The KIT English-French Translation systems for IWSLT 2011

Mohammed Mediani, Eunah Cho, Jan Niehues, Teresa Herrmann and Alex Waibel

Institute of Anthropomatics
KIT - Karlsruhe Institute of Technology
firstname.lastname@kit.edu

Abstract

This paper presents the KIT system participating in the English→French TALK Translation tasks in the framework of the IWSLT 2011 machine translation evaluation.

Our system is a phrase-based translation system using POS-based reordering extended with many additional features. First of all, a special preprocessing is devoted to the Giga corpus in order to minimize the effect of the great amount of noise it contains. In addition, the system gives more importance to the in-domain data by adapting the translation and the language models as well as by using a word-cluster language model. Furthermore, the system is extended by a bilingual language model and a discriminative word lexicon.

The automatic speech transcription input usually has no or wrong punctuation marks, therefore these marks were especially removed from the source training data for the SLT system training.

1. Introduction

In this paper we describe the systems developed for our participation in the IWSLT 2011 TALK tasks for text and speech translation [1].

The TALK tasks consists of translating the transcripts and automatic recognition output of talks held at the TED conferences[1]. The task is very special in the sense that the TED talks differ a lot in respect to topic and domain. However, the style in which the speakers give their presentations is rather similar. A corpus consisting of TED talks is made available for training, which presents data that exactly matches the test condition in genre or style. However, most of the training data consists of large corpora selected from different sources. In some cases they originate from carefully redacted translations such as the EPPS, but for other cases the data was collected from the Web and therefore is rather noisy.

The challenge in developing machine translation systems for this task therefore lies in making the best use of the available data by identifying the benefit that can be drawn from each of the corpora and exploiting them in the best possible way. In our systems this is done on the one hand by processing and filtering the huge, but by tendency noisy data, and on the other hand by exploiting the small data collections for domain and genre adaptation in various ways.

Another challenge is to adapt to the specifics of the speech recognizer output, which produce unreliable punctuation marks so that these cannot be used as cues for the translation system, but rather introduce more noise.

In the following sections we describe the system development. First, we discuss briefly the preprocessing techniques. Afterwards, the baseline system and the different data sets are presented as well as the reordering model, bilingual language model and the different adaptation variants. Then a detailed description of the discriminative word lexicon and the genre cluster language model is given. In the end, the results of the different experiments are presented and conclusions are drawn.

2. Baseline System

For the workshop, the following training data was provided. As parallel sources, the EPPS, NC, UN, TED and Giga corpus were available and as monolingual sources there were the monolingual version of the News Commentary and the News Shuffle corpus. In addition to that, a language model was created based on the Google n-grams.

Our baseline system was trained on the EPPS, TED, NC and UN corpora. For language model training, in addition to the French side of these corpora, we used the provided monolingual data. Systems were tuned and tested against the provided Dev and Test sets.

Before training any of our models, we perform the usual preprocessing, such as removing long sentences and sentences with length difference exceeding a certain threshold. In addition, special symbols, dates and numbers are normalized; then the first word of every sentence is smart-cased.

All the language models used are 4-gram language models with Kneser-Ney smoothing, trained with the SRILM toolkit [2].

The word alignment of the parallel corpora was generated using the GIZA++-Toolkit [3] for both directions. Afterwards, the alignments were combined using the grow-diag-final-and heuristic. The phrases were extracted using the Moses toolkit [4] and then scored by our in-house parallel phrase scorer [5].

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

Word reordering is addressed using the POS-based reordering model and is described in detail in Section 4. The part-of-speech tags for the reordering model are obtained using the TreeTagger [6].

Tuning is performed using Minimum Error Rate Training against the BLEU score as described in [7]. All translations are generated using our in-house phrase-based decoder [8].

3. Preprocessing

The Giga corpus received a special preprocessing in a similar manner to [5]. An SVM classifier was used to filter out bad pairs from the Giga parallel corpus. We used the same set of features. These are: IBM1 score in both directions, the number of unaligned source words, the difference in number of words between source and target, the maximum source word fertility, the number of unaligned target words, and the maximum target word fertility. The lexicons used were generated by Giza++ alignments trained on the EPPS, TED, NC, and UN corpora. The training and test sets used to train and tune the SVM classifier are randomly selected from the aforementioned corpora. Table 1 lists the number of sentences selected from each corpus for the filter training. After the selection process, the sets were augmented with false pairs. In the training, every source sentence is paired with 6 target sentences randomly selected from the target sentences except the corresponding true translation. The same process is performed for the test set but the number of negative examples this time is only 3. Table 2 presents the number of sentences and words in the Giga parallel corpus before and after filtering.

