The LIUM Arabic/English Statistical Machine Translation System for IWSLT 2008

Holger Schwenk, Yannick Estève and Sadaf Abdul Rauf

LIUM, University of Le Mans, FRANCE
name.surname@lium.univ-lemans.fr

Abstract

This paper describes the system developed by the LIUM laboratory for the 2008 IWSLT evaluation. We only participated in the Arabic/English BTEC task. We developed a statistical phrase-based system using the Moses toolkit and SYSTRAN's rule-based translation system to perform a morphological decomposition of the Arabic words. A continuous space language model was deployed to improve the modeling of the target language. Both approaches achieved significant improvements in the BLEU score. The system achieves a score of 49.4 on the test set of the 2008 IWSLT evaluation.

1. Introduction

This paper describes the system developed by the LIUM laboratory for the 2008 IWSLT evaluation. We only participated in the Arabic/English BTEC task. The architecture of the system is very similar to a large system built for the NIST Arabic/English task [1] or a system built for the translation between French and English [2]. All three are statistical phrase-based machine translation systems based on the freely available Moses decoder [3], with extensions for rescoring n-best lists with a continuous space language model in a second pass. No system combination is used.

The training data of the translation model of the IWSLT system is limited to the provided BTEC corpora. Small improvements could be achieved using additional language model training data, namely LDC's Gigaword corpus. All the models are case sensitive and include punctuation markers. We compare two different tokenization of the Arabic source text: a full word mode and a morphological decomposition kindly provided by SYSTRAN. The later one achieved improvements in the BLEU score of several points.

This paper is organized as follows. In the next section, the main architecture of the SMT system architecture is presented. In the following section the experimental results are provided and commented. The paper concludes with a discussion of future research issues.

2. System architecture

The goal of statistical machine translation is to produce a target sentence $\mathbf{e}$ from a source sentence $\mathbf{f}$. It is today common practice to use phrases as translation units [4, 5] and a log linear framework in order to introduce several models explaining the translation process:

$$\begin{aligned} \mathbf{e}^* &= \arg\max_e p(\mathbf{e}|\mathbf{f}) \\ &= \arg\max_e p(\mathbf{f},\mathbf{e})P(\mathbf{e}) \\ &= \arg\max_e \{exp(\sum_i \lambda_i h_i(\mathbf{e},\mathbf{f}))\} \end{aligned} \quad (1)$$

The feature functions h_i are the system models and the λ_i weights are typically optimized to maximize a scoring function on a development set [6]. In our system fourteen features functions were used, namely phrase and lexical translation probabilities in both directions, seven features for the lexicalized distortion model, a word and a phrase penalty and a target language model (LM).

The system is based on the Moses SMT toolkit [3] and constructed as follows. First, Giza++ is used to perform word alignments in both directions. Second, phrases and lexical reorderings are extracted. Both steps use the default settings of the Moses SMT toolkit. A 4-gram back-off target LM is then constructed as detailed in section 3.2. The translation itself is performed in two passes: first, Moses is run and a 1000-best list is generated for each sentence. The parameters of this first pass are tuned on development data using the cmert tool. These 1000-best lists are then rescored with a continuous space 4-gram LM and the weights of the feature functions are again optimized using the open source numerical optimization toolkit Condor [7]. The details of this optimization procedure are as follows:

1. The n-best lists are reranked using the current set of weights. A hypothesis is extracted and scored against the reference translations of the development data.

2. The obtained BLEU score is passed to *Condor*, which either computes a new set of weights (the algorithm then proceeds to step 1) or detects that a local maximum has been reached and the algorithm stops iterating.

It is stressed that Moses and the continuous space language model are only run once and that the whole second pass tuning operates on n-best lists. This usually takes less than an hour, most of the time being used by the NIST scoring tool. This basic architecture of the system is summarized in Figure 1.

