25(3):239–285, September.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Yuening Hu, Michael Auli, Qin Gao, and Jianfeng Gao. 2014. Minimum translation modeling with recurrent neural networks. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 20–29, Gothenburg, Sweden, April. Association for Computational Linguistics.

Minwoo Jeong, Kristina Toutanova, Hisami Suzuki, and Chris Quirk. 2010. A discriminative lexicon model for complex morphology. In *The Ninth Conference of the Association for Machine Translation in the Americas*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, USA, October.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665. Association for Computational Linguistics.

Ahmed El Kholy and Nizar Habash. 2012. Translate, predict or generate: Modeling rich morphology in statistical machine translation. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of HLT-NAACL 2003*, pages 127–133, Edmonton, Canada.

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

Philipp Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In Robert E. Frederking and Kathryn B. Taylor, editors, *Proceedings of the 6th Conference of the Association for Machine Translations in the Americas (AMTA 2004)*, pages 115–124.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, J Gauvain, and François Yvon. 2011. Structured output layer neural network language model. In *Proceedings of Proceedings of ICASSP*.

Yoong Keok Lee, Aria Haghighi, and Regina Barzilay. 2011. Modeling syntactic context improves morphological segmentation. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 1–9, Portland, Oregon, USA, June. Association for Computational Linguistics.

Arne Mauser, Saša Hasan, and Hermann Ney. 2009. Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 210–218, Stroudsburg, PA, USA. Association for Computational Linguistics.

Einat Minkov, Kristina Toutanova, and Hisami Suzuki. 2007. Generating complex morphology for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 128–135.

Andriy Mnih and Geoffrey E. Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, pages 641–648, New York, NY, USA.

Thomas Müller, Hinrich Schütze, and Helmut Schmid. 2012. A comparative investigation of morphological language modeling for the languages of the European Union. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 386–395, Montréal, Canada, June. Association for Computational Linguistics.

Eric W. Noreen. 1989. *Computer Intensive Methods for Testing Hypotheses. An Introduction*. Wiley-Interscience.

Chris Quirk and Arul Menezes. 2006. Do we need phrases? challenging the conventional wisdom in statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 9–16, New York City, USA, June. Association for Computational Linguistics.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Holger Schwenk, Daniel Dechelotte, and Jean-Luc Gauvain. 2006. Continuous space language models for statistical machine translation. In *Proceedings of the COLING/ACL 2006 Conference*, pages 723–730, Sydney, Australia, July. Association for Computational Linguistics.

Holger Schwenk. 2012. Continuous space translation models for phrase-based statistical machine translation. In *Proceedings of COLING*.

Serge Sharoff, Mikhail Kopotev, Tomaz Erjavec, Anna Feldman, and Dagmar Divjak. 2008. Designing and evaluating a russian tagset. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Noah A. Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the*

Conference on Empirical Methods in Natural Language Processing, pages 151–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics.

Le Hai Son, Alexandre Allauzen, and François Yvon. 2012. Continuous space translation models with neural networks. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–48, Stroudsburg, PA, USA. Association for Computational Linguistics.

Drahomíra Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec, and Pavel Květoň. 2007. The best of two worlds: Cooperation of statistical and rule-based taggers for czech. In *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing*, pages 67–74, Prague, Czech Republic, June. Association for Computational Linguistics.

Kristina Toutanova, Hisami Suzuki, and Achim Ruopp. 2008. Applying morphology generation models to machine translation. In *Proceedings of the Association for Computational Linguistics*.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with large-scale neural language models improves translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, October.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, USA, October.