Corpus	#train sentences	#test-sentences
EPPS	636	181
TED	1261	546
NC	1895	546
UN	636	181
Dev	579	364
Test	1222	364
Total	6229	2182

Table 1: Size of the training and test set for the SVM filter

	Original corpus	Filtered corpus
#sentences ($\times 10^6$)	22.52	16.80
#en words ($\times 10^6$)	575.7	447.8
#fr words ($\times 10^6$)	672	527.4

Table 2: Giga corpus size before and after filtering

A great number of Google n-grams include empty words. Consequently, we considered them as noisy. Apparently, this noise is the result of some cleaning operation which removed noisy words but still took them into consideration while extracting the n-grams. After removing them, we performed our usual preprocessing, as mentioned in Section 2, on every entry in the resulting list of n-grams. Table 3 shows the amounts of kept and removed n-grams.

Order	Clean n-grams ($\times 10^6$)	Noisy n-grams ($\times 10^6$)
2-grams	19.12	167.52
3-grams	52.89	1324.33
4-grams	34.92	1315.40
5-grams	32.30	1400.86

Table 3: Google n-gram sizes

3.1. Preprocessing for the Automatic Speech Transcripts

When translating text generated by an automatic speech recognition system, we try to match the text-based training data to the text produced by a speech recognizer. Since the automatic speech recognition system does not generate punctuation marks reliably, punctuation information learned from the training data may not help or may be even harmful when translating. Instead of using a translation model with phrase tables that are built on data containing punctuation, we tried to train the system for the speech translation task using the training corpus without punctuation marks.

Therefore we mapped the alignment from the parallel corpus with punctuation to the corpus without source punctuation. Then we retrained the phrase table, the POS-based reordering model and the bilingual language model.

4. Word Reordering Model

Our word reordering model relies on POS tags as introduced by [9]. The reordering is performed as preprocessing step. Rule extraction is based on two types of input: the Giza alignment of the parallel corpus and its corresponding POS tags generated by the TreeTagger for the source side.

For each sequence of POS tags, where a reordering between source and target sentences is detected, a rule is generated. Its head consists of the sequence of source tags and its body is the permutation of POS tags in the head which matches the order of the corresponding aligned target words.

After that, the rules are scored according to their occurrence and then pruned according to a given threshold. In our system, the reordering is performed as a preprocessing step. Therefore the rules are applied to the test set and possible reorderings are encoded in a word lattice, where the edges are weighted according to the rule's probability. Finally, the decoding is performed on the resulting word lattice.

5. Adaptation

In this translation task, only a quite limited amount of in-domain data exists, but a large amount of out-of-domain data, mainly gathered from the web. To achieve the best possible translation quality, we need to use the better estimated probabilities from all the data, but do not underestimate the domain information encoded in the in-domain part of the data.

In order to optimally use the in-domain data as well as the out-of-domain data, the large out-of-domain models are adapted towards in-domain part of the data. Since the statistical machine translation system consists of different components, they have to be adapted separately. In our case this adaptation was done on the translation models and on the language models.

5.1. Translation Model Adaptation

First, a large model is trained on all the available data. Then, a separate in-domain model is trained on the in-domain data only reusing the same alignment from the large model. This was done, since it seems to be more important for the alignment to have bigger corpora than having only in-domain data.

The two models are then combined using a log-linear combination to achieve the adaptation towards the target domain. The newly created translation model uses the four scores from the general model as well as the two smoothed relative frequencies of both directions from the small in-domain model. If the phrase pair does not occur in the in-domain part, a default score is used instead of a relative frequency. In our case, we used the lowest probability.

5.2. Language Model Adaptation

For the language model, it is also important to perform an adaptation towards the target domain. There are several word sequences, which are quite uncommon in general, but may be used often in the target domain. This is especially important in this task, since most of the training data for the language model is from written sources, while the task is to translate speech.

