An MT Error-driven Discriminative Word Lexicon using Sentence Structure Features

Jan Niehues and Alex Waibel
Institute for Anthropomatics
Karlsruhe Institute of Technology, Germany
firstname.secondname@kit.edu

Abstract

The Discriminative Word Lexicon (DWL) is a maximum-entropy model that predicts the target word probability given the source sentence words. We present two ways to extend a DWL to improve its ability to model the word translation probability in a phrase-based machine translation (PBMT) system. While DWLs are able to model the global source information, they ignore the structure of the source and target sentence. We propose to include this structure by modeling the source sentence as a bag-of-n-grams and features depending on the surrounding target words. Furthermore, as the standard DWL does not get any feedback from the MT system, we change the DWL training process to explicitly focus on addressing MT errors.

By using these methods we are able to improve the translation performance by up to 0.8 BLEU points compared to a system that uses a standard DWL.

1 Introduction

In many state-of-the-art SMT systems, the phrase-based (Koehn et al., 2003) approach is used. In this approach, instead of building the translation by translating word by word, sequences of source and target words, so-called phrase pairs, are used as the basic translation unit. A table of correspondences between source and target phrases forms the translation model. Target language fluency is modeled by a language model storing monolingual n-gram occurrences. A log-linear combination of these main models as well as additional features is used to score the different translation hypotheses. Then the decoder searches for the translation with the highest score.

One problem of this approach is that bilingual context is only modeled within the phrase pairs. Therefore, different approaches to increase the context available during decoding have been presented (Haque et al., 2011; Niehues et al., 2011; Mauser et al., 2009). One promising approach is the Discriminative Word Lexicon (DWL). In this approach, a discriminative model is used to predict the probability of a target word given the words in the source sentence.

In contrast to other models in the phrase-based system, this approach is capable of modeling the translation probability using information from the whole sentence. Thus it is possible to model long-distance dependencies. But the model is not able to use the structure of the sentence, since the source sentence is modeled only as a bag-of-words. Furthermore, the DWL is trained to discriminate between all translation options without knowledge about the other models used in a phrase-based machine translation system such as the translation model, language model etc. In contrast, we try to feedback information about possible errors of the MT system into the DWL. Thereby, the DWLs are able to focus on improving the errors of the other models of an MT system.

We will introduce features that encode information about the source sentence structure. Furthermore, the surrounding target words will also be used in the model to encode information about the target sentence structure. Finally, we incorporate information from the other models into the creation of the training examples. We create the negative training examples using possible errors of the other models.

2 Related Work

Bangalore et al. (2007) presented an approach to machine translation using discriminative lexical selection. Motivated by their results, Mauser et al. (2009) integrated the DWL into the PBMT ap-

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 512–520,
Sofia, Bulgaria, August 8-9, 2013

proach. Thereby, they are able to use global source information.

This was extended by Huck et al. (2010) by a feature selection strategy in order to reduce the number of weights. In Mediani et al. (2011) a first approach to use information about MT errors in the training of DWLs was presented. They select the training examples by using phrase table information also.

The DWLs are related to work that was done in the area of word sense disambiguation (WSD). Carpuat and Wu (2007) presented an approach to disambiguate between different phrases instead of performing the disambiguation at word level.

A different lexical model that uses target side information was presented in Jeong et al. (2010). The focus of this work was to model complex morphology on the target language.

3 Discriminative Word Lexicon

The DWL is a maximum entropy model used to determine the probability of using a target word in the translation. Therefore, we train individual models for every target word. Each model is trained to return the probability of this word given the input sentence.

The input of the model is the source sentence. Therefore, we need to represent the input sentence by features. In this approach this is done by using binary features. We use an indicator feature for every input word. Therefore, the sentence is modeled as a bag-of-words and the order of the words is ignored. More formally, a given source sentence $F = f_1 \dots f_I$ is represented by the features $I(F) = \{i_f(F) : f \in SourceVocabulary\}$:

$$i_f(F) = \begin{cases} 1 & : f \in F \\ 0 & : f \notin F \end{cases} \quad (1)$$

The models are trained on examples generated by the parallel training data. The labels for training the classifier of target word e are defined as follows:

$$label_e(F, E) = \begin{cases} 1 & : e \in E \\ 0 & : e \notin E \end{cases} \quad (2)$$

We used the MegaM Toolkit[1] to train the maximum entropy models. This model approximates the probability $p(e_j|F)$ of a target word e_j given the source sentence F.

