The LIG Arabic / English Speech Translation System at IWSLT08

L. Besacier, A. Ben-Youssef, H. Blanchon

LIG Laboratory, GETALP Team
University J. Fourier, Grenoble, France
Laurent.Besacier@imag.fr

Abstract

This paper is a description of the system presented by the LIG laboratory to the IWSLT08 speech translation evaluation. The LIG participated, for the second time this year, in the Arabic to English speech translation task. For translation, we used a conventional statistical phrase-based system developed using the *moses* open source decoder. We describe chronologically the improvements made since last year, starting from the IWSLT 2007 system, following with the improvements made for our 2008 submission. Then, we discuss in section 5 some post-evaluation experiments made very recently, as well as some on-going work on Arabic / English speech to text translation. This year, the systems were ranked according to the (BLEU+METEOR)/2 score of the primary ASR output run submissions. The LIG was ranked 5th/10 based on this rule.

1. Introduction

This paper is a description of the system presented by the LIG laboratory to the IWSLT08 speech translation evaluation. The LIG only participated in the Arabic to English speech translation task. For translation, we used a statistical phrase-based system developed using the *moses* open source decoder.

Section 2 of this paper gives a short overview of the data and tools we used to build our speech translation system. Then, we describe chronologically the improvements made since last year, starting from the IWSLT 2007 system (described in *section 3*), following with the improvements made for our 2008 submission (*section 4*). Then, we discuss in *section 5* some post-evaluation experiments made very recently, as well as some on-going work on Arabic / English speech to text translation. Finally, section 6 quickly concludes this work.

2. Task, data and tools

This year, the LIG laboratory participated for the second time in the Arabic – English (AE) speech translation task. We have used the data provided by the IWSLT08 organizers and a few publicly available additional data.

For training the translation models, the *train* part of the IWSLT08 data was used[1] (a training corpus of 19972 sentence pairs). As development data, we used several subsets provided: the *dev4* subset, made up of 489 sentences, which corresponds to the IWSLT06 development data (we will refer, in the rest of the paper, to *dev06* for this data set); the *dev5* subset, made up of 500 sentences, which corresponds to the IWSLT06 evaluation data (we will refer, in the rest of the paper, to *tst06* for this data set) ; and the *dev6* subset, made up of 500 sentences, which corresponds to the IWSLT07 evaluation data (we will refer, in the rest of the paper, to *tst07* for this data set). The tuning of the MT model parameters (minimum error rate training) was systematically done on the *dev06* subset.

As additional data, we first used an Arabic / English bilingual dictionary of around 84k entries. This dictionary can be found online[2]. For English LM training, we also used out-of-domain corpora taken from the LDC's Gigaword corpus[3].

Our baseline speech translation system was built using tools available in the MT community:

-GIZA++ [1] was used for the alignments,

-The *moses*[4] decoder (and the training / testing scripts associated) was used (2008-02-20 release),

-SRILM [2] was used to train the LMs and to deal with ASR word graphs,

-The Buckwalter morphological analyzer[5] was used for Arabic word segmentation,

-All the performances reported in this paper are BLEU [3] and sometimes METEOR [4]. This year, the systems were ranked according to the (BLEU+METEOR)/2 score of the primary ASR output run submissions.

3. Overview of the 2007 system

More details on the LIG IWSLT 2007 system can be

[1] preliminary experiments have shown that adding dev1, dev2 and dev3 sets to the training data do not significantly improve the performance, so we did not use these data sets

[2] http://freedict.cvs.sourceforge.net/freedict/eng-ara/

[3] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

[4] *Moses* open source project: http://www.statmt.org/moses

[5] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49

found in [5].

3.1. MT system

Our 2007 system was trained on the 20k *train* bitext provided, concatenated to the bilingual dictionary of 84k entries, described in section 2, leading to a total of 103497 lines for training. The *moses* training script (default options) was used to build a phrase translation table from the bitext. The Arabic part of the bitext was systematically segmented using the Buckwalter morphological analyzer, in order to increase vocabulary coverage. On the English side, we removed punctuation and case (both pieces of information are further restored after translation using *hidden-ngram* and *disambig* from the SRILM toolkit [2]).
For English language modeling, we used both in-domain (English part of the *train* bitext) and out-of-domain (LDC's Gigaword corpus) to train the English LM. The interpolation weights (0.7/0.3) optimize the perplexity on the dev06 corpus.