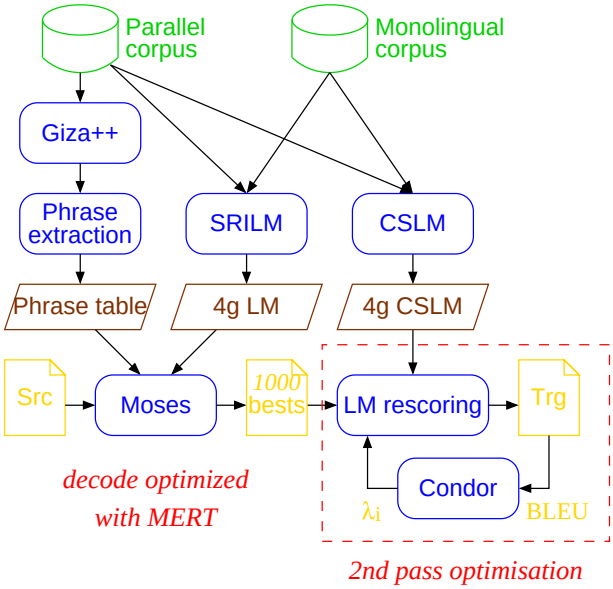

Figure 1: System architecture – see text for details.

2.1. Continuous space language model

For the BTEC task, there are less than 400k words of in-domain texts available to train the target language models. This is a quite limited amount in comparison to tasks like the NIST machine translation evaluations for which several billion words of newspaper texts are available. Small improvements were obtained by adding large amounts of generic news paper texts. We also deployed specific techniques to make the most out of the limited resources.

In this paper, we propose to use 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 [8]. 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 a 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 was already successfully applied in statistical machine translation systems, ranging from small IWSLT systems [9, 10] to large NIST systems [1].

A standard fully-connected multi-layer perceptron is used. The inputs to the neural network are the indices of the $n-1$ previous words in the vocabulary $h_j = w_{j-n+1}, \ldots, w_{j-2}, w_{j-1}$ and the outputs are the posterior probabilities of *all* words of the vocabulary:

$$P(w_j = i|h_j) \qquad \forall i \in [1, N] \tag{2}$$

where N is the size of the vocabulary. The input uses the so-called 1-of-n coding, i.e., the ith word of the vocabulary is coded by setting the ith element of the vector to 1 and all the other elements to 0. The ith line of the $N \times P$ dimensional projection matrix corresponds to the continuous representation of the ith word. Let us denote c_l these projections, d_j the hidden layer activities, o_i the outputs, p_i their softmax normalization, and m_{jl}, b_j, v_{ij} and k_i the hidden and output layer weights and the corresponding biases. Using these notations, the neural network performs the following operations:

$$d_j = \tanh\left(\sum_l m_{jl}\, c_l + b_j\right) \tag{3}$$

$$o_i = \sum_j v_{ij}\, d_j + k_i \tag{4}$$

$$p_i = e^{o_i} \,/\, \sum_{r=1}^{N} e^{o_r} \tag{5}$$

The value of the output neuron p_i corresponds directly to the probability $P(w_j = i|h_j)$.

Training is performed with the standard back-propagation algorithm minimizing the following error function:

$$E = \sum_{i=1}^{N} t_i \log p_i + \beta \left(\sum_{jl} m_{jl}^2 + \sum_{ij} v_{ij}^2\right) \tag{6}$$

where t_i denotes the desired output, i.e., the probability should be 1.0 for the next word in the training sentence and 0.0 for all the other ones. The first part of this equation is the cross-entropy between the output and the target probability distributions, and the second part is a regularization term that aims to prevent the neural network from over-fitting the training data (weight decay). The parameter β has to be determined experimentally. Training is done using a re-sampling algorithm as described in [11].

It can be shown that the outputs of a neural network trained in this manner converge to the posterior probabilities. Therefore, the neural network directly minimizes the perplexity on the training data. Note also that the gradient is back-propagated through the projection-layer, which means that the neural network learns the projection of the words onto the continuous space that is best for the probability estimation task.

3. Experimental evaluation

3.1. Available data

The organizers of IWSLT provide several task specific corpora that can be used to train and optimize the translation system. The characteristics of these corpora are summarized in Table 1. It is known that the choice of the development and internal test data may have an important impact on the quality of the system, in particular when the available corpora have different characteristics (for instance what concerns the average sentence length). We decided to develop

corpus	#lines	#words Arabic	avrg. sent. length	#refs
BTEC train	19972	159k		1
Dev1	506	3152	6.2	16
Dev2	500	3261	6.5	16
Dev3	506	3171	6.3	16
Dev4	489	4188	8.6	7
Dev5	500	4654	9.3	7
Dev6	489	2989	6.1	6

Table 1: Characteristics of the provided BTEC data.

our system on `Dev5` and to use `Dev6` as internal test data, mainly motivated by the possibility to compare our results with those from last year's evaluation. Once the this year's Arabic test data was available we build an interpolated language model on the *source part* of the BTEC corpus and all development corpora. After analyzing the interpolation coefficients, we found evidence that this year's test data has similar characteristics than `Dev4` and `Dev5` and to less extent `Dev6`. Therefore, we decided to add the last two corpora to the training material after optimizing the system and to retrain the full system keeping all settings unmodified. This idea was already successfully proposed in previous IWSLT evaluations [12].

