Character-based PSMT for Closely Related Languages

Jörg Tiedemann
Information Science
University of Groningen
PO Box 716
9700 AS Groningen, The Netherlands
j.tiedemann@rug.nl

Abstract

Translating unknown words between related languages using a character-based statistical machine translation model can be beneficial. In this paper, we describe a simple method to combine character-based models with standard word-based models to increase the coverage of a phrase-based SMT system. Using this approach, we can show a modest improvement when translating between Norwegian and Swedish. The potentials of applying character-based models to closely related languages is also illustrated by applying the character model on its own. The performance of such an approach is similar to the word-level baseline and closer to the reference in terms of string similarity.

1 Introduction

Closely related languages such as Norwegian and Swedish have many features in common. There are obvious similarities not only structurally but also lexically. Many differences are mainly due to writing conventions or consistent changes at the morpheme level. These facts should be beneficial for automatic translation especially for data-driven approaches. However, appropriate training material is often not available for such related languages, at least not in large amounts. This is due to the fact that speakers of these languages easily understand each other without switching to the foreign language and many documents are distributed in the original language only even in the neighboring countries. Still, there is a need for translation even for closely related language pairs as we can see in the Scandinavian situation. There are many types of textual data that have to be translated between languages such as Swedish, Norwegian and Danish ranging from movie subtitles, news to tourist information and others.

Due to the lack of training data for, e.g., Swedish-Norwegian a standard approach using phrase-based statistical machine translation faces the problem of handling unknown words probably more than, for example, the official EU languages for which sufficient amounts of training data is available. However, many of them (not only names) will actually be very similar to their translations. In this paper, we investigate the use of character-based PSMT models to translate such unknown words in order to improve the coverage of the MT system. In this way, we take Weaver's decoding idea to the extreme – translating foreign words as sequences of encoded characters. This approach has already been applied to another pair of closely related languages, Spanish and Catalan (Vilar et al., 2007). Our work mainly follows their approach. However, we use different settings and techniques for training our character-based model and also compare the various setups and their impact on translation quality.

The paper is organized as follows: First we will briefly mention related work. Thereafter, we describe the character-based model we will apply in the experiments discussed in the subsequent section. Finally we will summarize our study with some discussion and conclusions.

2 Related Work

As mentioned earlier, character-based SMT has already been applied to Spanish and Catalan (Vilar et al., 2007). Their letter-based system showed a quite acceptable performance and they concluded

Proceedings of the 13th Annual Conference of the EAMT, pages 12–19,
Barcelona, May 2009

that this technique is especially useful when training material is scarce. They also demonstrate a possible combination of letter-based and word-based models and obtained modest improvements in terms of BLEU scores.

Other solutions for the translation of special types of unknown words have been described in various articles. For example, the translation of named entities is discussed in (Chen et al., 1998; Al-onaizan and Knight, 2002). The treatment of compound words is discussed in (Koehn and Knight, 2003). Another idea for translating unknown words using analogical learning has been proposed by (Langlais and Patry, 2007). In their approach proportional analogies between strings are used to solve analogical equations to retrieve translations of previously unseen terms. The use of phrase-based statistical machine translation on the character level has already been described in (Matthews, 2007). In their work, these models are applied to the task of machine transliteration of Chinese-English and Arabic-English. Similar techniques can also be applied to languages using the same writing system in order to cover spelling differences of names even across related languages (Tiedemann and Nabende, submitted).

3 Character-based PSMT

Phrase-based statistical machine translation (PSMT) can be seen as one of the current state-of-the-art methods in data-driven machine translation. Due to the availability of tools such as Moses (Koehn et al., 2007) and GIZA++ (Och and Ney, 2003) this approach has received a lot of attention in the research community. Phrases in PSMT are usually defined as word N-grams and phrase translation models are estimated from word aligned parallel corpora. However, it is straightforward to apply the same tools used for training word-level PSMT models to train models on a different kind of segmentation level. For example, splitting sentences into character sequences makes it possible to train character based PSMT models in which phrases refer to character N-grams. The same applies for N-gram based language models which can be trained in a similar fashion on the character level. This is exactly the technique that has been applied in (Matthews, 2007) for transliteration and in (Vilar et al., 2007) for translation. In translation, the general assumption is that, similar to the transliteration task, many correspondences between lexical items of related languages can be explained on the character level. However, different to the transliteration approach we probably should not disable reordering as this can be important to capture consistent character movements. Furthermore, character sequences in the phrase table may often correspond to entire words and word phrases. Using reordering in the usual way we can still model phrase movements as in word based settings. In our experiments we will have a look at different settings for reordering in order to see the effect of these parameters.