[1] http://www.umiacs.umd.edu/ hal/megam/index.html

When we have the probability for every word e_j given the source sentence F, we need to combine these probabilities into a probability of the whole target sentence $E = e_1 \dots e_J$ given F. Making an assumption of independence on the target side as well, the models can be combined to the probability of E given F :

$$p(E|F) = \prod_{e_j \in e} p(e_j|F) \quad (3)$$

In this equation we multiply the probability of one word only once even if the word occurs several times in the sentence. Since we build the target sentence from left to right during decoding, we would need to change the score for this feature only if a new word is added to the hypothesis. If a word is added second time we do not want to change the feature value. In order to keep track of this, additional bookkeeping would be required. But the other models in our translation system will prevent us from using a word too often in any case. Therefore, we approximate the probability of the sentence differently as defined in Equation 4.

$$p(E|F) = \prod_{j=1}^{J} p(e_j|F) \quad (4)$$

In this case we multiply the probabilities of all word occurrences in the target sentence. Therefore, we can calculate the score for every phrase pair before starting with the translation.

4 Modeling Sentence Structure

As mentioned before one main drawback of DWLs is that they do not encode any structural information about the source or target sentence. We incorporated this information with two types of features. First, we tried to encode the information from the source sentence better by using a bag-of-n-grams approach. Secondly, we introduced new features to be able to encode information about the neighboring target words also.

4.1 Source Sentence Structure

In the default approach the sentence is represented as a bag-of-words. This has the advantage that the model can use a quite large context of the whole sentence. In contrast to the IBM models, where the translation probability only depends on the aligned source word, here the translation probability can be influenced by all words in the sentence.

On the other hand, the local context is ignored by the bag-of-words approach. Information about the word order get lost. No information about the previous and next word is available. The problem is illustrated in the example in Figure 1.

Figure 1: *Example for source structural information*

Source:	Die Lehrer wussten nicht, ...
Reference:	The teachers didn't know ...

The German word *Lehrer* (engl. *teacher*) is the same word for singular or plural. It is only possible to distinguish whether singular or plural is meant through the context. This can be determined by the plural article *die*. If only one teacher would be meant, the corresponding article would be *der*.

In order be able to use the DWL to distinguish between these two translations, we need to improve the representation of the input sentence. As shown in the example, it would be helpful to know the order of the words. If we know that the word *die* precedes *Lehrer*, it would be more probable that the word is translated into *teachers* rather than *teacher*.

Therefore, we propose to use a bag-of-n-grams instead of a bag-of-words to represent the input sentence. In this case we will use an indicator feature for every n-gram occurring in the input sentence and not only for every word. This way we are also able to encode the sequence of the words. For the example, we would have the input feature *die_Lehrer*, which would increase the probability of using *teachers* in the translation compared to *teacher*.

By increasing the order of the n-grams, we will also increase the number of features and run into data sparseness problems. Therefore, we used count filtering on the features for higher order n-grams. Furthermore, we combine n-grams of different orders to better handle the data sparseness problem.

4.2 Target Sentence Structure

In the standard DWL approach, the probability of the target word depends only on the source words in the input sentence. But this is a quite rough approximation. In reality, the probability of a target word occurring in the sentence also depends on the other target words in the sentence.

If we look at the word *langsam* (engl. *slow* or *slowly*) in the example sentence in Figure 2, we can only determine the correct translation by using the target context. The word can be translated as *slow* or *slowly* depending on how it is used in the English sentence.

In order to model the translation probability better we need structural information of the target side. For example, if the preceding word on the target side is *be*, the translation will be more probably *slow* than *slowly*.

