LIUM's Statistical Machine Translation System for the NTCIR Chinese/English PatentMT

Holger Schwenk
LIUM
University of Le Mans
Holger.Schwenk@lium.univ-lemans.fr

Sadaf Abdul-Rauf
LIUM
University of Le Mans
Sadaf.Abdul-Rauf@lium.univ-lemans.fr

ABSTRACT

This paper describes the development of a Chinese–English statistical machine translation system for the 2011 NTCIR patent translation task. We used phrase-based and hierarchical systems based on the Moses decoder, trained on the provided data only. Additional features include translation model adaptation using monolingual data and a continuous space language model. We report comparative results for these various configurations.

Categories and Subject Descriptors

I.2.7 [**Artificial intelligence**]: Natural Language Processing—*Machine Translation*

General Terms

Natural Language Processing

Keywords

Statistical machine translation, unsupervised training, continuous space language model

Team Name: LIUM
Subtasks/Languages: Chinese-to-English
External Resources Used: Stanford segmenter, Giza++, Moses, SRILM

1. INTRODUCTION

This paper describes the statistical machine translation systems developed by the computer science laboratory of the University of Le Mans (LIUM) for the patent translation task organized in the framework of the NTCIR workshop. We only participated in the Chinese–English patent translations task. This was our first participation to this evaluation. A detailed description of all the tasks and comparative results can be found in [3]. We developed a hierarchical system based on the Moses software [4] and our own extensions. These include translation model adaptation using monolingual data and continuous space language models. The systems are described in the following.

This paper is organized as follows. In the next section, we first describe the architecture of our approach. A detailed experimental comparison of the various models is given in section 3. The paper concludes with a discussion and possible directions of future research.

2. ARCHITECTURE OF THE APPROACH

The overall architecture of our approach is depicted in figure 1. Translation is performed in two passes. First, we use `moses` or `moses_chart` to produce an n-best list of possible translations. The LM probabilities on these n-best lists are then rescored with a special language model called continuous space language model and the coefficients of all models are retuned to optimize the BLEU score on the development data. For this we use a publicly available tool named CONDOR [2].

The translation mode (phrase- or rule-table), is trained in several steps. We want to translate from Chinese to English. For this, we dispose of parallel data and monolingual data in the target language. This data is used to build a baseline system. This translation model us then *adapted* using large amounts of monolingual data in the target language, e.g. English. For this task, we have approximately 36M words of parallel training data. This is a small amount in comparison to other tasks like French—English news translation (WMT tasks) or the translation from Arabic and Chinese into English in the framework of the NIST evaluations. Based on our previous experiences [8, 9, 5] we applied unsupervised training. The main idea is to use an existing system to translate monolingual data and to add the source texts together with the automatic translations to the parallel training data, after some filtering.

A somehow similar approach was named self-enhancing of the translation model [11]. The idea is to translate the test data only, to filter the translations with help of a confidence score and to use the most reliable ones to train an additional small phrase table that is jointly used with the generic phrase table. In follow up work, this approach was refined [12]. Unsupervised training as proposed in [8] differs from self-enhancing since it does not adapt itself to the test data, but large amounts of monolingual training data are translated and a completely new model is built. This model can be applied to any test data.

The approach proposed in this work is related to both self-improvement and lightly-supervised training. Our idea is to exploit the available in-domain monolingual data in the target language. This data is usually available in large amounts since it is needed to train the target language model. We propose to use information retrieval (IR) techniques to select a small subset of relevant sentences in this collection. The queries for IR are either the reference translations of the development data or the automatic translations of the test data as produced by a baseline system. These sentences are then translated back to the source language by an *inverse*

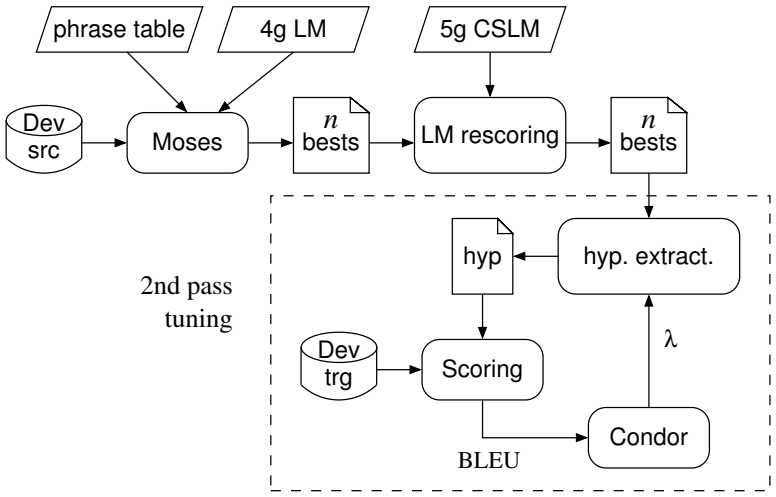

Figure 1: Overall architecture of the SMT system.

system and this data is used as additional parallel training data, without any additional filtering. By these means we perform unsupervised training similar to [8]. An important difference is that we actively select which data to translate instead of blindly translating large amounts and then applying a threshold on some confidence score.