The process of training character-based PSMT models includes the following steps: First the training data has to be split into character sequences. Important is to treat whitespace characters in a special way in order to keep the information of word boundaries in the data. We simply use the underscore character to replace whitespace characters. Consider the following example to illustrate the format of our training data:

Swedish: - _ D e t _ r ä c k e r _ !
Norwegian: - _ D e t _ e r _ n o k _ !

After pre-processing training data in this way we can use the same procedure as training a word-level model but now on the character level:

- creating a language model of character N-grams from the target language side of the corpus
- cleaning the training data (which includes the removal of sentences longer than 40 characters!)
- aligning characters with GIZA++ (using standard settings for all models involved up to IBM 4)
- symmetrizing character alignments, extracting N-gram translations and estimating their translation probabilities
- tuning the model with an independent development set (also in the same format using character sequences).

The maximum length of 40 tokens per sentence is typically applied for efficiency reasons when aligning with GIZA++. Tokens in our setting refers to characters and the restriction to 40 characters is very unfortunate. A large portion of the

data will be discarded in this way, which is, of course, a serious problem for statistical MT. This problem has already been pointed out by (Vilar et al., 2007). Therefore, they used a different technique for estimating their character-level translation model. They first aligned the corpus at the word level, extracted aligned phrases according to this alignment and then trained the character-based model on those phrase pairs. In this way, the whole corpus can be used assuming that the word alignment and phrase extraction is (mainly) correct.

Fortunately, our data consists of rather short sentences and sentence fragments and, therefore, the reduction of the training corpus due to preprocessing is not as severe as for other types of material. However, we still loose a lot of training data and, therefore, we also apply the two-step procedure as proposed by (Vilar et al., 2007) to compare our results with that approach. Interesting here is especially if the additional training data compensates for possible alignment errors in the phrase extraction.

Hvor er Lamborghinien ? Var är Lamborghinin ?
Jeg er lei av den . Jag är less på den .
Klar til å tape ? Redo att förlora ?
- Hvilke løp har du vunnet ? - Vad har du vunnit för lopp ?
- Ingen . - Inga .
Slår du meg i limousinen til mora di ? Slår du mig i din mammas limousine ?
Ja , jeg slår deg i mors limousin som jeg har trimmet på ymse måter . Ja , jag slår dig i mammas limousine som jag har trimmat på diverse vis .
Så du bø være temmelig redd . Så du bör vara mycket rädd .
Ikke like redd som du når mora di høer det . Inte lika rädd som du när din mamma hör det .

Figure 1: Examples from the Norwegian-Swedish training data.

4 Experiments

4.1 Data

The data for training, tuning and testing our approach is taken from the OpenSubtitle corpus, which is part of the OPUS collection (Tiedemann, 2008). The corpus contains a fair amount of Norwegian-Swedish aligned movie subtitles – still very little with respect to the requirements of statistical MT. Here are some statistics of the data used in our experiments:

training data : two different sets:

word model: 142,654 sentence pairs
1,015,844 Norwegian tokens
990,431 Swedish tokens
character model: ($<= 40$ char/sentence)
108,380 sentence pairs
601,100 Norwegian tokens
595,208 Swedish tokens

development set: 500 alignment units

evaluation set: 500 alignment units

Note, that the aligned sentences/sentence fragments are rather short which is common in movie subtitles. The average length for the training data of the character-based model is even less. The tuning and test sets are used in all experiments and do not have any length restriction. Figure 1 shows some example alignments from our training data.

As we can see in this little sample, there are a lot of similarities between lexical items in Swedish and Norwegian. However, there are also various syntactic differences even though both languages are structurally very close related with each other. Hence, a character-based SMT model will probably not be powerful enough on its own (not even with very long phrases that correspond to words and word n-grams) to take care of these phenomena without a decent reordering model on the word/phrase level. Therefore, we focus on the combination of both, a word-level and a character-level model.

In our experiments, the language models (for both, word LM and character LM) are simply trained on the target language side of our parallel data. We also add more out-of-domain data coming from the Europarl corpus (Koehn, 2005) to test a larger language model also on the character level.