3.2. Use of ASR output

Since we are using the *moses* open source decoder, we were able to exploit confusion networks (CN) as interface between speech recognition and machine translation [6]. CN permit to represent a huge number of transcription hypotheses while leading to efficient search algorithms for statistical machine translation.

However, one major problem we had to deal with was the fact that the word graphs provided for IWSLT do not have necessarily word decomposition compatible with the word decomposition used to train our MT models. Thus, a word lattice decomposition process was proposed in 2007, to make the lattices (and then the word CN) compatible with our own level of decomposition. This process is described more precisely in [5] and [7].

In a few words, the decomposition algorithm can be described with the following steps:

1. Based on a word/sub-word dictionary or a morphological analyzer, all decompoundable words in the word lattice are identified.

2. Each of these words is decomposed into a sequence of sub-words that depends on the number of sub-words in the word. Some new nodes and links are then inserted in the word lattice.

3 For each new decomposed sub-word in the current word lattice, the new acoustic score and the duration are modified: the duration and the acoustic scores of the initial word are proportionally divided into sub-words duration and scores as a function of the number of graphemes in the sub-words.

4. An approximation is made for the LM score: the LM score corresponding to the first sub-word of the decomposed word is equal to the LM score of the initial word, while we assume that after the first sub-word, there is only one path to the last sub-word of the word (so the following LM scores are made equal to 0).

5. Finally, the new subword lattice is converted into a CN using an algorithm similar to [8].

3.3. System 2007 performance

Table 1 gives an overview of the performances of our IWSLT 2007 system. This system was ranked 7th/14 for IWSLT07 AE ASR task.

Table 1: *2007 system performance (BLEU score).*

	dev06	tst06	tst06 (asr)	tst07	tst07 (asr)
Sys. 2007	0.2948	0.2271	0.2253	0.4263	0.3904

4. Experiments made for 2008 system submission

4.1. Improvement of the 2007 system

Table 2 gives an overview of the experiments made for 2008 system submission. All lines of this table correspond basically to the IWSLT07 system modified with an increasing number of options. For instance, the third line (+mbr) means that both *drop-unknown* and *mbr* options were used.

We tried the following improvements to our 2007 system :

- during the translation process, unknown words are copied verbatim to the output. We tried to drop unknown words in order to optimize BLEU even if it is not clear, from human judgements point of view, if this might help or not. Dropping unknown words resulted in better BLEU scores (2 points in average).

- we also tried to used Minumum Bayes Risk (MBR) decoding [9]. This statistical approach aims to minimize expected loss of translation errors under loss functions that measure translation performance. According to the *moses* documentation, the loss function used by the decoder is a smoothed BLEU score. The benefit of MBR is not clear : we observe improvements on tst06 and tst07 while the performance decreases on dev06.

- we also improved our repunctuation system; it is still based on the use of *hidden-ngram* tool from SRI-LM, but the punctuation marks which are at the end of the sentences are also appended at the beginning of the English sentences used to train the punctuated LM. This means that during the re-punctuation process, a punctuation mark is hypothesized both at the beginning and at the end of the sentence. In case of disagreement, we keep, as final punctuation, the punctuation mark which is obtained at the beginning (which was not the

case in our 2007 system) since the first word of a sentence is generally more informative than the ending word for re-punctuation. This new re-punctuation process consistently increased the BLEU score on our different development sets.

- all the latter improvements concerned decoder options and post-processing of the translation hypothesis. They did not include any retraining of the translation models. So, we tried to train lexicalized reordering models. Moses allows the combination of different reordering types (we used *msd-bidirectional-fe* option[1]). The log-linear parameters were retuned on dev06 set and the results can be found on the last line of *Table 2*. Again, the results are encouraging on dev06 and tst06 data while we observe degradation on tst07 which has shorter sentences. Because of this latter result on tst07, we finally decided not to use this reordering model for the IWSLT 2008 submission.

