Using Wikipedia to Translate Domain-specific Terms in SMT

Jan Niehues and Alex Waibel

Institute for Anthropomatics
Karlsruhe Institute of Technology, Germany
firstname.secondname@kit.edu

Abstract

When building a university lecture translation system, one important step is to adapt it to the target domain. One problem in this adaptation task is to acquire translations for domain specific terms. In this approach we tried to get these translations from Wikipedia, which provides articles on very specific topics in many different languages. To extract translations for the domain specific terms, we used the interlanguage links of Wikipedia .

We analyzed different methods to integrate this corpus into our system and explored methods to disambiguate between different translations by using the text of the articles. In addition, we developed methods to handle different morphological forms of the specific terms in morphologically rich input languages like German. The results show that the number of out-of-vocabulary (OOV) words could be reduced by 50% on computer science lectures and the translation quality could be improved by more than 1 BLEU point.

1. Introduction

Statistical machine translation (SMT) is currently the most promising approach to machine translation of large vocabulary tasks. The approach was first presented in [1] and has been used in many translation systems since then.

It performs very well if the training and test domains are quite similar, but there are still challenges in porting an MT system to new domains. In contrast to many other machine learning tasks, this is not only a problem of adapting our model in a way that it selects different hypotheses. Also we need to learn new translations for terms that are specific to a domain.

For example, if we go from one domain to another we have to learn that we have to translate the German word *Bank* not as *bench* but as the financial institution *bank*. But in addition, there are domain specific terms with no translations at all in a general MT system. For example, if we want to translate computer science lectures, we need also to learn translations for terms such as *sampling* or *quantisation*.

This knowledge can only be learned from bilingual resources, which are rare especially for uncommon domains. In the default machine translation setup, we learn the translations for phrases from parallel corpora. But in many domains, there is no parallel data available from which translation pairs could be learned. Another possibility to acquire bilingual knowledge is to integrate word lists or lexica into the SMT framework. But in this case additional problems on the integration of the resource into the general SMT framework arise.

In this work we will focus on the integration of a word list extracted from Wikipedia into an SMT system. Wikipedia is nowadays the world's biggest knowledge source. The English Wikipedia contains more than 3.7 million articles and localized versions exist in around 270 languages. Although the articles are not parallel, there are links between articles about the same topic in different languages. Therefore, the titles of the articles can be considered to be a parallel word list.

This word list contains quite specific terms from a wide range of domains. For example, we find translations for the above mentioned terms in computer science as well as, for example, translations for *Blende* (engl. *aperture*) in the domain of photography.

Since we do not have parallel text as normal, new problems arise on how to integrate the knowledge. We need to disambiguate between different translations provided by the Wikipedia titles. Furthermore, often only the lemma of a word is used in the title. But especially for German, many different morphological forms exist, which occur in the documents to be translated but not in the Wikipedia titles. Therefore, we explore ways to also translate morphological variants of a given term.

The paper is structured as follows: First, we will review related work. Afterwards, in Section 3, we will describe how we extract the bilingual information from Wikipedia and integrate it into the SMT framework. Then we will describe an approach to disambiguate between different translations for one word. In Section 5 we describe the generation of morphological operations that allow us to handle different morphological forms of the domain specific terms. In the end, we will evaluate the approach and close with a conclusion.

2. Related Work

To adapt an SMT system towards a new domain, different problems have to be solved. One important question is to find translations for domain specific terms. The other main direction of research is to adapt the parameters of the proba-

bilistic models.

There have been several approaches to acquire domain specific vocabulary. [2] used canonical correlation analysis to mine unseen words in comparable data. They used different approaches to integrate the new found translations into their SMT system and could show improvements on 4 different domains in German to English and French to English translation. Another approach to extract translations for rare words from comparable corpora was presented in [3]. In [4] domain specific vocabulary was acquired by using a bilingual dictionary. Different ways to integrate the vocabulary were investigated. They also performed research on adapting the parameters with in-domain monolingual data by linear and log-linear combinations of the models.

The approaches to adapt the parameters were inspired by similar approaches in speech recognition ([5]). Among them approaches using only monolingual data can be distinguished from those using parallel in-domain data.

If only monolingual data is used either only the language model is adapted or the translation model is adapted using synthetic parallel data which is generated by translating the monolingual data using a baseline MT system. This is for example done in [6], [7] and [8]. In this case, of course, no new translation options can be learned.