4.2 Evaluation

For evaluation we apply the common automatic measures BLEU and NIST using the target side of the test set as our reference data (hence, we only have one reference per sentence). BLEU and NIST

are computed in the common way at the word-level for all three approaches: word-level SMT, character-level SMT and the combined models. In addition we also look at the string similarity between the translation and the reference sentence. We use the longest common subsequence ratio (LCSR) for this purpose, which is defined as the length of the longest common character sequence of two strings divided by the length of the longer of the two strings. We use this measure only to complement the MT evaluation measures without claiming that higher LCSR scores correlate with more acceptable translations. In future, we would like to investigate if it actually is possible to measure translation quality on the character level as well (using, for example, LCSR) as compared to word error rates, which is frequently used in MT evaluation as well. Here, the assumption would be that words which look more like target language words than others would make translations more acceptable. Especially for unknown words, which are usually just copied from source to target, it could well be the case that a character-based translation that comes close to the correct translation is more acceptable than an untranslated source language item. However, it could also be even more disturbing to see a lot of non-sense words instead of foreign words.

Finally, we also want to look at the significance of some of our result. For this we computed BLEU scores for individual sentences in our test set and compared paired BLEU scores using the nonparametric Wilcoxon matched-pairs signed-ranks test. One problem with BLEU is that it automatically becomes zero if for one of the N-gram sizes no match is found. The chance of seeing such a problem is of course quite high when running on single sentences especially for larger N-grams. Therefore, it is sometimes useful to test BLEU significance for different maximum N-gram sizes.

4.3 Baselines

For all our experiments we applied the Moses toolkit in connection with GIZA++ (Och and Ney, 2003) for word alignment and IRSTLM (Frederico et al., 2008) for language modeling.

4.3.1 Word-based PSMT

For the baseline of word-level PSMT we used a 5-gram language model and a maximum phrase length of 7 words. The alignment heuristics was set to `grow-diag-final-and` and all other parameters for training, phrase extraction and tuning were the default ones. We trained two word-level models using the two different training sets: the "big" training corpus and the "small" corpus that has been reduced in length for the character-level models. The parameters of both models were tuned using the same development set of 500 sentence pairs. Table 1 shows the BLEU and NIST scores for both corpora with different reordering models.

	BLEU	NIST	LCSR
$word_{big}$			
monotone	0.5167	7.3784	0.7675
distance ($<= 6$)	0.5293	7.4596	0.7725
lexicalised ($<= 6$)	0.5273	7.4688	0.7744
monotone (grow)	0.5169	7.4049	0.7668
distance (+EP-LM)	0.5298	7.4650	0.7728
$word_{small}$			
distance ($<= 6$)	0.4968	7.2863	0.7620
lexicalised ($<= 6$)	0.5012	7.2952	0.7595

Table 1: Baseline PSMT models with different types of reordering. The setting $grow$ refers to the alignment heuristics used in combining GIZA++ alignments. Otherwise the standard $grow-diag-final-and$ is applied. *EP-LM* refers to the additional data from the Europarl corpus used for language modeling. The lexicalised reordering model uses the option *msd-bidirectional-fe*.

As expected, the scores for all measures are lower for the smaller training set than for the bigger one[1]. However, the differences are not very big considering that the smaller set only includes about half of the tokens of the bigger set. We can also see that reordering is still important also for closely related languages. However, the improvements are rather modest compared to monotone decoding. The lexicalised reordering model did not add to the performance in this case (except of a very modest improvement on the small dataset). Furthermore, the additional data from Europarl used for language modeling does not increase the performance significantly.

4.3.2 Character-based PSMT

The second type of baseline refers to applying the character-based model to the test set in order to see its potentials when used on its own. We do not

[1]We did not use all reordering models for the smaller corpus. We simply wanted to compare the basic settings only when applied with different amounts of training data.

expect any improvements compared to the word-level baseline especially due to the limited reordering possibilities and the danger in producing nonsense words. We used two different settings for the character-based approach: The first one uses exactly the same settings as the word-based PSMT models. For the second, we increased the maximum length of extracted phrases, the distortion limit and the n-gram size of the language model to 10. The results of both approaches are shown in table 2.