2.1 Continuous space language model

We also applied the so-called continuous space language model. The basic idea of this approach is to project the word indices onto a continuous space and to use a probability estimator operating on this space [1]. Since the resulting probability functions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. A neural network can be used to simultaneously learn the projection of the words onto the continuous space and to estimate the n-gram probabilities. This is still an n-gram approach, but the language model posterior probabilities are "interpolated" for any possible context of length $n-1$ instead of backing-off to shorter contexts. This approach is expected to take better advantage of the limited amount of training data.

Training is performed with the standard back-propagation method using weight decay and a re-sampling algorithm. This approach is described in detail in [7].

3. EXPERIMENTAL RESULTS

All results described in this paper were obtained using only the data provided in the framework of the Patent translation task of the NTCIR workshop (constrained condition). For this task, 1 million sentences of parallel data and 2000 sentences of development data are available. We randomly split the available development data into two parts: one to tune the parameters of our systems and the other one as internal test set. Only one reference translation is available. For language modeling huge amounts of monolingual English patent texts are available, coming from several years of US patent data. Detailed statistics of these corpora are given in Table 1. Note that all the available data can be considered as specific to the task, i.e in-domain.

Corpus	#lines	Chinese [M chars]	English [M words]
Train	1M	167M	36.5M
Development	999	166k	33.7k
Internal test	1001	165k	34.1k
US patents	-	-	13.6T

Table 1: Statistics of the available data.

3.1 Baseline Chinese/English systems

We build baseline systems using all the available training data. The translation model was trained on 1M sentences of parallel data. The Chinese characters were segmented using the Stanford segmenter with the PKU standard. This resulted in approximately 38M "words". On the English side, we kept the case of the words.

A 4-gram back-off language model was trained on all the available monolingual data, e.g. the English side of the bitexts and 13.6T words of US patent texts. This data was split into several parts, individual LMs were trained using modified Kneser-Ney smoothing as implemented in the SRI LM toolkit [10] and then interpolated to get one huge LM. The corresponding interpolation coefficients were calculated to optimize the perplexity on the development data using the usual EM procedure. In addition, we have observed small improvements by keeping all observed n-grams, i.e. using a cut-off value of 1. The perplexity on the development data of this huge LM is 79.5, and the file occupies 40 GBytes on the disk in the binary representation of the SRI toolkit. A continuous space language model was trained on a subset of

Bitexts	LM data	BLEU	
		Dev	Test
Phrase-based	36.5M	33.26	31.64
	13.6T	35.17	**33.56**
Hierarchical	36.5M	33.55	32.48
	13.6T	36.14	**34.93**

Table 2: Performances of the baseline Chinese/English systems

System type	Bitexts corpus	M words	BLEU Dev	BLEU Test
Baseline systems:				
Phrase	human only	37.8M	35.17	33.56
Hiero			36.14	34.93
Selection with sentence confidence level:				
Phrase-based	human+threshold 0.005	38.9M	35.17	33.83
	human+threshold 0.01	39.2M	35.14	33.74
	human+threshold 0.02	39.8M	34.87	33.50
	human+threshold 0.03	40.7M	35.05	33.45
	human+threshold 0.04	41.8M	35.09	33.46
Hiero	human+threshold 0.005	38.9M	35.99	35.08
Selection on n best IR sentences:				
Phrase-based	human+10-best IR	41.0M	35.20	33.81
	human+20-best IR	42.0M	**35.29**	**34.20**
	human+25-best IR	43.0M	35.29	34.00
Hiero	human+20-best IR	38.9M	**36.41**	**35.45**

Table 3: Comparison of different adaptation techniques of phrase-based and hierarchical systems.

the data using a resampling technique and interpolated with the back-off LM. This gives a perplexity of 71.7.

We build standard phrase-based and hierarchical SMT systems using the Moses toolkit [4]. For both models we used the default procedure. The result of both models are summarized in Table 2. The BLEU scores are calculated with the tool `multi-bleu.perl` as provided by the Moses tool kit. Scoring is case sensitive and includes punctuation.

The hierarchical system clearly performs better than the phrase-based one. This observation is in-line with other reports on the translation of Chinese to English.

3.2 English/Chinese translation system

Since no additional Chinese monolingual data was available, we used (a subset) of the English monolingual data to perform unsupervised training. We argue that this is a very common setting: usually there are large amounts of in-domain data available in the target language that are collected for language modeling.