In the other case, where also parallel in-domain data is available, authors tried different linear and log-linear combinations of the in-domain and out-of-domain data ([9], [10]). Others could improve the adaptation by using a discriminative approach to select the adaptation weights ([11], [12]).

Another direction of research is to find in-domain documents, for example by using cross-lingual information retrieval techniques ([13]).

Wikipedia has already been shown to be a valuable resource for natural language processing. For example, Erdmann et al. proposed to extract bilingual terms from the Wikipedia titles ([14]) or Yu et. al. use terms from a comparable corpus created from the Wikipedia articles using the inter-language links ([15]).

There have also been several attempts to model the morphology more explicitly in machine translation. Toutanova et al. ([16]) showed improvements when translating into morphologically rich languages by integrating an additional mophological model using an maximum entropy model introduced by Minkov et al. ([17]). Bojar and Tamachyna tried to handle morphological problems by using reverse self training ([18]).

Macherey et al. ([19]) tried to learn morphological operations for parts of the translation system. They could improve compound splitting as done in a machine translation system by learning morphological operations for the compound parts [19].

3. Lexicon creation

Wikipedia is a multilingual encyclopedia containing articles about the same topics in different languages. To be able to extract bilingual terms the so-called inter-language links are very important. They link pages in different languages about the same topic. Using these links we can align the articles in source and target language. Although the articles are no translations of each other and cannot be used directly in the translation system, the titles themselves tend to be translations of each other.

Therefore, the first step to create a corpus of translations of wikipedia titles is to extract the links for the articles and generate the alignment between the articles. As a result, we get two alignments, since there are links in the source language article to a target language article and links in the target language article to a source language article.

The next step in generating the parallel corpus is to symmetrize this alignment. Due to different reasons, it is not always the case that there is a link in both directions. Sometimes a source language article is aligned to a target language article, but this one is not aligned to any source language article or it is aligned to a different source language article. The main reason is that both articles are not about the same topic, but only closely related. In some cases, the link directs only to a paragraph of another article, which is itself about a more general topic. Another option is that the article is directly linked to a more general article since there is no equivalent one in the other language.

In contrast to other parallel data, most phrases occur only once in the corpus. Therefore, we cannot calculate reliable statistics and consequently it is even more important that the corpus is of high quality. Therefore we use the intersection of both alignments in order to get only high quality translations.

3.1. Disambiguation Pages

The main problem of creating this corpus are the word ambiguities. The German word *Bank* translates to *bank*, if the financial institution is meant and if the furniture is meant, the translation is *bench*.

In Wikipedia, this is modeled in the following way: There is a disambiguation page for *Bank* with links to the different articles. And then there are separate articles, one for the financial institution and one for the furniture. Typically, the disambiguation page in one language for *bank* is linked to the disambiguation page in the other language. The articles give us helpful translations, but the links between the disambiguation pages may be misleading. To avoid problems with these links and to acquire a high quality corpus, we ignore all inter-language links to disambiguation pages. We only use the inter-language links of the articles, the disambiguation page is linking to.

3.2. Integration

After acquiring the corpus, we can apply the same preprocessing as for the parallel training corpus of the translation system. Since in the titles often all content words are written in upper case, we case each word as it is most often cased in

the wikipedia corpus. The baseline approach is to use it as a lexicon. Therefore, every pair of titles is used as an additional phrase pair in the phrase table.

By integrating the data in this way, problems with the titles of the ambiguous expressions occur. For example the title for the bench in German is *Bank (Möbel)* (in English: *bench (furniture)*). During translation we will of course only find the word alone without the additional domain information in parenthesis. Therefore, this phrase pair will nearly never match. To bypass this problem, we tried a second approach using the corpus in the same way as a normal training corpus to extend the phrase table. We trained a Giza++ Alignment on the corpus and then performed phrase extraction. Afterwards, we added the additional phrase pairs to the general phrase table.

4. Article disambiguation

As mentioned before, for ambiguous words there are separate articles for each meaning. A special case are terms that are also surnames of famous persons. For example, the German word for *cone* is *Kegel*. But there are also different articles about persons named *Kegel*.