We encoded the target context of the word by features indicating the preceding or next word. Furthermore, we extend the context to up to three words before and after the word. Therefore the following target features are added to the set of features for the classifier of word e:

$$i_{TC\_e'\_k}(E) = \begin{cases} 1 & : \exists j : e_j = e \wedge e_{j+k} = e' \\ 0 & : else \end{cases} \quad (5)$$

where $k \in \{-1, 1\}$ for a context of one word before and after.

5 Training

Apart from the missing sentence structure the DWL is not able to make use of feedback from the other models in the MT system. We try to incorporate information about possible errors introduced by the other models into the training of the DWL.

The DWL is trained on the parallel data that is available for the task $T = (F_1, E_1), \ldots, (F_M, E_M)$. In order to train it, we need to create positive and negative examples from this data. We will present different approaches to generate the training examples, which differ in the information used for creating the negative examples.

In the original approach, one training example is created for every sentence of the parallel data and for every DWL classifier. If the target word occurs in the sentence, we create a positive example and if not the source sentence is used as a negative example as described in Equation 2. For most words, this results in a very unbalanced set of training examples. Most words will only occur in quite few sentences and therefore, we have mostly negative examples.

Mediani et al. (2011) presented an approach to create the training examples that is driven by looking at possible errors due to the different

Figure 2: *Example for target structural information*

Source:	Die Anerkennung wird langsam sein in den Vereinigten Staaten ...
Reference:	The recognition is going to be slow in the United States, ...

translations in the phrase table (**Phrase pair approach**). Since a translation is generated always using phrase pairs $(\widetilde{f}, \widetilde{e})$ with matching source side, wrong words can only be generated in the translation if the word occurs in the target side words of those matching phrase pairs. Therefore, we can define the possible target vocabulary $TV(F)$ of a source sentence:

$$TV(F) = \{e|\exists(\widetilde{f}, \widetilde{e}) : \widetilde{f} \subseteq F \wedge e \in \widetilde{e}\} \quad (6)$$

As a consequence, we generate a negative training example for one target word only from those training sentences where the word is in the target vocabulary but not in the reference.

$$label_e(F, E) = \begin{cases} 1: & e \in E \\ 0: & e \notin E \wedge e \in TV(F) \end{cases} \quad (7)$$

All training sentences for which the label is not defined are not used in the training of the model for word e. Thereby, not only can we focus the classifiers on improving possible errors made by the phrase table, but also reduce the amount of training examples and therefore the time needed for training dramatically.

In the phrase pair approach we only use information about possible errors of the translation model for generating the negative training examples. But it would be preferable to consider possible errors of the whole MT system instead of only using the phrase table. Some of the errors of the phrase table might already be corrected by the language model. The possible errors of the whole system can be approximated by using the N-Best list.

We first need to translate the whole corpus and save the N-Best list for all sentences $NBEST(F) = \{E'_1 \dots E'_N\}$. Then we can approximate the possible errors of the MT system with the errors that occur in the N-Best list. Therefore, we create a negative example for a target word only if it occurs in the N-Best list and not in the reference. Compared to the phrase pair approach, the only difference is the definition of the target vocabulary:

$$TV(F) = \{e|e \in NBEST(F)\} \quad (8)$$

The disadvantage of the N-Best approach is, of course, that we need to translate the whole corpus. This is quite time consuming, but it can be parallelized.

5.1 Training Examples for Target Features

If we use target features, the creation of the training examples gets more difficult. When using only source features, we can create one example from every training sentence. Even if the word occurs in several phrase pairs or in several entries of the N-Best list, all of them will create the same training example, since the features only depend on the source sentence.

When we use target features, the features of the training example depend also on the target words that occur around the word. Therefore, we can only use the N-Best list approach to create the target features since previous approaches mentioned in the last part do not have the target context information. Furthermore, we can create different examples from the same sentence. If we have, for example, the N-Best list entries *I think ...* and *I believe ..*, we can use the context *think* or the context *believe* for the model of *I*.