Table 2: *Experiments (BLEU) for 2008 submission (the final system submitted to IWSLT08 is put in bold)*

	dev06	tst06	tst06 (asr)	tst07	tst07 (asr)
Sys. 2007	0.2948	0.2271	0.2253	0.4263	0.3904
+drop-unknown	0.3139	0.2442		0.4554	
+mbr	0.3083	0.2446		0.4569	
+new repunct	**0.3134**	**0.2514**	**0.2290**	**0.4775**	**0.4194**
+lex. reordering	0.3206	0.2573	0.2302	0.4688	0.4125

4.2. IWSLT 2008 LIG system results

The system submitted corresponds to the one in bold in Table 2. The official results obtained this year can be found in Table 3.

Table 3: *LIG official results for IWSLT08*

	(bleu+meteror)/2	bleu	meteor
verbatim	0.5674	0.4595	0.6752
ASR	0.5022	0.3931	0.6113

This year, the systems were ranked according to the (BLEU+METEOR)/2 score of the primary ASR output run submissions. The LIG was ranked 5th/10 based on this rule. The results of the subjective evaluation recently made available by the organizers confirmed this ranking (5th/10).

[1]see http://www.statmt.org/moses/?n=Moses.AdvancedFeatures#ntoc1 for more details.

5. Post-evaluation experiments

5.1. New Arabic segmenter

We tried to use SVM-POS, a free Arabic segmenter (which also performs POS tagging) developed at Columbia University[2]. It is based on Support Vector Machines trained on the Arabic TreeBank. This system is an adaptation to Arabic of the YamCha software initially developed for Japanese and then English[3].
This segmenter was used to pre-process the Arabic training data and the translation table was retrained using the new parallel data obtained (no change was made on the English side). While the Buckwalter segmentation leads to an average of 10.6 segment units / sentence, the ASVM is much less "aggressive": 7.5 units / sentence in average (no segmentation at all gives 6.7 units / sentence).
The performance of this new system is presented in Table 4, and compared to our 2008 submitted system (bold line of Table 2).

Table 4: *Comparison and combination of the two Arabic segmentation methods (Buckwalter versus SVM-POS).*

	dev06	tst06	tst07
Buckwalter	0.3134	0.2514	0.4775
SVM-POS	0.2757	0.2267	0.4944
Combination (post-edition)			0.5191

Table 4 shows that the problem of Arabic segmentation is very tricky for machine translation with sparse data. The efficiency of the segmentation seems very dependent of the test set. Post-editing the *tst07* outputs by selecting the best one (between both segmenters) lead to a significant improvement (0.5191 BLEU). Consequently, since no segmentation is really better than another, it might be interesting to use both during decoding (or in a re-scoring framework).
In fact, we analyzed more deeply the translation output obtained (on tst07), as well as the Arabic input segmented by both segmenters. This can be seen in Table 5 where correct segmentations and correct translations are put in bold. While the correct segmentation may lead to the correct translation (cases 1, 2 and 7), we also observed some sentences for which none of the segmentations is correct (cases 3 and 4). In that case, one translation output might still be correct. One reason may be that an incorrect segmentation can remain consistent with the segmentation applied on the training data (bad segmentation on the training data will probably lead to bad alignments but these errors may be somehow recovered during the phrase-table

[2] http://www1.cs.columbia.edu/~mdiab/
[3] http://www.chasen.org/~taku/software/yamcha/

construction). Finally, we also observe cases (5 and 6) where a correct segmentation does not necessarily lead to the best translation output.

So, when dealing with sparse data, like in IWSLT, it seems that the Arabic segmentation heavily influences the translation quality: segmentation affects the translation models (alignments, phrase table) as well as the translation input. We believe that simultaneously using multiple segmentations is a promising way to improve machine translation of Arabic and our next efforts will be dedicated to this.

Table 5: *Qualitative comparison of the two Arabic segmentation methods (Buckwalter versus SVM-POS). Correct segmentations and translations are put in bold.*

	Seg. buckwalter	Seg. SVM-POS
1	**ما مقاس ك** **what size do you wear**	ما مقاسك what your size
2	س ي ستغرق ذلك حوالى ثلاث ون دقيق ة it will take that thirty minutes	**سيستغرق ذلك حوالى ثلاثون دقيقة** **it will take about thirty minutes**
3	أيمكنني استعمال هاتف ك **can i use your telephone**	أيمكنني استعمال هاتف ك can i your phone
4	عند ي حمى منذ ال بارح ة i have a fever since	عندي حمى منذ البارحة **i have a fever since yesterday**
5	**هل ب إمكان ي ال تحدث إلى ال سيد كارتر** can i talk to mr	هل ب إمكان ي التحدث إلى السيد كارتر **may i speak to mr carter**
6	صدم ت ني سيار ة **i was hit by a car**	**صدمت ني سيارة** was hit by a car
7	**عند ي ألم في ال أذن** **i have a pain in my ear**	عند ي ألم في الأذن i have a pain in turn