The default approach in statistical machine translation is to calculate statistics about the translation probabilities. But in our case, this would not be very helpful. When there is more than one person called *Kegel*, the translation of *Kegel* into *Kegel* gets a high probability.

In our setup we can use additional information for every entry in our parallel Wikipedia title corpus. In addition to the translation of the title, we can also make use of the articles themselves. We can compare the source article to our test set. If the article is similar to the test set, the probability that the translation of the title will give us a good translation is higher than the translation extracted from an article about a completely different topic. Therefore, we tried to use the similarity between the article and the document to be translated as an additional feature.

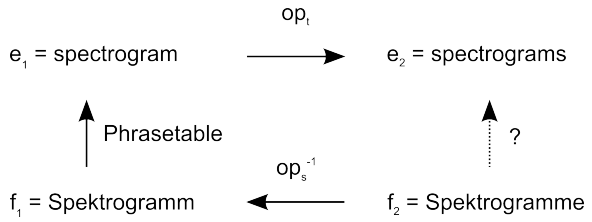
In our example, if we want to translate a computer science lecture, the input should be more similar to an article about a cone than to one about an author or musician called *Kegel*.

To measure the similarity we first represent the Wikipedia article as well as the test document in the TFIDF vector space. Then we calculate the cosine similarity to measure the similarity between the article and the document that needs to be translated. Afterwards, for every source word, the translation from the article with the highest similarity is chosen.

5. Quasi-Morphological Operations

Since we are using only the titles of the articles and not the articles themselves, in most cases the words occur only as lemmas. In contrast, in the test set we also see other morphological forms like genitive or plural form. This is especially problematic when using languages like German as input language, which are highly inflective.

Figure 1: Quasi-morphological operations



5.1. Motivation

In the case of German computer science lectures, our system might be able to translate the German word *Spektrogramm* (engl. *spectrogram*), but not the plural form *Spektrogramme*. Furthermore, if we can translate *trigonometrische Funktion* using the additional knowledge source, we are not necessarily able to translate *von trigonometrischen Funktionen* because the titles of Wikipedia articles usually contain only the lemma, as it is the case for most encyclopedias.

To address this problem, we try to automaticly learn rules how words can be modified. If we look at the first example, we would like the system to learn the following rule. If an *"e"* is appended to the German word, as it is done when creating the Plural form of *Spektogramm*, we need to add an *"s"* to the end of the English word in order to perform the same morphological word transformation. Or in the second example adding an *"n"* to the German adjective represents its transformation into the dative case. For the English equivalent, no transformation needs to be applied since cases do not have a distinct surface word form.

Depending on the part-of-speech, number, gender or case of the involved words, the same operation on the source side does not necessarily correspond to the same operation on the target side.

To account for this ambiguity, we try to use additional information for selecting the target operation. First, we should not generate target words that do not exist. Here, we have the advantage that we can use monolingual data to determine whether a word exists. Furthermore, we use the ending of the source and target word to determine which pair of operations should be used.

An advantage when translating from German to English, i.e. from a morphologically complex to a less complex language, is that many alternative word forms in German map to the same English words. This means when generating the English target word we can ignore the target context in many cases. In contrast, when translating in the other direction the quasi-morphological operations proposed here would need additional context information to decide for one of the possible German word forms.

5.2. Approach

We assume we have two source words f_1 and f_2. These two words are in some way similar, and there is an operation op_s, that transforms f_1 into f_2. Furthermore, we have a target word e_1 which is a translation of f_1. The goal is to find a target word e_2, which is a translation of f_2. To achieve that we try to find an operation op_t given the source operation op_s and the target word e_1. Then we can generate the target word e_2 by:

$$e_2 = op_t(e_1) \tag{1}$$

As shown in Figure 1 the complete procedure can be modelled in the following way. To get a translation for an unknown word f_2, we will apply the inverse source operation on it to get f_1. Then we translate f_1 and apply the target operation on e_1:

$$e_2 = trans(f_2) = op_t(trans(op_s^{-1}(f_2))) \tag{2}$$

As a result, we obtain e_2, a translation of f_2 which we were not able to translate before.

