The LIG Arabic / English Speech Translation System at IWSLT07

L. Besacier, A. Mahdhaoui, V-B Le

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 IWSLT07 speech translation evaluation. The LIG participated, for the first 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. Our baseline MT system is described and we discuss particularly the use of an additional bilingual dictionary which seems useful when few training data is available. The main contribution of this paper concerns the proposal of a lattice decomposition algorithm that allows transforming a word lattice into a sub word lattice compatible with our MT model that uses word segmentation on the Arabic part. The lattice is then transformed into a confusion network which can be directly decoded using *moses*. The results show that this method outperforms the conventional 1-best translation which consists in translating only the most probable ASR hypothesis. The best BLEU score, from ASR output obtained on IWSLT06 evaluation data is 0.2253. The results confirm the interest of full CN decoding for speech translation, compared to traditional ASR 1-best approach. Our primary system was ranked 7/14 for IWSLT07 AE ASR task with a BLEU score of 0.3804.

1. Introduction

This paper is a description of the system presented by the LIG laboratory to the IWSLT07 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. Our baseline MT system is described in *section 3* where we discuss particularly the use of an additional bilingual dictionary which seems particularly useful when few training data is available. The main contribution of this paper is presented in *section 4* where a lattice decomposition algorithm is proposed. It allows transforming a word lattice into a sub word lattice compatible with our MT model that uses a segmentation of Arabic words into prefix, stem and suffixes. The lattice is then transformed into a confusion network which can be directly decoded using *moses*. The results presented in *section 4* show that this method outperforms the conventional 1-best translation which consists in translating only the most probable ASR hypothesis. The best BLEU score, from ASR output obtained on IWSLT06 evaluation data is 0.2253 which is very near the performance of the best system presented in 2006 at IWSLT evaluation.

2. Task, data and tools

This year, the LIG laboratory participated for the first time in the Arabic – English (AE) speech translation task. We have used the data provided by the IWSLT07 organizers and a few publicly available additional data.

For training the translation models, the *train* part of the IWSLT07 data was used (a training corpus of 20k sentence pairs). As development data, we used two subsets of the development data 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); and 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). The tuning of the MT model parameters 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[1]. The way this dictionary is used will be explained later in this paper. For English LM training, we also used out-of-domain corpora taken from the LDC's Gigaword corpus[2].

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

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

-The *moses*[3] decoder (and the training / testing scripts associated) was used,

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

-The Buckwalter morphological analyzer[4] was used for Arabic word segmentation,

-All the performances reported in this paper are BLEU [3] scores calculated using the scoring script provided by NIST.

3. MT experiments from verbatim transcriptions

Our baseline phrase-based system was trained on the 20k bitext provided. The *moses* training script was used to build a phrase translation table from the bitext. The first English LM

[1] http://freedict.cvs.sourceforge.net/freedict/eng-ara/
[2] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05
[3] *Moses* open source project: http://www.statmt.org/moses
[4] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002L49

was trained on the English part of the bitext. The Arabic part of the bitext was systematically segmented using the Buckwalter morphological analyzer, in order to increase vocabulary coverage. It is important to note that the Buckwalter analyzer can give different decompositions possible for a single word; we decided to keep systematically the first proposed solution. We are aware that this method is sub-optimal but it has the advantage to be simple and consistent.

3.1. Restoring punctuation and case information

Two separated punctuation and case restoration tools were built using *hidden-ngram* and *disambig* commands of the SRILM toolkit [2]. We built these restoration tools following the instructions provided to the participants for IWSLT06. Table 1 shows the performance of different MT systems that deal differently with the punctuation and case restoration problems:
(1) is a system trained on the bitext *with* punctuation and case; however, in the translated output, both informations are removed before being restored using SRILM tools;
(2) is a system trained on the bitext *without* punctuation and case; in the translated output, both informations are restored using SRILM tools;
(3) is similar to (1) except that for training data, the case and punctuation informations were removed before being restored using SRILM tools.

Table 1: *Different experiments for punctuation and case restoration.*

	(1) train with case & punct	(2) train without case & punct	(3) train with restored case & punct
dev06	0.2341	0.2464	0.2298
tst06	0.1976	0.1948	0.1876

From this table, we see that option (2) seems to be the best. Removing case and punctuation information reduces the vocabulary size, on the English side, and thus reduces the complexity of the models. For the next experiments, the MT models will be trained on data were the punctuation and case informations are removed. These informations will be ultimately restored before scoring.

3.2. Using out-of-domain data for English LM training and using minimum error rate training

As suggested in [4] for IWSLT06, we used both in-domain (English part of the 20k bitext) and out-of-domain (LDC's Gigaword corpus) to train the English LM. Table 2 compares the performance of a baseline LM trained only on the English part of the bitext, with the performance of LM resulting from the interpolation of an in-domain LM (weight 0.7) and an out-of-domain LM (weight 0.3). The out-of-domain LM was trained on the LDC Gigaword and the vocabulary was limited to the most frequent 20k words (bigger out-of-domain LMs did not show significant improvement).