As it was done for the translation model, the adaptation of the language model is also achieved by a log-linear combination of different models. This also fits well into the global log-linear model used in the translation system. Therefore, we trained a separate language model using only the in-domain data for TED provided in the workshop. Then it was used as an additional language model during decoding and received optimal weights during tuning by the Minimum Error Rate training.

6. Bilingual Language Models

To increase the context used during the translation process, we use a bilingual language model as described in [10]. To model the dependencies between source and target words even beyond borders of phrase pairs, we create a bilingual token out of every target word and all its aligned source words. The tokens are ordered like the target words.

For training, we create a corpus of bilingual tokens from each of the parallel corpora (TED, UN, EPPS, NC and Giga) and then we train one SRI language model based on all the corpora of bilingual tokens. We use an n-gram length of four words. During decoding, this language model is then used to score the different translation hypotheses.

7. Cluster Language Models

As mentioned in the beginning, the TED corpus is very important for this translation task because it exactly matches the target genre. It is characterized by a hugh variety of topics, but the style of the different talks of the corpus is quite similar. When translating a new talk from the same domain, we may not find a good translation in the TED corpus for many topic specific words, since it is quite small compared to the other existing corpora. However, we should try to generate sentences using the same style.

As mentioned in Section 5, we try to model this by introducing an additional language model, which is separately trained on the TED corpus and then combined (in a log-linear way) with the other models. Since the TED corpus is much smaller than the other corpora, the probabilities cannot be estimated as reliably. Furthermore, for the style of a document the word order may not be as important, but the sequence of used word classes may be sufficient to specify the style. To tackle both problems, we try to use a language model based on word classes in addition.

This is done in the following way: In a first step, we cluster the words of the corpus using the MKCLS algorithm [11]. Then we replace the words in the TED corpus by their cluster IDs and train a n-gram language model on this corpus consisting of word classes (all cluster language models used in our systems are 5-gram). During decoding we use the cluster-based language model as an additional model in the log-linear combination.

8. Discriminative Word Lexica

In [12] it was shown that the use of discriminative word lexica (DWL) can improve the translation quality quite significantly. For every target word, they trained a maximum entropy model to determine whether this target word should be in the translated sentence or not. As features for their classifier they used one feature per source word.

One specialty of this task is that we have a lot of parallel data we can train our models on, but only a quite small portion of these data, the TED corpus, is very important to the translation quality. Since building the classifiers on the whole corpus is quite time consuming, we try to train them on the TED corpus only.

When applying a DWL in our experiments we would like to have the same conditions for the training and test case. For this we would need to change the score of the feature only if a new word is added to the hypothesis. If a word is added a second time we do not want to change the feature value. In order to keep track of this, additional bookkeeping would be required. Also the other models in our translation system will prevent us from using a word too often in any case.

Therefore, we ignore this problem and can calculate the score for every phrase pair before starting with the translation. This leads to the following definition of the model:

$$p(e|f) = \prod_{j=1}^{J} p(e_j|f) \tag{1}$$

In this definition $p(e_j|f)$ is calculated using a maximum-likelihood classifier.

Since a translation is generated always using phrase pairs with matching source side, we can restrict the target vocabulary for every source sentence to the respective target side words of those matching phrase pairs. As a consequence, the ME classifier for a given target word, i.e. when learning whether the given target word should occur in the current sentence or not, is trained only on all the sentences that have this target word in their target vocabulary and not on the whole corpus.

As described later on in Section 9.2, this leads in our experiments to a positive influence on the translation quality and as a nice side effect also reduces training time.

9. Results

In the following, we present a summary of our experiments for both MT and SLT tasks and show the impact of the individual models on our system. All the reported scores are the case-sensitive BLEU, and are calculated based on the provided Dev and Test sets.

9.1. Effect of the Google Language Model

A 4-gram language model was trained based on the provided counts by Google[2] as explained in Section 3. This model was tested within different configurations as summarized by Table 4. Baseline 1 and Baseline 2 include (but not limited to) an in-domain language model. Previous experiments on adding more data like the Giga corpus suggested that using more data often improves the translation quality. However, our experiments with the Google n-grams demonstrate that introducing the Google language model dilutes the effect of the smaller models and significantly harms the overall performance of the system. We will further investigate how to best exploit this data so that it can also be beneficial for this translation task.