We have envisaged the use of additional sources of bitexts in order to improve the translation model, in particular large amounts of data that are available to build an Arabic/English translations system for the NIST task. However, initial experiments were not very concluding and that data is not used in the final system. We also investigated the possibility to increase the amount of monolingual data. This is detailed in the next section.

We have realized that the test data of this year's evaluation did contain only few punctuation marks. Many sentence had no punctuation at the end. This is in contrast to the development and test data of previous years and had a negative impact on our system that treats punctuation as a word. We tried to correct this by a simple post-processing: triggered by the first word, either a point or a question mark was added at the end of the sentence.

3.2. Training the language model

It is generally admitted that is easier to find additional monolingual resources to improve the target language model (LM). Large amounts of newspaper texts are easily available, for instance the Gigaword collection from LDC, but these texts are of course out-of-domain for the short tourism related sentences of the BTEC task. In this work, the following resources were used for language modeling:

- the English part of the BTEC training data and the development corpora (all English references were used),
- the LDC news collection of almost 3.3 billion words, known as Gigaword corpus,
- the GALE part of the 2006 NIST test set (1.1M words).

Corpus	train #words	LM size	Perplexity on Dev5
BTEC train	153k	3.3M	109.8
+BTEC dev1-4	+205k	6.5M	75.0
+Gale	+1.1M	309M	71.6
+Gigaword	+3.3G	1.1G	58.4
+ CSLM	3.4G	71M	49.3

Table 2: Characteristics of various language models. The perplexity is given for the development data (dev5).

The last corpus was included since it contains data collected from WEB blogs which may cover tourism related topics. This corpus has the LDC catalog number LDC2007E59, but we only realized after the evaluation that it was only available for research sites that had participated in the 2008 NIST evaluation. For all these texts independent 4-gram language models were built using modifier Kneser-Ney smoothing as implemented in the SRI toolkit [13]. These language models were then interpolated and merged into one LM, using the usual EM procedure to calculate the interpolation coefficients which minimize perplexity on Dev5.

The perplexities on the development data are summarized in Table 2. It is not surprising to see that adding the development data of previous evaluations improves the perplexity since this more than doubles the amount of in-domain data. The small Gale as well as the large Gigaword corpus have also a noticeable effect.[1] The continuous space language model was trained on all the available data, including the large Gigaword corpus, using a resampling algorithm [11]. This approach achieved a reduction in perplexity of more than 15% in comparison to the large back-off language model. This is inline with results obtained in previous IWSLT evaluations [9], but here both language models are trained on substantially more data.

3.3. Baseline systems

As a baseline experiment we applied our NIST Arabic/English system [1] to the BTEC task of this evaluation. It can be seen in Table 4, first line, that a system optimized on a news task does not perform very well on tourism related short sentences of the BTEC task. Note that both systems use exactly the same tokenization. Using a language model optimized on the BTEC task does improve the BLEU score by 3.8 points on Dev6, but only marginally on Dev5. Also, the BLEU score obtained by this generic system is comparable to the one obtained when using the in-domain BTEC corpus to train the translation model. On the other hand, there is a

[1] But the Gale data is not very useful anymore for language modeling once we have added the Gigaword corpus.

Translation model	Language model	Dev5	Dev6	Test08
Default tokenization:				
BTEC train	BTEC train	21.35	47.09	43.45
	BTEC train + dev1-4	22.90	45.16	42.98
	BTEC train + dev1-4 + Giga	23.18	44.15	43.70
BTEC train + dev1-4	BTEC train + dev1-4	28.15	47.33	42.71
	BTEC train + dev1-4 + Giga	28.39	47.62	44.19
BTEC train + dev1-4 + Gale	BTEC train + dev1-4 + Giga	28.17	47.82	43.52
	idem but larger word list	30.49	49.51	45.08
Improved tokenization:				
BTEC train + dev1-4	BTEC train + dev1-4 + Giga	31.20	52.10	48.09
	idem CSLM	32.38	52.42	47.46
BTEC train + dev1-4 + Gale	BTEC train + dev1-4 + Giga	31.63	50.76	47.16
BTEC train + dev1-6	BTEC train + dev1-6 + Giga	-	-	48.04
	idem CSLM	-	-	**49.39**

Table 3: Comparison of the BLEU scores of several systems. CSLM denotes the continuous space language model.

large gain when this model is applied on Dev6, showing the particularities of the IWSLT evaluations.