In the approach using all target features (**All TF**), we created one training example for every sentence where the word occurs. If we see the word in different target contexts, we create all the features for these contexts and use them in the training example.

$$I(F, E) = max(I(F); I(E); I(E')|E' \in NBEST(F)) \quad (9)$$

The maximum is defined component-wise. So all features, which have in $I(F)$,$I(E)$ or $I(E')$ the value one, also have the value one in $I(F, E)$. If we use the context that was given by the reference, this might not exist in the phrase-based MT system. Therefore, in the next approach (**N-Best TF**), we only used target features from the N-Best list.

$$I(F, E) = max(I(F); I(E')|E' \in NBEST(F)) \quad (10)$$

In both examples, we still have the problem that we can use different contexts in one training ex-

ample. This condition can not happen when applying the DWL model. Therefore, we changed the set of training examples in the separate target features approach (**Separate TF**). We no longer create one training example for every training sentence (F, E), but one for every training sentence N-Best list translation (F, E, E'). We only considered the examples for the classifier of target word e, where e occurs in the N-Best list entry E'. If the word does not occur in any N-Best list entry of a training sentence, but in the reference, we created an additional example $(F, E, "")$. The features of this examples can then be created straight forward as:

$$I((F, E, E')) = max(I(F); I(E')) \qquad (11)$$

If we have seen the word only in the reference, we create an training example without target features. Therefore, we have again a training example which can not happen when using the DWL model. Therefore, we removed these examples in the last method (**Restricted TF**).

6 Experiments

After presenting the different approaches to perform feature and example selection, we will now evaluate them. First, we will give a short overview of the MT system. Then we will give a detailed evaluation on the task of translating German lectures into English and analyze the influence of the presented approaches. Afterwards, we will present overview experiments on the German-to-English and English-to-German translation task of WMT 13 Shared Translation Task.

6.1 System Description

The translation system was trained on the EPPS corpus, NC corpus, the BTEC corpus and TED talks.[2] The data was preprocessed and compound splitting (Koehn and Knight, 2003) was applied for German. Afterwards the discriminative word alignment approach as described in Niehues and Vogel (2008) was applied to generate the alignments between source and target words. The phrase table was built using the scripts from the Moses package (Koehn et al., 2007). A 4-gram language model was trained on the target side of the parallel data using the SRILM toolkit (Stolcke, 2002). In addition we used a bilingual language model as described in Niehues et al. (2011).

[2]http://www.ted.com

Reordering was performed as a preprocessing step using part-of-speech information generated by the TreeTagger (Schmid, 1994). We used the reordering approach described in Rottmann and Vogel (2007) and the extensions presented in Niehues and Kolss (2009) to cover long-range reorderings, which are typical when translating between German and English.

An in-house phrase-based decoder was used to generate the translation hypotheses and the optimization was performed using MERT (Venugopal et al., 2005).

We optimized the weights of the log-linear model on a separate set of TED talks and also used TED talks for testing. The development set consists of 1.7k segments containing 16k words. As test set we used 3.5k segments containing 31k words. We will refer to this system as System 1.

In order to show the influence of the approaches better, we evaluated them also in a second system. In addition to the models used in the first system we performed a log-linear language model and phrase table adaptation as described in Niehues and Waibel (2012). To this system we refer as *System 2* in the following experiments.

6.2 German - English TED Experiments

6.2.1 Source Features

In a first set of experiments, we analyzed the different types of source structure features described in Section 4.1. In all the experiments, we generate the negative training examples using the candidate translations generated by the phrase pairs. The results can be found in Table 1.

First, we added the unigram DWL to the baseline system. The higher improvements for the System 1 is due to the fact that the DWL is only trained on the TED corpus and therefore also performs some level of domain adaptation. This is more important for the System 1, since System 2 is already adapted to the TED domain.

If we use features based on bigrams instead of unigrams, the number of features increases by a factor of eight. Furthermore, in both cases the translation quality drops. Especially for System 1, we have a significant drop in the BLEU score of the test set by 0.6 BLEU points. One problem might be that most of the bigrams occur quite rarely and therefore, we have a problem of data sparseness and generalization.