If we look at the example above that means that we have the source words f_1 = *Spektrogramm* and f_2 = *Spektrogramme* and the source operation to append and *"e"* ("" $\rightarrow$ "e"). The target word f_1 would be *spectrogram*. To get the correct translation for f_2, we need to learn that the best target operation op_t in this case is to append an *"s"*("" $\rightarrow$ "s"). Then we can apply the target operation on *spectrogram* and get the correct translation e_2 = *spectrograms*.

To be able to apply the model to our data, we need to define a set of source and target operations as well as a model for the target operations. In our experiments, we tried two different types of operations.

5.3. Operations

The first type of operations are simple replacement operations as described before. These operations allow to replace the ending letters of one word by some other letters. The beginning of the word will not be changed. We will refer to the beginning of the word as word stem. We further restrict the operations in two ways to prevent the model to replace the whole word. First, in German as well as in English in many cases only very few letters are changed at the end of the word. Therefore, we only allow the model to replace up to n letters by at most n other letters. Second, we want to prevent that the whole word is replaced. Therefore, we restrict the word stem to be at least m letters long. Consequently, the first m letters of the word cannot be changed. In some inital experiments, $n = 3$ and $m = 4$ lead to resonable rules. In the remaining of the paper we use these values.

We only use operations where the first letter of both sides is different. As a result, there is only one operation to change e_1 into e_2. In the example, the operation ("m" $\rightarrow$ "me") would not be allowed.

As described in the previous section, if we use this type of operations, there are often different possible target operations for one source operation. Therefore, we also learn operations which are restricted to the ending of the stem. For example, we could learn the operation ("" $\rightarrow$ "e") given that the word ends with an *"m"*. In our experiments, we restrict the model to endings up to a length of five letters.

In this case different operations to convert e_1 to e_2 exist. We can use the operation ("" $\rightarrow$ "e") as well as the same operation given that the source word stem ends with an *"m"* and the target word stem ends with *"am"*.

5.4. Features

After defining the operations, we need to assign a rank to the target operations given a source operation and a target word:

$$R(op_t|op_s, e_1) \tag{3}$$

Then we can select the best ranked target operation and apply it to e_1 to get a translation for f_2. We used different features to determine the ranking.

First, we want to make sure that the operation op_t applied to e_1 generates a valid target language word. Of course, in most cases it will not occur in the parallel corpus or in the titles of the wikipedia articles, otherwise we could translate f_2 in the first place. But we can also use the complete articles of wikipedia on the target side. If the word is a valid word in the target language, it will probably occur in the text. Therefore, we use a feature that indicates whether the generated word is in the target wikipedia corpus.

Furthermore, a target operation that often coincides with a given source operation should be better than one that is rarely used together with the source operation. We therefore look at pairs of entries in the lexicon and count in how many of them the source operation can be applied to the source side and the target operation can be applied to the target side. This count is then used as an additional feature. In the case of the operations with restricted ending, we used an indication feature that the operation has at least a count of n. In our case, we performed several experiments and $n = 3$ lead to the best score on the development data. Therefore, we use that threshold for all experiments.

We calculate additional features, if we apply operations that are restricted to the ending of the word stem. It is better to use operations with more context. Therefore, we use the length of ending in the source operations as well as the one in the target operations as separate features.

For both types of operations we used a product of these features to rank them. For the first type we multiplied the indication feature for a valid word with the count feature. The second type of operations is ranked by the product of the valid feature, the count $> n$ feature and both context features. The features and the corresponding operation types are summarized in Table 1.

5.5. Training

To train a model for the morphological operations we need tofind a set of candidate operations and then the associated

Table 1: *Features for Operations*

Feature \ Operation	Type 1	Type 2
valid Word	x	x
count feature	x	
count $> n$		x
source context length		x
target context length		x

feature values to be able to rank them. For that we use a word aligned corpus. We used the parallel training corpus together with the automatically generated word alignment. From this corpus, we extracted all source and target words that are aligned to each other to build a lexicon.

In the next step we check if a source operation exists that changes one source word to another source word and a target operation that can be applied to the target words. If this is the case, we extract the pair of source and target operations. This provides us with the source and target operations as well as the coocurrence counts of these operations.

5.6. Integration

After being able to find a translation for OOV words, we need to integrate the approach into the phrase-based machine translation system.

We will only use the proposed method for OOVs and will not try to improve translations of words that the baseline system already covers. Therefore, in a first step we extract the OOVs from the source text. Since we want to make use of phrase pairs and not only do word translation for the OOVs, we try to find phrase pairs in the original phrase table that contain a word that is similar to the OOV word.