Translation model	Language Model	Dev5	Dev6
NIST	NIST	21.01	33.49
NIST	BTEC+Giga	21.62	37.29
BTEC	BTEC	21.35	47.09
BTEC	BTEC+Giga	23.18	44.15

Table 4: BLEU scores of a generic Arabic/English translation system (NIST task).

3.4. Adding more parallel data

Starting with this baseline, we added the development corpora of the previous IWSLT evaluations to the bitexts (Dev1 to Dev4). The Arabic source text was duplicated for each of the English reference translation. This had a huge impact on the BLEU score on Dev5, but only a minor impact on Dev6 (see Table3 upper part). This confirms our suspicion that Dev4 and Dev5 are very similar, and that Dev6 is mainly close to the BTEC training corpus. The much larger language model which includes the Gigaword corpus has only a relative modest impact on the BLEU scores, +0.25 on Dev5 and +0.29 on Dev6 respectively. This is in contrast to 22% improvement in perplexity (see Table 2). Finally, we added the Gale part of the 2006 NIST evaluation test to the parallel texts. This improved the BLEU score by almost 2 points on all data sets. Apparently these bitexts provided several new translations that were previously missing. This also explains that most of the improvements are only obtained when the new English words are added to the vocabulary and the language model is retrained. Word coverage is an important problem when translating from Arabic due to the large morphological variety of this language. In the next section we will describe an alternative tokenization that tries to tackle this problem.

3.5. Improved tokenization

There is a large body of work in the literature showing that a morphological decomposition of the Arabic words can improve the word coverage and by these means the translation quality, see for instance [10, 14, 15]. This is in particular true for under-resourced tasks like this evaluation. Most of the published work is based on the freely available tools, like the Buckwalter transliterator and the MADA and TOKAN tools for morphological analysis from Columbia University.

In this work, we used the sentence analysis module of SYSTRAN's rule-based Arabic/English translation software. Sentence analysis represents a large share of the computation in a rule-based system. This process applies first decomposition rules coupled with a word dictionary. For words that are not known in the dictionary, the most likely decomposition is guessed. In general, all possible decompositions of each word are generated and then filtered in the context of the sentence. This steps uses lexical knowledge and a global analysis of the sentences.

The lower part of Table 3 summarizes the results with this tokenization. Substantial improvements in the BLEU score of up to 4.5 BLEU were obtained. This underlines the benefit of a morphological decomposition when translating from Arabic to English. It is also striking that the additional Gale bitexts are not necessary any more. In fact, they even worsen the performance. It seems that the morphological decomposition enables better translations than adding additional bilingual out-of domain data.

In the last years, there is increasing interest in the interaction between rule-based and statistical machine translation. A popular and successful idea is *statistical post editing* [16, 17]. The principle idea is to train an SMT system to correct the outputs of a rule-based translation system.

The operation performed by the rule-based translation system could also be seen as a very good tokenization or preprocessing, that actually performs many of the translation steps. Therefore, the task of the SMT system itself is very simplified. We argue that the tokenization performed in this evaluation, which includes a global sentence analysis, could be classified somewhere in the continuum between an SMT system operating directly on the raw words and an SPE system.

This system was further improved by rescoring the n-best lists with a continuous space language model. This approach achieved gains in the BLEU score of about 1.2 BLEU points on Dev5, 0.3 points on Dev6 and 1.3 BLEU points on the current test data respectively. Finally, the last two lines of the table provide the performance on the test data when all the BTEC development data is used to learn the translation and language model. Surprisingly, this had no impact on the performance on this year's test set.

3.6. Interface with speech recognition

There are several reports in the literature showing that a careful design of the interface between automatic speech recognition (ASR) and machine translation can be important to limit the performance degradation observed when translating an automatic transcription (as opposed to a manual transcription). These works include the translation of richer data structures than the 1-best ASR output, see for instance [18] or various aspects of case, punctuation and word normalization [19, 20].

We have started working on these issues, but none of it was finally used in our system, mainly due to the fact that no native speaker of the Arabic language was available. The submitted system was only retuned on the ASR 1-best development data. Table 5 compares the BLEU score on various data sets of the text and ASR condition. We observe a degradation of about 11% relative when translating the ASR output of Dev5 and of 16% for Dev6 respectively. Unfortunately, translation of the ASR output did not work very well on this year's test data.