If we combine the features of unigram and bi-

Table 1: *Experiments using different source features*

System	FeatureSize	System 1		System 2	
		Dev	Test	Dev	Test
Baseline	0	26.32	24.24	28.40	25.89
Unigram	40k	27.46	25.56	28.58	26.15
Bigram	319k	27.34	24.92	28.53	25.82
Uni+bigram	359k	27.69	25.55	28.66	26.51
+ Count filter 2	122k	27.75	25.71	28.75	26.74
+ Count filter 5	63k	27.81	25.67	28.72	26.81
+ Trigram	77k	27.76	25.76	28.82	26.94

gram features, for System 1, we get an improvement of 0.2 BLEU points on the development data and the same translation quality on the test data as the baseline DWL system using only unigrams. For System 2, we can improve by 0.1 on the development data and 0.4 on the test data. So we can get a first improvement using these additional source features, but the number of features increased by a factor of nine.

In order to decrease the number of features again, we applied count filtering to the bigram features. In a first experiment we only used the bigram features that occur at least twice. This reduced the number of features dramatically by a factor of three. Furthermore, this even improved the translation quality. In both systems we could improve the translation quality by 0.2 BLEU points. So it seems to be quite important to add only the relevant bigram features.

If we use a minimum occurrence of five for the bigram features, we can even decrease the number of features further by a factor of two without losing any translation performance.

Finally, we added the trigram features. For these features we applied count filtering of five. For System 1, the translation quality stays the same, but for System 2 we can improve the translation quality by additional 0.2 BLEU points.

In summary, we could improve the translation quality by 0.2 for the System 1 and 0.8 BLEU points for the System 2 on the test set. Due to the count filtering, this is achieved by only using less than twice as many features.

6.3 Training Examples

In a next step we analyzed the different example selection approaches. The results are summarized in Table 2. In these experiments we used the source features using unigrams, bigrams and trigrams with count filtering in all experiments.

In the first experiment, we used the original approach to create the training examples. In this case, all sentences where the word does not occur in the reference generate negative examples. In our setup, we needed 8,461 DWL models to translate the development and test data. These are all target words that occur in phrase pairs that can be used to translate the development or test set.

In each of approaches we have 0.75M positive examples for these models. In the original approach, we have 428M negative examples. So in this case the number of positive and negative examples is very unbalanced. This training data leads to models with a total of 659M feature weights.

If we use the target side of the phrase pairs to generate our training examples, we dramatically reduce the number of negative training examples. In this case only 5M negative training examples are generated. The size of the models is reduced dramatically to 38M weights. Furthermore, we could improve the translation quality by 0.3 BLEU points on both System 1 and System 2.

If we use the 300-Best lists produced by System 1 to generate the training examples, we can reduce the model size further. This approach leads to models only half the size of the phrase pairs approach using only 1.59M negative examples. Furthermore, for System 1 the translation quality can be improved further to 25.87 BLEU points. For System 2 the BLEU score on the development data increases, but the score on the test sets drops by 0.4 BLEU points.

In the next experiment we used the N-Best lists generated by System 2. The results are shown in the line *N-Best list 2*. In this case, the model size is slightly reduced further. And on the adapted system a similar performance is achieved. But for

Table 2: *Experiments using different methods to create training examples*

System	#weight	#neg. Examples	System 1		System 2	
			Dev	Test	Dev	Test
Original Approach	659 M	428 M	27.39	25.44	28.64	26.63
Phrase pairs	38 M	5.26 M	27.76	25.76	28.82	26.94
N-Best list 1	16 M	1.59 M	27.93	25.87	29.07	26.57
N-Best list 2	11 M	1.22 M	27.46	25.37	28.79	26.59
N-Best list 1 nonUnique	16 M	1.41M	27.99	25.97	29.07	26.65

System 1 the performance of this approach drops.

Consequently, it seems to be fine to use an N-Best list of a more general system to generate the negative examples. But the N-Best list should not stem from an adapted system.