Table 2: *Benefit of out-of-domain data for English LM and use of MERT to tune the MT model parameters.*

	In domain LM No MERT	Interpolated in-domain and out-of-domain LM No MERT	Interpolated in-domain and out-of-domain LM MERT on dev06
dev06	0.2464	0.2535	0.2674
tst06	0.1948	0.2048	0.2050

Table 2 shows the benefit of out-of-domain data for English LM and also the benefit of minimum error rate training (MERT). In the next experiments, the system used will be the one of the last column of table 2 (out-of-domain data + MERT).

3.3. Use of a bilingual dictionary

The bilingual dictionary of 84k entries, described in section 2, was concatenated to the training data and a phrase table was re-obtained using this new training input. The MT parameters were tuned using a new MERT procedure. The performance of this new model is reported in the last column of table 3.

Table 3: *Use of a bilingual dictionary.*

	No bilingual dict.	Use of a bilingual dict.
dev06	0.2674	0.2948
tst06	0.2050	0.2271

The results suggest that the use of a bilingual dictionary may be particularly interesting when few data (only 20k) is available to train the MT models. This last system will be the system used to translate verbatim transcriptions. The next section describes our speech translation experiments using ASR output. It is, from our point of view, the main contribution of the LIG submission this year.

4. Speech translation by confusion network decoding

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 [5]. 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 IWSLT07 did not have necessarily word decomposition compatible with the word decomposition used to train our MT models. Thus, a word lattice decomposition process was needed, to make the lattices (and then the word CN) compatible with our own level of decomposition used. This process is described in the next sections.

4.1. From lattices to confusion networks

CNs can be obtained from lattices by means of the *lattice-tool* package included in the SRILM toolkit [2], which implements the lattice-to-CN converting algorithm described in [7].

Figure 1 illustrates an example (in English) of a word

lattice outputted from a speech recognizer and its corresponding confusion network. In this example, 'CANNOT' and 'CAN' are merged in an alignment in the confusion network although their time durations could be different. This alignment creates a deletion (labeled by 'ε') in the next alignment.

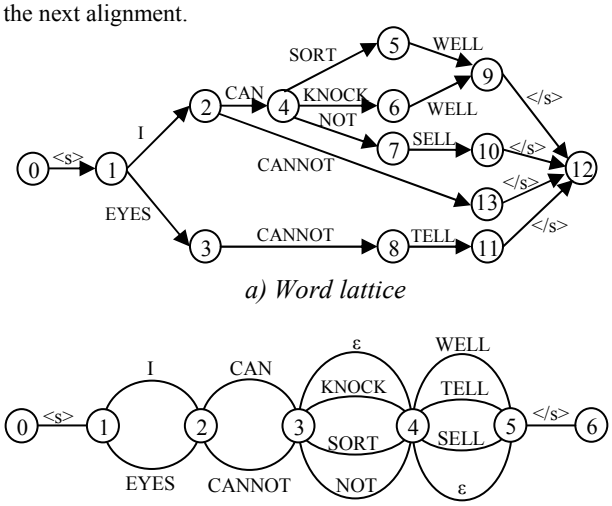


Figure 1: *Word lattice and word-based confusion network.*

As already said, to deal with a language like Arabic, the use of classical word units in ASR and MT can be replaced by subword units like morphemes [6]. Such decomposition can reduce the high out-of-vocabulary rate and improve the lack of text resources in statistical language modeling. If a word segmenter is already available (like the Buckwalter morphological analyzer), applying such decomposition is obvious on word strings (verbatim transcriptions, N-best lists). It is however more problematic when such decomposition must be applied to a word lattice, at the output of an ASR system. The problem, in that case, can be formulated as following: how the word lattice should be modified when words are segmented into subword units?

4.2. Word lattice decomposition

In fact, a word lattice can already be decomposed using the latest version (v.1.5.2) of the lattice tool in the SRILM toolkit [2]. By using the *-split-multiwords* option of the *lattice-tool*, we can split a node with words in the lattice into a sequence of subword nodes (morphemes in our case with Arabic). In that case, the first node in this sequence keeps all the information (acoustic score, language score, duration) from the original node while the other inserted nodes have null scores and zero-duration. However, since the used lattice-to-CN converting algorithm (proposed in [7]) takes into account the duration of each word, this word splitting could cause some error during the converting process. Figure 2 illustrates a subword lattice (where CANNOT is segmented into CAN and NOT) which is converted by the SRILM *lattice-tool* from the word lattice presented in figure 1. To decompose the word lattice in this example, two new nodes 14 and 15 are added in the lattice and they have the same time value with nodes 8 and 13. This decomposition causes a wrong alignment in the confusion network: the word 'NOT' in the link 13-15 is aligned with 'WELL", 'SELL' and 'TELL' (figure 2.b).