High word error rates of the speech recognition module favor the translation of consensus networks [18] since the oracle error rate of such data structures is usually two to three times smaller. However, this data structure is incompatible with SYSTRAN's tokenization that operates at the sentence level.

Condition	Dev5	Dev6	Test08
Text input	32.38	52.42	49.39
ASR 1-best input	28.98	43.94	38.26

Table 5: Comparison of text and speech translation.

4. Conclusion

This paper described the statistical phrase-based system developed by the LIUM laboratory for the 2008 IWSLT evaluation. We focused on the translation from Arabic to English. The system is based on the freely available Moses toolkit, complemented by two features: a morphological word decomposition based on SYSTRAN's rule-based translation system and n-best list rescoring with a continuous space language model. Both approaches aim to tackle the problem of limited amounts of in-domain training data and have obtained significant improvement of the BLEU scores in our experiments. Small improvements were also obtained by adding additional monolingual data to train the target language model.

4.1. Acknowledgments

This work has been partially funded by the French Government under the project INSTAR (ANR JCJC06 143038). We are very thankful to Jean Senellart and Jean-Baptiste Fouet from the company SYSTRAN S.A. who provided the support for the Arabic tokenization.

5. References

[1] H. Schwenk and Y. Estève, "Data selection and smoothing in an open-source system for the 2008 NIST machine translation evaluation," in *Interspeech*, 2008, p. to appear.

[2] H. Schwenk, J.-B. Fouet, and J. Senellart, "First steps towards a general purpose French/English statistical machine translation system," in *Third Workshop on SMT*, 2008, pp. 119–122.

[3] P. Koehn et al., "Moses: Open source toolkit for statistical machine translation," in *ACL, demonstration session*, 2007.

[4] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrased-based machine translation," in *HLT/NACL*, 2003, pp. 127–133.

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

[6] ——, "Discriminative training and maximum entropy models for statistical machine translation," in *ACL*, 2002, pp. 295–302.

[7] 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*, vol. 181, pp. 157–175, 2005.

[8] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *JMLR*, vol. 3, no. 2, pp. 1137–1155, 2003.

[9] H. Schwenk, M. R. Costa-jussà, and J. A. R. Fonollosa, "Continuous space language models for the IWSLT 2006 task," in *IWSLT*, November 2006, pp. 166–173.

[10] P. Lambert, M. R. Costa-jussà, J. M. Crego, M. Khalilov, J. B. M. no, R. E. Banchs, J. A. Fonollosa, and H. Schwenk, "The TALP ngram-based SMT system for IWSLT 2007," in *IWSLT*, 2007, pp. 169–174.

[11] H. Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, pp. 492–518, 2007.

[12] A. Mauser, R. Zens, E. Matusov, S. Hasan, and H. Ney, "The RWTH statistical machine translation system for the IWSLT 2006 evaluation," in *IWSLT*, 2007, pp. 103–100.

[13] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *ICSLP*, 2002, pp. II: 901–904.

[14] N. Habash and F. Sadat, "Arabic preprocessing schemes for statistical machine translation," in *NAACL*, 2006, pp. 49–52.

[15] K. Kirchhoff and M. Yang, "The university of washington machine translation system for the iwslt 2007 competition," in *IWSLT*, 2007.

[16] M. Simard, N. Ueffing, P. Isabelle, and R. Kuhn, "Rule-based translation with statistical phrase-based post-editing," in *Second Workshop on SMT*, 2007, pp. 203–206. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0728

[17] L. Dugast, J. Senellart, and P. Koehn, "Statistical post-editing on SYSTRAN's rule-based translation system," in *Second Workshop on SMT*, 2007, pp. 220–223. [Online]. Available: http://www.aclweb.org/anthology/W/W07/W07-0732

[18] N. Bertoldi and M. Federico, "A new decoder for spoken language translation based on confusion networks," in *ASRU*, 2005.

[19] D. Déchelotte, H. Schwenk, G. Adda, and J.-L. Gauvain, "Improved machine translation of text-to-speech outputs," in *Interspeech*, 2007, pp. 2441–2444.

[20] N. Bertoldi, M. Cettolo, R. Cattoni, and M. Federico, "Fbk @ iwslt 2007," in *IWSLT*, 2007.