Finally, the phrase table was trained on the same corpus as the one that was used to generate the N-Best lists for DWL training. Since we have seen the data before, longer phrases can be used than in a real test scenario. To compensate partly for that, we removed all phrase pairs that occur only once in the phrase table. The results are shown in the last line. This approach could slightly improve the translation quality leading to a BLEU score of 25.97 for System 1 and 26.65 for the System 2.

6.4 Target Features

After evaluating the different approaches to generate the negative examples, we also evaluated the different approaches for the target features. The results are summarized in Table 3. In all these experiments we use the training examples generated by the N-Best list of System 1 using the phrase table without unique phrase pairs.

First, we tested the four different methods using a context of one word before and one word after the word.

In the experiments the first two methods, All TF and N-Best TF , perform worse than the last two approaches, Separate TF and Restricted TF. So it seems to be important to have realistic examples and not to mix different target contexts in one example. The Separate and Restricted approach perform similarly well. In both cases the performance can be improved slightly by using a context of three words before and after instead of using only one word.

If we look at the model size, the number of weights increases from 16M to 17M, when using a context of one word and to 21M using a context of three words.

If we compare the results to the systems using no target features in the first row, no or only slight improvements can be achieved. One reason might be that the morphology of English is not very complex and therefore, the target context is not as important to determine the correct translation.

6.4.1 Overview

In Table 4, we give an overview of the results using the different extensions to DWLs given in this paper. The baseline system does not use any DWL at all. If we use a DWL using only bag-of-words features and the training examples from the phrase pairs, we can improve by 1.3 BLEU points on System 1 and 0.3 BLEU points on System 2.

By adding the source-context features, the first system can be improved by 0.2 BLEU points and the second one by 0.8 BLEU points. If we use the training examples from the N-Best list instead of using the ones from the phrase table, we improve by 0.2 on System 1, but perform 0.3 worse on System 2. Adding the target context features does not improve System 1, but System 2 can be improved by 0.3 BLEU points. This system results in the best average performance. Compared to the baseline system with DWLs, we can improve by 0.4 and 0.8 BLEU points, respectively.

Table 4: *Overview of results for TED lectures*

System	System 1		System 2	
	Dev	Test	Dev	Test
Baseline	26.32	24.24	28.40	25.89
DWL	27.46	25.56	28.58	26.15
sourceContext	27.76	25.76	28.82	26.94
N-Best	27.99	25.97	29.07	26.65
TargetContext	28.15	25.91	29.12	26.90

6.5 German - English WMT 13 Experiments

In addition to the experiments on the TED data, we also tested the models in the systems for the

Table 3: *Experiments using different target features*

System	Context	System 1		System 2	
		Dev	Test	Dev	Test
No Target Features	0-0	27.99	25.97	29.07	26.65
All TF	1-1	27.80	25.48	28.80	26.38
N-Best TF	1-1	27.99	25.74	28.86	26.37
Separate TF	1-1	28.06	25.81	28.98	26.80
Restricted TF	1-1	28.13	25.84	28.94	26.68
Separate TF	3-3	27.87	25.90	28.99	26.75
Restricted TF	3-3	28.15	25.91	29.12	26.90

WMT 2013. The systems are similar to the one used before, but were trained on all available training data and use additional models. The systems were tested on newstest2012. The results for German to English are summarized in Table 5. In this case the DWLs were trained on the EPPS and the NC corpus. Since the corpora are bigger, we perform an additional weight filtering on the models.

The baseline system uses already a DWL trained with the bag-of-words features and the training examples were created using the phrase table. If we add the bag-of-n-grams features up to a n-gram length of 3, we cannot improve the translation quality on this task. But by additionally generating the negative training examples using the 300-Best list, we can improve this system by 0.2 BLEU points.

Table 5: *Experiments on German to English WMT 2013*

System	Dev	Test
Unigram DWL	25.79	24.36
+ Bag-of-n-gram	25.85	24.33
+ N-Best	25.84	24.52

6.6 English - German WMT 13 Experiments

We also tested the approach also on the reverse direction. Since the German morphology is much more complex than the English one, we hope that in this case the target features can help more. The results for this task are shown in Table 6. Here, the baseline system again already uses DWLs. If we add the bag-of-n-grams features and generate the training examples from the 300-Best list, we can again slightly improve the translation quality. In this case we can improve the translation quality by additional 0.1 BLEU points by adding the target features. This leads to an overall improvement by nearly 0.2 BLEU points.