	BLEU	NIST	LCSR
$char_{standard}$			
distance ($<= 6$)	0.4769	7.1455	**0.8030**
lexicalised ($<= 6$)	0.4898	7.1678	**0.8065**
$char_{long}$			
monotone	0.4894	7.1834	**0.8036**
distance ($<= 6$)	0.4917	7.2033	**0.8049**
lexicalised ($<= 6$)	0.5007	7.2775	**0.8094**
distance (+EP-LM)	0.4790	7.1151	**0.8029**
$two - steps$			
monotone	0.4738	7.1005	**0.7983**

Table 2: Character-based PSMT. "long" uses longer phrases (character N-grams – maximum of 10) and a 10-gram language model. In the $two - step$ model we used the phrases extracted from the word aligned corpus using the *grow* alignment heuristics. Otherwise the settings from the $char_{long}$ model apply.

The models are tuned with the same data set as the word-based models (but, of course, split into character sequences). As we can see in table 2, the models with longer phrases and a large N-gram model perform considerably better than the standard models when used with the same type of reordering[2]. They actually perform equally well as the word-based models trained on the same amount of data, which is a very encouraging result. We can also see that reordering still has a positive effect. Even on the character level, reordering still seems to be useful even if the improvements are very modest. Looking at the LCSR scores, we can also see that we get very close to the reference translation in terms of string similarity. Actually the translations are closer to the reference corpus than the ones from the word-based models. However, this does not necessarily have to mean that they are more acceptable as translations.

Finally, we also tested the significance of the BLEU score differences between some of the character-based models and the corresponding word-based models. According to the Wilcoxon matched-pairs test the differences between the character-based model with long phrases and distance-based reordering ($char_{long}$-distance) and the corresponding word-based model ($word_{big}$-distance) is not significant ($p > 0.05$ for BLEU; computed with both, a maximum of 3 and 4 for the size of N-grams to be checked). The same applies to the models using monotone decoding ($p > 0.1$ for max-3-gram-BLEU and max-4-gram-BLEU). For the models with lexicalized reordering, BLEU score differences are weakly significant ($p < 0.05$) for matches up to 4-grams but not for max-3-gram-BLEU scores. This seems to indicate that we indeed get very close to the performance of word-level models for all settings tested.

4.3.3 Character-based PSMT with prior word alignment

As the last baseline, the two-step procedure using word alignment and phrase extraction first to create the training data for the character level model is presented at the bottom of table 2. The advantage here, as mentioned earlier, is that we use the entire corpus for estimating our model instead of restricting ourselves to sentences with a maximum of 40 characters. We used the *grow* alignment heuristics for the word alignment in the first step in order to obtain reliable phrase pairs. Using other heuristics where unaligned tokens are added in the final steps add too much garbage to the training data which seriously harms our character-based model. Using extracted phrases as training material increases the size tremendously. We used the standard phrase extraction implemented in Moses and obtained over 4.5 million phrase pairs after word alignment. This, of course, includes a lot of overlapping phrases extracted from the aligned corpus. This might harm the model and further investigations are needed to check the influence of phrase extraction on this approach. Looking at the results in table 2 we can actually see that there is no improvement to be measured when using the large phrase pair corpus for estimating the character model, at least with monotone reordering as we have tested here. We doubt that other

[2]We did not apply monotone reordering on the smaller set mainly because of time issues. However, we expect the same tendency also for this type of model. We also omit the setting with a larger language model for the first setup as it already fails for the second one.

reordering models would change the results significantly. We will certainly try that in future experiments.

In the next two sections we will now look at two ways of combining character-based and word-based models. In both cases, character-based models are used for unknown words only – the ones we cannot find in the vocabulary files of our word-level model.

4.4 Merging Training Data

The first idea of combining word and character-based models is to merge training data and to train a new global model using both types of data. For this purpose, we simply attached the training data of the character-based model to the training data of the word-based model and trained as usual. Tuning is than also done on a combination of word-level tuning data and character-level tuning data. Certainly, this solution is a bit ad-hoc and especially the confusion between normal one-character words and character level parameters is very disturbing.

For testing, we like to focus on the translation of unknown words with the character-based model whereas other parts of the sentence will be taken care by the word-level model. This will cause another confusion in the model which is related to the distortion parameters learned from data which is either split on word or character level (but not both in the mixed test case). Results of this approach ("split unknown") using our test set are shown in table 3.

	BLEU	NIST	LCSR
standard	0.4979	7.2171	0.7598
split unknown	0.4758	6.9652	0.7602

Table 3: PSMT with merged training data (character-level & word-level). We used standard settings for model estimation, i.e. distance-based reordering and grow-diag-final-and for alignment. "standard" treats unknown words in the usual way by simply copying them to the target language output. In "split unknown" unknown words are split into character sequences before translating.