System	BLEU on Dev	BLEU on Test
Baseline 1	27.51	30.31
+ Google LM	27.34	30.16
Baseline 2	28.58	30.94
+Google LM	28.29	30.43

Table 4: Summary of experiments with Google Language Model

9.2. Effect of the Discriminative Word Lexica

While building the translation system, we compared different methods of building the discriminative word lexicon as described in Section 8. The results are summarized in Table 5. When training the classifiers on all sentences, we could not gain anything on the Dev and slightly lose performance on the Test set. By training the ME classifier only on the sentence, where the word is in the vocabulary, we could improve the translation quality by 0.1 BLEU points on both dev and test sets. Therefore the second variant is used in further experiments.

System	BLEU on Dev	BLEU on Test
Baseline	28.35	31.07
DWL all Sentences	28.33	30.98
DWL subset	28.50	31.16

Table 5: Summary of experiments with DWL

9.3. MT Task

Table 6 presents a summary of the experiments performed while developing the translation system for the MT task. The baseline system was built without the Giga corpus, since the translation model with all data took much longer to train. In other words, the baseline system was trained on the EPPS, TED, NC, and UN corpora. Three language models were combined log-linearly. The first consists of the target side of the parallel data. The remaining language models are built from the monolingual data, one for each available corpus. This baseline configuration, led to a BLEU score of 25.84 on Dev and 28.38 on Test. Considerable gain of around 0.7 could be obtained on Dev and Test by introducing the POS reordering model. Based on our previous experience with this pair of languages, we only used the short range reordering rules. These rules were trained on the same corpus excluding UN documents, because extracting rules from larger corpora has little effect on the performance but on the other hand consumes too much resources.

Next, the Giga corpus data were introduced. These add an important gain to our system: 1.22 points on Dev and 0.81 points on Test.

The following two experiments demonstrate the importance of adaptation for this task. First, additional 0.43 points on Dev and 0.97 on Test could be added to our system by adapting the language model. An indomain language model built on the TED data was used as explained in Section 5.2. Second, as for the language model adaptation, TED data were used as an in-domain translation model to adapt the general model. This increases our scores on Dev by around 0.16 and on Test by around 0.37.

Afterwards, little increase of 0.07 could be gained on Test by performing a 2-step adaptation procedure: first, the complete model consisting of all data is adapted towards the

[2] http://books.google.com/ngrams/datasets

cleaner but smaller part, namely, EPPS, TED, and NC. Then the result of the first step is again adapted towards the in-domain model consisting of TED only.

The genre model was of great effect for this task. By including the cluster-based language model trained only on the TED corpus, we could gain around 0.4 points on Dev and 0.3 on Test. The discriminative word lexicon approach using only the TED corpus improves our scores by 0.11 both on Test and on Dev.

Finally, we added a bilingual language model to our system.This improves the score on Dev by around 0.2 and on Test by around 0.4 leading to final scores 28.98 BLEU points on Dev and 31.95 on Test. This last system, was the system we used to translate the evaluation set (Test2011) for our submission.

System	BLEU on Dev	BLEU on Test
Baseline	25.84	28.38
+POS reordering	26.54	29.02
+Giga data	27.76	29.83
+LM adaptation	28.19	30.70
+TM adaptation	28.35	31.07
+2-step TM adaptation	28.30	31.14
+Cluster LM	28.69	31.43
+DWL	28.80	31.54
+Bilingual LM	28.98	31.95

Table 6: Summary of experiments for the En-Fr MT task

9.4. SLT Task

Our system for the SLT task evolved as shown in Table 7. The baseline system of the speech translation task used the same configuration as the one for the MT task, for which the POS reordering, the Giga data, and the adaptation for both translation and language model were added to the baseline. In other words, it corresponds to the system with 28.35 on Dev and 31.07 on Test of the MT task in Table 6. The scaling factors used in this baseline system were imported from the corresponding MT system. We used the models built with punctuation marks and there was no treatment regarding punctuation marks on the test set.

Then we tried applying translation models built using the corpus without punctuation as described in the previous section. The bilingual language model and phrase table were trained on EPPS and all other available parallel data, whose punctuation marks on the source side were all removed. The punctuation marks on the test set were also removed. By doing this, we gained more than 2.9 BLEU points.