In the next steps we look for phrase pairs, for which a source operation op_s exists that changes one of the source words f_1 into the OOV word f_2. Since we need to apply a target operation to one word on the target side of the phrase pair, we only consider phrase pairs, where f_1 is aligned to one of the target words of the phrase containing e_1.

If a target operation exists given f_1 and op_s, we select the one with the highest rank. Then we generate a new phrase pair by applying op_s to f_1 and op_t to e_1.

We do not change the features of the phrase pairs. But they are anyway not directly competing with the unchanged phrase pairs, since there is no phrase pair that exactly matches for f_2.

6. Results

We evaluated the described approach to integrate data from Wikipedia into a translation system on the task of translating German computer science lectures into English. The baseline system is described in detail below.

In addition to the improvements measured by automatic metrics we analyzed the number of the OOV words. Therefore, we checked the number of OOV words as well as the number of occurrences. Furthermore, there are several OOV words like names, which are the same in source and target. Since they need not be translated but can be just passed on to the translation, we calculated also a modified number of OOVs where all OOVs which also occur in the reference translation of the sentence are ignored.

6.1. System description

The translation system was trained on the European Parliament corpus, News Commentary corpus and the BTEC corpus. The data was preprocessed and compound splitting was applied. Afterwards the discriminative word alignment approach as described in [20] was applied to generate the alignments between source and target words. The phrase table was built using the scripts from the Moses package [21]. The language model was trained on the target side of the parallel data.

Reordering was performed as a preprocessing step using POS information from the TreeTagger [22]. We used the reordering approach described in [23] together with the extensions presented in [24] 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 MER training. When adding the Wikipedia data no new optimizations were performed.

We used transcribed university lectures from the computer science department as development and test data. Each set contains around 30K words.

6.2. Integration

The results for the baseline system as well as the results for both methods (Lexicon and Corpus) to integrate the data described in Section 3.2 are summarized in Table 2. As it can be seen, the BLEU score could be improved in both cases. The best results could be achieved when using the titles as a parallel corpus and extracting phrase pairs from it. In this case an improvement of 1 BLEU point on Dev and 0.5 BLEU points on the test data can be observed.

By using phrase pairs and not only the title as a whole, a lot more OOV words can be translated. If we want to translate the German word *Abtastung* (engl. *sampling*), this is only possible by using the phrase extraction, since the article is called *Abtastung (Signalverarbeitung)* (engl. *sampling (signal processing)*). Although we are only able to translate 15% of the OOVs of the baseline system, this leads to translation of 45% of the occurrences of OOVs and 39% of the modified OOVs. For the test data this difference is not as big, but the general tendency is similar.

Table 2: *Integration of Wikipedia data*

System	Dev					Test				
	BLEU	OOV		mod. OOV		BLEU	OOV		mod. OOV	
		total	unique	total	unique		total	unique	total	unique
Baseline	25.37	1095	378	891	329	22.72	1437	624	1150	521
Lexicon	25.67	992	359	801	312	23.01	1236	584	958	483
Corpus	26.43	601	315	542	279	23.18	1007	501	825	429

6.3. TFIDF

For the development and test set we calculated the cosine similarity of the TFIDF vectors to all the wikipedia articles. We then used this score as in addition to the four default scores for the phrase pairs. When a phrase pair was extracted from different articles, the pair is kept only once and receives the similarity score of the most similar article. When calculating the TFIDF vector, we ignored the most frequent 10K words.

As shown in Table 3 using the TFIDF score of the article as an additional feature did not improve the translation quality as measured in BLEU. However, in some cases the translation can be improved. For example, the German word *Kegel*(engl. *cone*) is no longer translated into *Kegel* because of an article about a person named *Kegel* but correctly to *cone* because due to the similarity check only the translation into *cone* is kept in the phrase table. But these are only a few examples. In some cases choosing a phrase originating from a more similar article over the phrase with the highest probability even leads to a wrong translation, because of an erroneous alignment between the words in the wikipedia titles.