The scores are very disappointing. First of all, training on the combined data sets decreases already the performance of standard word-level PSMT. This was to be expected due to the ambiguity between character-level data and single-character words as discussed earlier. More disappointing is the combined approach when splitting unknown words into character sequences. The model does not seem to cope well with the input mixture.

4.5 Prior Translation of Unknown Words

The second approach is a cascaded one of translating unknown words first using a character-based model and then translating the rest using a word-level model with the already translated words escaped. Fortunately, Moses supports XML markup for such an escape mode in which translations of certain words are specified with special markup. We use the "exclusive" mode in which these translations will be fixed and copied to the target language output. Table 5 shows the result of applying the two models in such a sequential combination. We compare two settings: one with the "big" word-level model and one with the "small" word-level model. For both cases we use the tuned settings and the settings of the "long" character-based model. We only used distance based reordering without additional data for language modeling. Unfortunately, lexicalised reordering does not seem to work in Moses with additional XML markup in the input.

	BLEU	NIST	LCSR
$char_{long} + word_{small}$	0.5062	7.3513	0.7670
$char_{long} + word_{big}$	**0.5364**	**7.5116**	0.7769

Table 5: Two-step translation: First the character-based models for translating unknown words and then translating sentences with the word-based models (translated unknown words escaped).

Here, we can see a slight improvement in all scores compared to corresponding word-level baselines. The improvements are rather modest but considering that we actually translate only 175 unknown words ($word_{small}$) and 139 unknown words ($word_{big}$) within the 500 test sentences with the character-based model this result is still encouraging. This improvement is also significant according to the Wilcoxon matched-pairs test for both max-3-gram-BLEU scores and max-4-gram-BLEU scores ($p < 0.05$) when compared to the corresponding word-level baseline which is reassuring.

Reference	word-level baseline	$char_{long}$	$char_{long} + word_{big}$
- Välbevandrad .	- Velbevandret .	- Välbevandrats .	- Välbevandrat .
Häll i blekmedlet så här ...	Häll i blekemiddelet så här ...	Töm i blekamedelet sådan ...	Häll i blekamedelet så här ...
Du måste utforska möjligheterna och sen göra ditt val .	Du måste undersöka möjligheter och bestämma dig .	Du måste utforskar möjligheterna och bestämma dig .	Du måste undersöka möjligheter och bestämma dig .
Håller du med ?	Håller du ?	Är du med ?	Håller du ?
Strunta i idiot- tvillingarna .	Skit i idiottvillingene .	Skit i håret idiottvillingarna .	Skit i idiottvillingarna .
Jag måste ta över familjeföretaget .	Jag måste ta över familiefirmaet .	Jag måste ta över familjefirman .	Jag måste ta över familjefirman .
Ska jag inbilla mig att han är drömprinsen ?	Ska jag inbilla mig att han är drømmeprinsen ?	Ska jag inbilla mig att han är drömprinsen ?	Ska jag inbilla mig att han är drömprinsen ?
Han kör så det ryker , men är långt efter .	Han kjøer så det åker , men ligger långt bakom .	Han köar så det ryker , den ligger långt bakom .	Han köer så det åker , men ligger långt bakom .
Du är sån distraktion som jag ville undvika .	Du är ett sånt forstyrrende element som jag ville undvika .	Du är ett sånt förstörande elemente som jag ville undvika .	Du är ett sånt förstörande element som jag ville undvika .
Det är en naiv skolflicksdröm .	Det är en naiv skolejentedrøm .	Det är en naiv skolflickadröm .	Det är en naiv skola identifiedröm .

Table 4: Example translations from the Norwegian-Swedish test set. The first column shows the reference translation and the second column includes the baseline translation using a standard word-level PSMT model. The third column contains the translations of the character model on its own and the last column shows the combined model with unknown word translation as a pre-processing step.

5 Discussion & Conclusions

In this paper, we investigated the use of character-based PSMT models for the translation between closely related languages. The main goal of this approach is to combine such character-level transformations with standard word-level models in order to support the translation of unknown words. In our experiments with Norwegian and Swedish the potentials of such an approach could be seen even when applying such a character-based model on its own. Using this model for unknown words only in a pre-processing step resulted in a slight improvement according to automatic evaluation measures such as BLEU and NIST. Some example translations from the test set are shown in table 4.