Table 6: *Experiments on English to German WMT 2013*

System	Dev	Test
unigram DWL	16.97	17.41
+ Bag-of-n-gram	16.89	17.45
+ N-Best	17.10	17.47
+ Target Features	17.08	17.58

7 Conclusion

Discriminative Word Lexica have been recently used in several translation systems and have shown to improve the translation quality. In this work, we extended the approach to improve its modeling of the translation process.

First, we added features which represent the structure of the sentence better. By using bag-of-n-grams features instead of bag-of-words features, we are able to encode the order of the source sentence. Furthermore, we use features for the surrounding target words to also model the target context of the word. In addition, we tried to train the DWLs in a way that they help to address possible errors of the MT system by feeding information from the MT system back into the generation of the negative training examples. Thereby, we could reduce the size of the models and improve the translation quality. Overall, we were able to improve the translation quality on three different tasks in two different translation directions. Improvements of up to 0.8 BLEU points could be achieved.

8 Acknowledgements

This work was partly achieved as part of the Quaero Programme, funded by OSEO, French State agency for innovation. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287658.

References

S. Bangalore, P. Haffner, and S. Kanthak. 2007. Statistical Machine Translation through Global Lexical Selection and Sentence Reconstruction. In *Annual Meeting-Association for Computational Linguistics*, volume 45, page 152.

M. Carpuat and D. Wu. 2007. Improving Statistical Machine Translation using Word Sense Disambiguation. In *In The 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

R. Haque, S.K. Naskar, A. Bosch, and A. Way. 2011. Integrating source-language context into phrase-based statistical machine translation. *Machine Translation*, 25(3):239–285.

M. Huck, M. Ratajczak, P. Lehnen, and H. Ney. 2010. A Comparison of Various Types of Extended Lexicon Models for Statistical Machine Translation. In *Proc. of the Conf. of the Assoc. for Machine Translation in the Americas (AMTA)*.

M. Jeong, K. Toutanova, H. Suzuki, and C. Quirk. 2010. A Discriminative Lexicon Model for Complex Morphology. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*.

P. Koehn and K. Knight. 2003. Empirical Methods for Compound Splitting. In *EACL*, Budapest, Hungary.

P. Koehn, F.J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Edmonton, Canada.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Demonstration Session*, Prague, Czech Republic.

A. Mauser, S. Hasan, and H. 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, Singapore.

M. Mediani, E. Cho, J. Niehues, T. Herrmann, and A. Waibel. 2011. The KIT English-French Translation Systems for IWSLT 2011. *Proceedings of the eight International Workshop on Spoken Language Translation (IWSLT)*.

Jan Niehues and Mutsin Kolss. 2009. A POS-Based Model for Long-Range Reorderings in SMT. *Fourth Workshop on Statistical Machine Translation (WMT 2009), Athens, Greece*.

J. Niehues and S. Vogel. 2008. Discriminative Word Alignment via Alignment Matrix Modeling. *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 18–25.

J. Niehues and A. Waibel. 2012. Detailed Analysis of different Strategies for Phrase Table Adaptation in SMT. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*.

J. Niehues, T. Herrmann, S. Vogel, and A. Waibel. 2011. Wider Context by Using Bilingual Language Models in Machine Translation. *Sixth Workshop on Statistical Machine Translation (WMT 2011), Edinburgh, UK*.

K. Rottmann and S. Vogel. 2007. Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model. In *TMI*, Skövde, Sweden.

H. Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.

A. Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Icslp*, Denver, Colorado, USA.

A. Venugopal, A. Zollman, and A. Waibel. 2005. Training and Evaluation Error Minimization Rules for Statistical Machine Translation. In *Workshop on Data-drive Machine Translation and Beyond (WPT-05)*, Ann Arbor, MI.