Table 3: *Using TFIDF*

System	Dev	Test
Baseline	25.37	22.72
Corpus	26.43	23.18
TFIDF	26.43	23.19

6.4. Quasi-Morphological operations

In a next group of experiments we analyzed the impact of the morphological operations to improve the coverage of the Wikipedia corpus. The results are summarized in Table 4. We use operations ignoring the word stem ending (Type 1) and the ones including also the operations restricted to specific word stem endings (Type 2).

Using both types of morphological operations we could improve the translation quality measured in BLEU. For the first type of operations, we only applied the 100 most frequent operations otherwise the translation quality does not improve. Using the second type of operations, we could obtain the best results. We could achieve improvements of 1.3 BLEU points on Dev and 0.7 on the test data. Furthermore, the occurrence of OOV words could be reduced by around 50%.

6.5. Adaptation

We also analyzed how the proposed method can be combined with other adaptation techniques. Although it is hard to find parallel data that matches the domain of university lectures in computer science, it is possible to find data that at least matches the genre. In our case, we used the TED corpus consisting of the subtitles and translations of the talks published on the TED Website[1]. We built additional translation systems, one which just uses the additional data from the TED corpus and one that is also adapted towards TED using a log-linear combination for the phrase table as well as for the language model as for example described in [25].

The results of these experiments are described in Table 5. First, we repeated the result for using the information from Wikipedia on the baseline system without and with the morphological operations. Afterwards, we performed the same series of experiments first with the system using in addition the TED corpus and secondly, using the system also adapted to the TED corpus.

As it can be seen using the additional data from the same genre could improve the translation quality significantly on the devlopment as well as on the test set. But in all cases further improvements using also the Wikipedia data could be obtained. Consequently, the adaptation towards the genre together with the adapation of the vocabulary can be combined successfully. They seem to achieve complementary improvements.

7. Conclusions

We could show that the translation performance can be improved by using Wikipedia as an additional language resource. We analyzed different approaches to include the data into our existing phrase-based translation system and the best performance could be reached by extracting phrase pairs from the corpus of parallel Wikipedia article titles.

We could achieve additional improvements in OOV coverage by learning how different morphological forms of source words map to morphological forms on the target side. Using this approach the coverage of the additional vocabulary could be improved significantly. In a last series of ex-

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

Table 4: *Results for Morphological Operations*

System	Dev					Test				
	BLEU	OOV		mod. OOV		BLEU	OOV		mod. OOV	
		total	unique	total	unique		total	unique	total	unique
Baseline	25.37	1095	378	891	329	22.72	1437	624	1150	521
No Morph Op	26.43	601	315	542	279	23.18	1007	501	825	429
Type 1	26.57	463	263	409	229	23.34	784	413	615	350
Type 2	26.67	415	242	375	213	23.40	718	378	563	326

Table 5: *Results for Adaptation*

System	Dev					Test				
	BLEU	OOV		mod. OOV		BLEU	OOV		mod. OOV	
		total	unique	total	unique		total	unique	total	unique
Baseline	25.37	1095	378	891	329	22.72	1437	624	1150	521
+ WikiPT	26.43	601	315	542	279	23.18	1007	501	825	429
+ Morph Operations	26.67	415	242	375	213	23.40	718	378	563	326
Add. Data	25.84	916	344	741	299	23.55	1219	563	959	467
+ WikiPT	26.74	556	294	497	258	23.92	870	462	703	393
+ Morph Operations	26.98	385	228	345	199	23.04	654	358	514	309
Adapted	27.22	916	344	741	299	25.00	1219	563	959	467
+ WikiPT	28.22	556	294	497	258	25.39	870	462	703	393
+ Morph Operations	28.37	385	228	345	199	25.50	654	358	514	309

periments we could show that this method to adapt the system towards a domain specific vocabulary can be successfully combined with other methods to adapt an SMT system towards a translation task.

Since most words occur only once in the Wikipedia corpus, the alignment quality is significant for the results. Therefore, we will try to improve the alignment quality on the corpus. Furthermore, the morphologic operations currently can be used on morphologically rich languages as input language, since the current setup does not consider the context of the target word. We will try to extend the approach to also handle morphologically rich languages as output language.

8. Acknowledgements

This work was realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

9. References

[1] P. Brown, J. Cocke, S. D. Pietra, V. D. Pietra, F. Jelinek, R. Mercer, and P. Roossin, "A statistical approach to language translation," in *Proceedings of the 12th conference on Computational linguistics - Volume 1*, ser. COLING '88, 1988.