After applying re-optimization to match more accurately the models built without punctuation, we gained more than 1.5 BLEU points on Test. By adding the bilingual language model to extend the context of source language words available for translation, we could improve further by 0.4 on Dev and Test. To train the bilingual language model, we removed the punctuation from the corpus and trained the language model on this corpus together with the target side corpus with punctuation. We then included the cluster-based language model trained on the TED corpus. By adding this language model we gained 0.2 both on Dev and Test. The discriminative word lexicon was trained using the punctuation-free TED corpus as well. When applying the discriminative word lexicon, we used a big language model built using all parallel training data, News corpora and monolingual data. This yielded more improvements, i.e. 0.3 points on Test. This system was the system we used to translate the SLT evaluation set for our submission.

System	BLEU on Dev	BLEU on Test
Baseline	-	16.14
+ Punctuation Removal	-	19.05
+ Re-optimization	24.94	20.61
+ Bilingual LM	25.33	21.00
+ Cluster LM	25.58	21.24
+ DWL[1]	25.47	21.58

Table 7: Summary of experiments for the En-Fr SLT task

10. Conclusions

We have described the systems developed for our participation in the TALK translation in both speech translation and text translation tasks from English into French. Our phrase-based machine translation system was extended with different models.

The different word order between languages, one of the most problematic issues in machine translation, was addressed by a POS-based reordering model, which improves the word order in the generated target sentence.

The experiments clearly show the advantage of exploiting the large amount of information integrated in the out-of-domain corpora. This is particularly noticeable for the Giga corpus which would not have such influence without the special cleaning and filtering to minimize the noise it infiltrates into the translation model.

Removing the punctuation marks in automatic transcription input, which is sometimes wrongly punctuated or has no punctuation at all, is extremely beneficial. Our SLT experiments demonstrate that the system's performance was boosted using this procedure.

Unfortunately, the language model built based on the Google n-grams did not help us in this task, in spite of the effort devoted to making them useful. A potential reason for this negative impact would be the timeline of these n-grams, some of which go two centuries back in history.

It seems that in such tasks data should not be given equal importance. Indeed, the improvements we got using different

[1] no News LM, no Mono LM

adaptation approaches teach us two facts. First, cleaner parts should be given higher weight because of the correlation between corpus quality and translation performance. Second, the in-domain parts should be particularly distinguished and given a weight which corresponds to their degree of representation of the target domain.

In fact, even if only a little amount of in-domain data that is very close to the test data is available, it can improve the system's performance when exploited in the right way. For instance, the increase in translation quality gained by the discriminative word lexica was measurable on both tasks and the cluster-based language model brought about additional improvements.

11. Acknowledgements

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

12. References

[1] M. Federico, L. Ventivogli, M. Paul, and S. Stueker, "Overview of the IWSLT 2011 Evaluation Campaign," in *IWSLT 2011*, 2011.

[2] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit." in *International Conference on Spoken Language Processing*, Denver, Colorado, USA, 2002.

[3] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[4] 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 *Proceedings of ACL 2007, Demonstration Session*, Prague, Czech Republic, 2007.

[5] T. Herrmann, M. Mediani, J. Niehues, and A. Waibel, "The Karlsruhe Institute of Technology Translation Systems for the WMT 2011," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, 2011.

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

[7] A. Venugopal, A. Zollman, and A. Waibel, "Training and Evaluation Error Minimization Rules for Statistical Machine Translation," in *Workshop on Data-drive Machine Translation and Beyond (WPT-05)*, Ann Arbor, Michigan, USA, 2005.

[8] S. Vogel, "SMT Decoder Dissected: Word Reordering." in *International Conference on Natural Language Processing and Knowledge Engineering*, Beijing, China, 2003.

[9] K. Rottmann and S. Vogel, "Word Reordering in Statistical Machine Translation with a POS-Based Distortion Model," in *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Skövde, Sweden, 2007.

[10] J. Niehues, T. Herrmann, S. Vogel, and A. Waibel, "Wider Context by Using Bilingual Language Models in Machine Translation," in *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, 2011.

[11] F. J. Och, "An Efficient Method for Determining Bilingual Word Classes." in *EACL'99*, 1999, pp. 71–76.

[12] A. Mauser, S. Hasan, and H. Ney, "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*, ser. EMNLP '09, Singapore, 2009.