5.2. Towards using POS tags and factored models

Before trying to use the POS tags obtained by SVM-POS, we tried to evaluate this tagger on spoken data (SVM-POS was trained on different type of data). The 100 last sentences of the *train* set were manually tagged for evaluation purpose (using the same tag set as SVM-POS). The tag error rate was high (9%) and we decided to improve this POS tagger before trying any experiment with factored models. For this, we manually corrected 20% of the *train* data set, tagged with SVM-POS. Then, we built a n-gram-based POS tagger, using the corrected corpus and the *disambig* function of SRI-LM. This new tagger was evaluated on the same data set (100 last sentences of the *train*). The tag error rate was only 1.7% with this new POS-tagger.

We are currently working on using this POS tagger to improve our translation models (using factored models). In a preliminary experiment reported here, the Arabic words are translated into English words and lemmas. Simultaneously, the Arabic POS are translated into English POS. The final English surface form is then generated given the obtained English lemma and POS, except for unknown lemmas where the direct English word translation is used. Note that for these latter experiments, we used train+dev1-3 for training (using only the first english sentence on dev ; without any bilingual dictionary) ; this explains that the results in the case *unfactored* are different to the ones given in *table 4* (ligne 2).

Again, these preliminary results (presented in *Table 6*) are not conclusive, showing improvements or degradations (compared to unfactored models), according to the evaluation set used.

Table 6: *Comparison of factored and un-factored models.*

	dev06	tst06	tst07
Unfactored	0.2799	0.2465	0.4714
Factored	0.2835	0.2395	0.5024

6. Conclusions

This paper was a description of the system presented by the LIG laboratory to the IWSLT08 speech translation evaluation. The LIG only participated in the Arabic to English speech translation task. The main features of this system are the following:

-adding out-of-domain training corpora for English LM;

-concatenating a bilingual dictionary to the bitext available for training;

-using ASR word graphs to perform speech translation by direct confusion network decoding;

-using different Arabic segmentation techniques, which is potentially interesting for further recombination.

Future work will focus on taking advantage of our efficient Arabic POS tagger and using multiple segmentations during the decoding process.

7. References

[1] Och, F. J. and Ney, H., “A Systematic Comparison of Various Statistical Alignment Models”, Computational Linguistics, vol. 29, no. 1, pp. 19-51, March 2003.

[2] Stolcke, A., “SRILM - An Extensible Language Modeling Toolkit”, ICSLP’02, vol. 2, pp. 901-904, Denver, Colorado, September 2002.

[3] Papineni, K., Roukos, S., Ward, T., and Zhu, W., “BLEU: A method for automatic evaluation of machine translation”, ACL’02, pp. 311-318, Philadelphia, USA, July 2002.

[4] Lavie, A., A. Agarwal. "METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments", Proceedings of Workshop on

Statistical Machine Translation at the 45th Annual Meeting of the ACL (ACL-2007), Prague, June 2007.

[5] L. Besacier, A. Mahdhaoui, V-B Le, « The LIG Arabic / English Speech Translation System at IWSLT07 » IWSLT07. Trento. Italy. October 2007.

[6] Bertoldi, N., Zens, R. and Federico, M., "Speech Translation by Confusion Network Decoding", ICASSP'07, vol. 4, pp. 1297-1300, Honolulu, Hawaii, April 2007.

[7] V-B. Le, S. Seng, L. Besacier, B. Bigi. « Word/Sub-word lattices decomposition and combination for Speech Recognition » IEEE ICASSP 2008. Las Vegas, USA, 2008.

[8] Mangu, L., Brill, E., and Stolcke, A., "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks", Computer Speech and Language, vol. 14, no. 4, pp. 373-400, 2000.

[9] S. Kumar and W. Byrne, "Minimum Bayes-Risk Decoding for Statistical Machine Translation" *Proceedings of HLT-NAACL 2004*, May 2004, Boston, MA, USA.