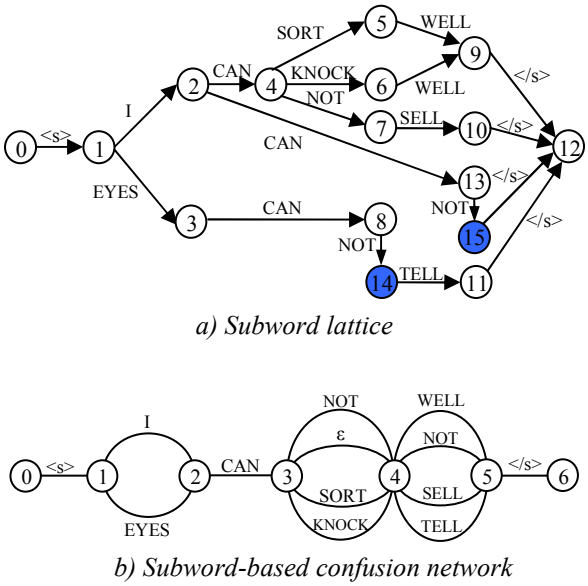


Figure 2: *Subword lattice converted from word lattice by SRILM lattice-tool (-split-multiwords option).*

In our work, we propose a new algorithm for splitting a word into a sequence of subword units (or morphemes). Depending on the number of decomposed subword units, some new nodes are also added to the lattice. The subword labels are assigned to new links. The difference with our algorithm is that the duration of each new subword unit is calculated as a function of the number of its graphemes. For each original link in the lattice, the acoustic score is also distributed to new links in proportion to the durations of new assigned subword units.

More precisely, the decomposition algorithm can be described with the following steps:

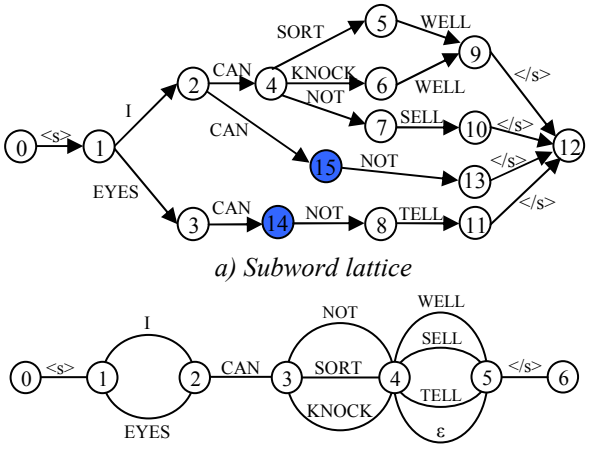
-identify the arcs of the graph that will be split: all the arcs corresponding to decompoundable words (in our case, the decomposition is based on the Buckwalter morphological analyzer) will be split;

-then, each arc to be split is decomposed into a number of arcs that depends on the number of subword units that compose the initial word; for instance, the arc between node 3 and 8 of *figure 1*, will be decomposed into two arcs, and a new node (node 14 of *figure 3*) will be inserted in the graph;

-the start / end times of the arcs are then modified according to the number of graphemes into each subword unit: for instance, if the arc between nodes 3 and 8 (of *figure 1*) starts at time t_1 and ends at time t_2, then the new arc between nodes 3 and 14 (of *figure 3*) will starts at time t_1 and ends at time $t_1+(t_2-t_1)/2$; similarly, the new arc between nodes 14 and 8 (of *figure 3*) will starts at time $t_1+(t_2-t_1)/2$ and ends at time $t2$;

-similarly, the acoustic scores are approximately modified according to the number of graphemes into each subword unit: for instance, for a word split into two subword units of equal length (like the insertion of node 14 in *figure 3*), the initial acoustic score *a* will become *a/2* for both new arcs;

-finally, concerning the language model scores, we make an approximation that the LM score corresponding to the first subword of the decomposed word is equal to the initial LM score of the word, while we assume that after the first subword, there is only one path to the last subword of the word (and then the following LM scores are made equal to 0); for instance, on *figure 3*, the LM score of the arc between nodes 3 and 14 will be the same as the LM score between nodes 3 and 8 of *figure 1*, while the LM score between nodes 14 and 8 will be set to 0.