In addition, we have observed in the past that it is better to translate from the target to the source language instead of the inverse direction [5] Unsupervised training can of course produce wrong translations. When those are added on the target side of the phrase-table they may be actually used in future translations and the errors propagate. On the other hand, when translating from the target to the source language the possible errors will appear in the source phrases. We argue that this will have less impact since it is less likely that wrong phrases will be matched when translating grammatically correct sentences.

Pruning Beam	Pruning Stack	Pruning Transl.	BLEU Dev	BLEU Test	Speed [word/s]
0.4	200	50	29.63	27.79	54
0.4	100	50	29.67	27.53	116
0.4	100	20	29.64	27.83	280
0.4	100	10	**29.83**	**27.94**	566
0.4	50	10	29.71	27.80	998
0.5	100	20	29.67	27.75	296
0.6	100	20	29.76	27.62	310

Table 4: Performance of the English/Chinese phrase-based systems.

On the other hand, we need to build an SMT system for the inverse translation direction, in our case from English to Chinese. We only built a phrase-based system for this translation direction since it is faster to build and to run, using the available resources (1M sentence pairs of bitexts and 167M characters for language modeling). Their performances are given in Table 4, for different settings of the pruning parameters.

First of all, one realizes that the BLEU scores are about 5.5 BLEU points lower for this translation direction. The English/Chinese system will be used to potentially translate large amounts of monolingual data. Consequently, we investigated different pruning settings to speed up the system without a large impact on the translation quality. We were surprised to see that the pruning parameters seem to have almost no impact on the translation quality, the system that performs 10 times faster than our default actually performs slightly better (BLEU on test 27.79 → 27.94). We used this system to translate the monolingual data.

3.3 Unsupervised training

We have first done experiments with the two adaptation methods for a phrase-based system only since they are faster to build and to tune (due to the large rule-tables of hierarchical systems). We then built hierarchical systems for the most interesting operation points only. For this, we translated all the English US patents of the year 2005 to Chinese, i.e. 1145M words. The first adaptation method was implemented as proposed in [8], i.e. keeping only the most reliable translations according to the word normalized confidence score. The second method, the new one proposed in this paper, consists in using the English side of the development data as queries to retrieve related sentences in the huge corpus of 1145M words of LM training data. For each query sentence, we retrieved up to 100 sentences, sorted according to the score of the IR process. The Lemur IR toolkit [6] was used for sentence extraction. We then used the n first sentences and their automatic translations as additional bitexts. It is important to note that our new method is much more efficient: instead of translating blindly more than 1 bil-

	Baseline system	Improved system with conf. score	Improved system using IR
Number of entries	8.1M	8.0M	8.1M
Number of different source phrases	41993	42037	43227
Number of new source phrases	n/a	809	2148
Average number of translations	191.8	191.3	187.3
Average length of source phrases	2.64	2.65	2.67

Table 5: Characteristics of the phrase tables of the baseline and the improved systems. In both cases the table was filtered to include only entries that could be applied on the test data.

lion of words in order to keep only a couple of million words, we first select the interesting sentences and then translate them. This is more than two orders of magnitude faster. The results of these two methods, together with the baseline systems, are summarized in Table 3. All systems use the huge LM trained on 13.6 billion words.

Adding the the most reliable translations according to the word-normalized sentence confidence scores yielded only modest improvements in the BLEU scores on the test data: 33.56 to 33.83 for the phrase-based system and 34.93 to 35.08 for the hierarchical system, and this for a very restrictive confidence score (only 1.1M words were added). Adding more words degraded the performances.

On the other hand, the proposed new method yielded significant improvement in the BLEU score on the test data of more than 0.5 BLEU. The phrase-based system improved from 33.56 to 34.20 BLEU and the hierarchical system from 34.93 to 35.45 BLEU. This was obtained by using the 20 top ranking sentences for each IR query, i.e. around 4.2M words, roughly 10% of additional parallel training data. An improvement of 0.5 BLEU may sound modest, but experience has shown that such gains in heavily tuned state-of-the-art systems are not easy to obtain. Note also that we have only one reference translation. We conjecture that a gain of 0.5 BLEU with one reference would correspond to more than 1 point BLEU difference if we had four reference translations.

Finally, we combined both adaptation techniques and applied a threshold on the word normalized confidence scores of the sentences that were selected by the IR process. In our experience, we were not able to improve the results obtained by using the IR retrieved sentences. This seems to indicate that the topic closeness of the adaptation data seems to be more important than the quality of the translations (since we translate from the target to the source).