Here, we can see some interesting examples. Some of the character-level translations of unknown words are very close to the reference translation, for example "Välbevandrat" (reference: "Välbevadrad") and "skolflickadröm" (reference: "skolflicksdröm"). Others are actually also acceptable even though they are not in the reference translation (for example "familjefirman" – reference: "familjeföretaget" and "förstörande element" – reference: "distraktion"). Other character-level translations are not acceptable, such as "köar" (to queue) instead of "kör" (to drive).

Furthermore, we can definitely see that character-based translation for related languages can be applied to various kinds of unknown words. This makes it very different from machine transliteration for which similar models have been applied before. The phrase-based character model can actually take care of word level translations as well as we can see in the compound "skolflickadröm" translated from the Norwegian "skolejentedrøm". Here, the Norwegian "jente" as part of the compound is translated into the Swedish "flicka" which is most certainly not a cognate word. There are many of such examples in the actual data where the character model takes care of word-level translations. In our data we can find examples such as "rolig - lugn", "akkurat - precis", "greit - okej", "trenger - behöver" and "begynne - att börja".

Furthermore, we have seen that a character-base model is able to generalize over certain regular transformations such as suffix correspondences, for example in translations such as "klippene - klipparna" and "sonettene - sonetterna" or "klarte - klarade" and "mente - menade". Other quite regular character transformations can also be detected, such as "e" to "ä" ("der - där", "kveld - kväll", "er - är", "rett - rätt", "foreldrene - föräldrar"), "kjø" to "kö" ("kjøttet - köttet") or "sjo" to "tio" ("vaksinasjoner - vaccinationer" or "ambisjoner - ambitioner"). All these examples are taken from the actual translations found in our data. Of course, character transformation also leads to many mistakes. However, often these translations come very

close to the correct expressions or at least to a hypothesis that looks very much like the target language. The same applies to the combined method. In most cases, including character level translations produces sentences that look more like the target language than the ones including unknown source language words, even if they contain certain mistakes that often look like typos. Some cases, however, are far from being correct and may disturb the readability more than leaving the original words in the target language output. Some user oriented study should be carried out to formally evaluate this impression. We would also like to investigate other ways of combining character level knowledge with word-level models. For example, we might be able to recognize character-level regularities that can directly be used in a word level model.

An interesting task for future work would be to see if similar techniques may be applied to improve unknown word translation for more distant language pairs as well. This has been tried already for the translation of names for which statistical transliteration modules could be used. This work can easily be extended to include historical cognates and more recent loan words. Furthermore, the character-based translation approach might also be successful for other language pairs with differences in compounding. As we have discussed earlier, compounds, which otherwise would be unknown to the system, can be covered using character-based translation tables.

The main difficulties of applying character-based models to distant language pairs are firstly the recognition of cases in which these models succesfully can be applied (named entity/loanword recognition) and, secondly, the collection of large amounts of appropriate training data, which should include cognates, transliterated names and loan words only. For the coverage of compounds it is even more difficult to find appropriate training data especially because compounding is usually very productive. Simple approaches to compound splitting might still be more effective.

References

Al-onaizan, Yaser and Kevin Knight. 2002. Translating named entities using monolingual and bilingual resources. In *In Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL02*, pages 400–408.

Chen, Hsin-hsi, Sheng-jie Huang, Yung-wei Ding, and Shih-chung Tsai. 1998. Proper name translation in cross-language information retrieval. In *In Proceedings of 17th COLING and 36th ACL*, pages 232–236.

Frederico, M., N. Bertoldi, and M. Cettelo, 2008. *IRSTLM Language Modeling Toolkit, Version 5.10.00*. FBK-irst, Trento, Italy.

Koehn, Philipp and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of EACL*, pages 187–193.

Koehn, Philipp, 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 *Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, Prague, Czech Republic.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit*.

Langlais, Philippe and Alexandre Patry. 2007. Translating unknown words by analogical learning. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 877–886, Prague, Czech Republic, June. Association for Computational Linguistics.

Matthews, David. 2007. Machine transliteration of proper names. Master's thesis, School of Informatics, University of Edinburgh.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Tiedemann, Jörg and Peter Nabende. submitted. Translating transliterations. In *Annual International Conference on Computing and ICT Research (ICCIR 2009)*.

Tiedemann, Jörg. 2008. Synchronizing translated movie subtitles. In *Proceedings of the 6th International Conference on Language Resou rces and Evaluation (LREC'2008)*, Marrakesh, Marocco.

Vilar, David, Jan-Thorsten Peter, and Hermann Ney. 2007. Can we translate letters? In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 33–39, Prague, Czech Republic, June. Association for Computational Linguistics.