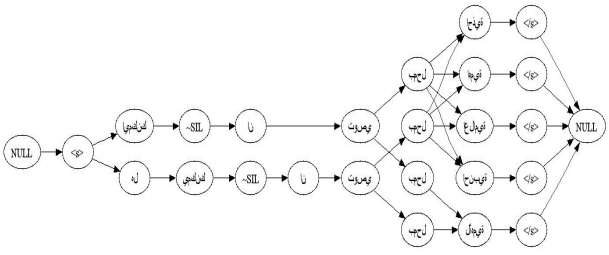
a) Subword lattice

b) Subword-based confusion network

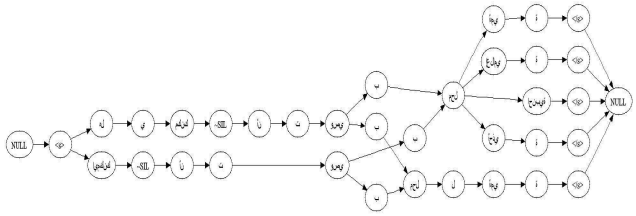
Figure 3: *Subword lattice obtained with our decomposition algorithm and the associated subword-based confusion network.*

Figure 3 presents a new subword lattice and the resulting confusion network converted. We note that the words 'CANNOT' in the link 2-13 and the link 3-8 are decomposed into two pairs of syllables 'CAN' and 'NOT' by adding two new nodes in the lattice (node 14 and node 15). The duration of 'CAN' in the new link 2-15 and 'NOT' in the new link 15-13 are equal due to the same number of graphemes. The new confusion net obtained in that case seems more reasonable than the ones shown in *figure 1* and *figure 2*.

Since we worked on Arabic ASR outputs for IWSLT07, the *figure 4* also gives an example of word-based and subword based lattices in Arabic (the latter being obtained after applying our word lattice decomposition algorithm).



a) Word lattice



a) Subword lattice

Figure 4: *Word lattice and Subword lattice obtained with our decomposition algorithm in Arabic (the English translation of the utterance is "could you recommend me with a shoe shop?").*

4.3. Speech translation experiments

As explained in the previous paragraph, we have developed a word lattice decomposition tool that accepts, as input, an ASR word lattice, as well as a dictionary of words and their decomposition into subword units. This decomposition is obtained with the baseline Buckwalter morphological analyzer (as already explained in the first paragraph of *section 3*). Then, a new lattice made up of subword units is obtained, as output. This new lattice can be converted to a subword confusion network, compatible with the decomposition made during the training of the MT models.

Table 4 is a summary of our speech translation performance using :

(1) verbatim transcription as input (segmented using the buckwalter morphological analyzer);

(2) 1-best ASR as input (segmented using the buckwalter morphological analyzer);

(3) taking the 1-best word sequence from the word CN, as input (segmented using the buckwalter morphological analyzer) ; this corresponds to consensus decoding;

(4) taking the subword CN as input (obtained after applying our word lattice decomposition algorithm); this corresponds to full CN decoding; it is important to note that all the parameters of the log-linear model used for the CN decoder were retuned on *dev06* set (since an additional parameter, corresponding to the CN posterior probability is added in that case, as described in [5]).

Table 3: *Speech translation experiments.*

	(1) verbatim	(2) 1-best	(3) cons-dec	(4) full-cn-dec
dev06	0.2948	0.2469	0.2486	0.2779
tst06	0.2271	0.1991	0.2009	0.2253

These results show that the full CN decoding, using our lattice decomposition algorithm, outperforms the conventional 1-best translation which consists in translating only the most probable ASR hypothesis. The best BLEU score, from ASR output, obtained on IWSLT06 evaluation data is 0.2253.

5. IWSLT07 submission results

Table 4 gives our results for the IWSLT07 evaluation (AE task). The results confirm the interest of full CN decoding for

speech translation, compared to traditional ASR 1-best approach. Our system was ranked 7/14 for this year's AE ASR task.

Table 4: *IWSLT07 results of LIG laboratory.*

	clean verbatim	ASR 1-best	ASR full-cn-dec
Eva07	0.4135	0.3644	0.3804

6. Conclusions

This paper was a description of the system presented by the LIG laboratory to the IWSLT07 speech translation evaluation. The LIG only participated in the Arabic to English speech translation task. The experiments reported here show the benefit of the following techniques:

-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;

-in the case of MT models trained on data segmented into sub word units, use of a lattice decomposition algorithm, to make the ASR output compatible with the existing MT models.

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] Lee, Y-S., "IBM Arabic-to-English Translation for IWSLT 2006", IWSLT'06 Worshop, pp. 45-52, Kyoto, Japan, November 2006.

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

[6] Afify, M., Sarikaya, R., Jeff Kuo, H-K., Besacier, L. and Gao, Y., "On the use of morphological analysis for dialectal Arabic Speech Recognition", Interspeech'06, pp. 277-280, Pittsburgh, PA, September 2006.

[7] 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.