[2] H. D. III and J. Jagarlamudi, "Domain adaptation for machine translation by mining unseen words," in *ACL (Short Papers)*, 2011.

[3] E. Prochasson and P. Fung, "Rare word translation extraction from aligned comparable documents," in *ACL*, 2011.

[4] H. Wu, H. Wang, and C. Zong, "Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora," in *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, ser. COLING '08, 2008.

[5] I. Bulyko, S. Matsoukas, R. Schwartz, L. Nguyen, and J. Makhoul, "Language Model Adaptation in Machine Translation from Speech," in *ICASSP 2007*, Honolulu, USA, 2007.

[6] N. Ueffing, G. Haffari, and A. Sarkar, "Semi-Supervised Model Adaptation for Statistical Machine Translation," *Machine Translation*, vol. 21, no. 2, pp. 77–94, 2007.

[7] H. Schwenk and J. Senellart, "Translation Model Adaptation for an Arabic/French News Translation System by Lightly-Supervised Training," in *MT Summit XII*, Ottawa, Canada, 2009.

[8] N. Bertoldi and M. Federico, "Domain Adaptation for Statistical Machine Translation with Monolingual Resources," in *Fourth Workshop on Statistical Machine Translation*, Athens, Greece, 2009.

[9] G. Foster and R. Kuhn, "Mixture-Model Adaptation for SMT," in *ACL 2007*, Prague, Czech Republic, 2007.

[10] P. Koehn and J. Schroeder, "Experiments in Domain Adaptation for Statistical Machine Translation," in *Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, 2007.

[11] G. Foster, C. Goutte, and R. Kuhn, "Discriminative instance weighting for domain adaptation in statistical machine translation," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '10, 2010.

[12] S. Matsoukas, A.-V. I. Rosti, and B. Zhang, "Discriminative Corpus Weight Estimation for Machine Translation," in *Conference on Empirical Methods on Natural Language Processing (EMNLP 2009)*, Singapore, 2009.

[13] M. Snover, B. Dorr, and R. Schwartz, "Language and Translation Model Adaptation using Comparable Corpora," in *Conference on Empirical Methods on Natural Language Processing (EMNLP 2008)*, Honolulu, USA, 2008.

[14] M. Erdmann, K. Nakayama, T. Hara, and S. Nishio, "An approach for extracting bilingual terminology from wikipedia." *Lecture Notes in Computer Science*, no. 4947, pp. 380–392, 2008, springer.

[15] K. Yu and J. Tsujii, "Bilingual dictionary extraction from wikipedia," in *Proceedings of Machine Translation Summit XII*, 2009. [Online]. Available: http://www.mt-archive.info/MTS-2009-Yu.pdf

[16] K. Toutanova, H. Suzuki, and A. Ruopp, "Applying morphology generation models to machine translation," in *Proceedings of ACL-08: HLT*, Columbus, Ohio, June 2008.

[17] E. Minkov and K. Toutanova, "Generating complex morphology for machine translation," in *In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL07*, 2007, pp. 128–135.

[18] O. Bojar and A. Tamchyna, "Improving Translation Model by Monolingual Data," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, July 2011.

[19] K. Macherey, A. Dai, D. Talbot, A. Popat, and F. Och, "Language-independent compound splitting with morphological operations," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-11)*, 2011.

[20] J. Niehues and S. Vogel, "Discriminative Word Alignment via Alignment Matrix Modeling." in *Proc. of Third ACL Workshop on Statistical Machine Translation*, Columbus, USA, 2008.

[21] 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, "Moses: Open Source Toolkit for Statistical Machine Translation," in *ACL 2007, Demonstration Session*, Prague, Czech Republic, June 23 2007.

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

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

[24] J. Niehues, T. Herrmann, M. Kolss, and A. Waibel, "The Universität Karlsruhe Translation System for the EACL-WMT 2009," in *Fourth Workshop on Statistical Machine Translation (WMT 2009)*, Athens, Greece, 2009.

[25] J. Niehues, M. Mediani, T. Herrmann, M. Heck, C. Herff, and A. Waibel, "The KIT Translation system for IWSLT 2010," in *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, M. Federico, I. Lane, M. Paul, and F. Yvon, Eds., 2010, pp. 93–98.