3.4 Results analysis

In order to get more insight on this unsupervised training method we tried to analyze the phrase-table of the baseline system (BLEU score of 33.56 on test data), the system improved using selection with word-normalized sentence confidence score (BLEU score of 33.83) and the system improved using selection of the data with IR methods (BLEU score of 34.20 in Table 3). We show the total number of entries in the phrase-table, the number of entries with different source phrases, the number of new source phrases, the average number of translations per source phrase (actually the fraction of the both first quantities) and the average length of the source phrases.

[9] reported an significant decrease in the number of average translations per source phrase, in the order of more than a thousand for the generic system, and around 40 after adaptation. We did not observe this tendency in our experiments: there is no notable difference for all indicators when using the word-normalized sentence confidence score to select the sentences to add to the parallel training data. We explain this by the fact that in our system all the parallel training data can be considered as in-domain – we do not use generic corpora like the UN corpus. Therefore there is no need to "*filter*" the possible translations in the phrase-table by unsupervised training on in-domain monolingual data.

On the other hand, here is a slight decrease in the average number of translations for the system adapted using the IR methods: from 191.8 to 187.3. It is also striking to see that the proposed approach resulted in 2148 new source phrase in comparison to only 809 when using the confidence score to select the sentences to keep. It is important to remember that is impossible to learn new translations using unsupervised training. When adding automatically translated sentences to the bitexts the phrase extraction algorithm can only modify the probability distributions of existing phrase pairs or add new source phrases that were not previously observed (using parts of existing translations). An interesting effect is that we used a phrase-based system to produce automatic translations to improve a hierarchical system. We will investigate in the future whether this change of system architecture is actually beneficial.

3.5 Official system

The best system in Table 3 is the hierarchical system trained on all provided human translation and the 20-best automatic translations obtained by information retrieval (last line in Table 3). We used this system to generate 1000-best lists which were rescored with the continuous space language model. The coefficients of all models were retuned to optimize the BLEU score on the development data (see Figure 1). For this we use a publicly available tool named CONDOR [2]. The use of the continuous space language model

System type	Bitexts	LM	BLEU Dev	BLEU Test
Hiero	human+20-best IR	back-off	36.41	35.45
		CSLM	37.01	**35.91**

Table 6: Performance of the continuous space language model.

yielded an additional improvement of about 0.5 BLEU on the test set (see Table 6). The system which obtains a BLEU score of 35.91 on the test data is our official submission. The organizers of the evaluation reported a BLEU score of 34.76 for LIUM's system.

4. CONCLUSION

This paper described the system developed by LIUM for the Chinese—English NTCIR patent translation task. We described various experiments with phrase-based and hierarchical statistical machine translation systems. All systems are based on the Moses toolkit and our own extensions. We have described a new approach to adapt the translation model using monolingual data in the target language. We used information retrieval techniques to find a relevant subset of the available English patent data and translated it back to Chinese. This data was then used as additional parallel training data. This yielded an improvement of 0.4 BLEU on the internal test data. We also observed significant improvement by applying a continuous space language model.

5. ACKNOWLEDGMENTS

This work has been partially funded by the European Commission under the project EuromatrixPlus (ICT-2007.2.2-FP7-231720).

6. REFERENCES

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *JMLR*, 3(2):1137–1155, 2003.

[2] F. V. Berghen and H. Bersini. CONDOR, a new parallel, constrained extension of powell's UOBYQA algorithm: Experimental results and comparison with the DFO algorithm. *Journal of Computational and Applied Mathematics*, 181:157–175, 2005.

[3] I. Goto, B. Lu, K. P. Chow, E. Sumita, and B. K. Tsou. Overview of the patent machine translation task at the ntcir-9 workshop. In *The 9th NTCIR Workshop Meeting*, December 2011.

[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 *ACL, demonstration session*, 2007.

[5] P. Lambert, H. Schwenk, C. Servan, and S. Abdul-Rauf. Investigations on translation model adaptation using monolingual data. In *Sixth Workshop on SMT*, 2011.

[6] P. Ogilvie and J. Callan. Experiments using the Lemur toolkit. In *In Proceedings of the Tenth Text Retrieval Conference (TREC-10)*, pages 103–108, 2001.

[7] H. Schwenk. Continuous space language models. *Computer Speech and Language*, 21:492–518, 2007.

[8] H. Schwenk. Investigations on large-scale lightly-supervised training for statistical machine translation. In *IWSLT*, pages 182–189, 2008.

[9] H. Schwenk and J. Senellart. Translation model adaptation for an Arabic/French news translation system by lightly-supervised training. In *MT Summit*, 2009.

[10] A. Stolcke. SRILM - an extensible language modeling toolkit. In *ICSLP*, pages II: 901–904, 2002.

[11] N. Ueffing. Using monolingual source-language data to improve MT performance. In *IWSLT*, pages 174–181, 2006.

[12] N. Ueffing. Transductive learning for statistical machine translation. In *ACL*, pages 25–32, 2007.