The TÜBİTAK-UEKAE Statistical Machine Translation System for IWSLT 2007

Coşkun Mermer, Hamza Kaya, Mehmet Uğur Doğan

National Research Institute of Electronics and Cryptology (UEKAE)
The Scientific and Technological Research Council of Turkey (TÜBİTAK)
Gebze, Kocaeli 41470, Turkey
{coskun,hamzaky,mugur}@uekae.tubitak.gov.tr

Abstract

We describe the TÜBITAK-UEKAE system that participated in the Arabic-to-English and Japanese-to-English translation tasks of the IWSLT 2007 evaluation campaign. Our system is built on the open-source phrase-based statistical machine translation software *Moses*. Among available corpora and linguistic resources, only the supplied training data and an Arabic morphological analyzer are used in the system. We present the run-time lexical approximation method to cope with out-of-vocabulary words during decoding. We tested our system under both automatic speech recognition (ASR) and clean transcript (clean) input conditions. Our system was ranked first in both Arabic-to-English and Japanese-to-English tasks under the "clean" condition.

1. Introduction

Phrase-based statistical machine translation (SMT) has been the most actively researched machine translation approach in recent years. Since SMT is a corpus-based approach rather than relying on complicated linguistic rules, system development has been relatively easier and much less language-dependent. This has facilitated the emergence of publicly-available SMT development tools, which further lowered the entrance barrier to the research field. The International Workshop on Spoken Language Translation (IWSLT) series [1-3] have in addition provided common corpora and evaluation, creating an opportunity to comparatively evaluate machine translation technologies on the same test bed. This year's workshop [4] further encouraged sharing of linguistic resources, emphasizing the cooperative nature of the workshop.

In this paper, we report on our first-time participation in the IWSLT evaluation campaign. Among the four language tracks in IWSLT 2007, we participated in the Arabic-to-English and Japanese-to-English translation tasks. We built our baseline system using the open-source phrase-based statistical machine translation software *Moses*. Among shared corpora and tools, we used only the supplied training data and the Buckwalter Arabic Morphological Analyzer. In order to cope with previously unseen words during decoding, we present the run-time lexical approximation method, which replaces an out-of-vocabulary word with the closest known word having the same feature. We describe three different word-level features used in our system. For each language, both clean transcription and automatic speech recognition (ASR) outputs are translated, though only the 1-best hypotheses are used in the latter case.

The rest of the paper is organized as follows. Section 2 describes the corpora and tools used in our system. Corpus preprocessing and system training are described in Sections 3 and 4, respectively. The run-time lexical approximation method is presented in Section 5. Results and discussion are provided in Section 6, followed by the conclusion.

2. Experimental setup and resources

Our system is trained on the supplied BTEC corpus [5], whose characteristics are shown in Table 1. *train* and *devsets1-3* are used for training, *devset4* and *devset5* are reserved for performance testing and tuning.

Table 1: Supplied training corpus characteristics

Corpus name	Number of English references	Number of source segments	Avg. num. of tokens per segment (EN)	Source OOV rate
train	1	AR: 19972 JP: 39953	9.1	-
devsets1-3	16	1512	8.1	-
devset4	7	489	13.4	AR: 10.1% JP: 1.8%
devset5	7	500	14.6	AR: 10.9% JP: 2.3%

The word alignments are generated by *GIZA++* [6] using IBM Model-4 [7] and the phrase-based translation model generation and decoding is performed by *Moses* [8].

We have used the SRI language modeling toolkit [9] for (i) generating the target language model used by the decoder, and (ii) source language punctuation modeling and run-time punctuation insertion before translating ASR outputs.

As an aid in translating Arabic out-of-vocabulary words, the Buckwalter Arabic Morphological Analyzer Version 1.0 is used [10].

3. Corpus handling strategy

We performed Buckwalter transliteration on all Arabic training and test corpora. All training and decoding are done on tokenized, punctuated and lowercased data sets. This setup requires source-side punctuation insertion (for ASR output condition) and target-side case restoration before and after decoding, respectively.

For punctuation restoration in the case of ASR input, we used the SRILM tools as described in [11], which can be summarized as follows:

1) Preprocess the punctuated training set.
2) Train a punctuated language model.
3) Insert punctuation using `hidden-ngram`.
4) Postprocess punctuation decisions.

During system development, the punctuator failed to recognize the internal sentence boundaries in most of the multi-sentence segments in *devsets4-5*. One reason could be that the segments in *devsets4-5* are usually longer (see Table

1) and contain relatively more sentences than those in the training set. Therefore, in order to train the punctuation restorer with more occurrences of segment-internal sentence boundaries, we artificially merged *N* segments in the training set and thus trained the punctuator. In our system, we set *N* = 10. The results during the development experiments are shown in Table 2.

Table 2: Development BLEU scores with automatic punctuator trained on original vs. 10-merged segments

Task	*N*	*devset4*	*Devset5*
AR-EN	1	24.32	20.23
	10	**24.95**	**20.66**
JP-EN	1	15.59	14.26
	10	**17.82**	**16.12**

The smaller improvement in AR-EN task compared to JP-EN can be attributed to two factors: (i) The Arabic corpus has two types of sentence boundary punctuation '.' and '؟', while the Japanese corpus has only one '。'. (ii) The predictive power of the last and first words in a sentence is higher in Japanese than in Arabic.

For post-translation automatic case restoration, we used the *Moses* recasing tool according to the procedure outlined in [12].

4. Training

As shown in Table 1, *devsets1-3* have 16 English reference segments per source segment. In order to obtain better phrase alignments and to increase the system's target phrase coverage, all reference segments in *devsets1-3* were included in the training set with their corresponding source segments. We also performed this augmentation when training our language models. In both cases, the segments in the two data sets *train* and *devsets1-3* were given equal weight.

The *N*-gram English language models were trained with modified Kneser-Ney discounting and interpolation. 3-gram and 4-gram English language models were used for AR-EN and JP-EN translation, respectively.

Before translation model training, the multi-sentence segments are automatically split if the number of sentence boundary punctuations in both the source and reference segments are equal, so as to prevent erroneous word alignments across sentence boundaries. The resulting number of segments in each corpus are shown in Table 3.

Table 3: Number of segments in the training corpora before and after automatic splitting

Corpus	Automatic splitting	Number of segments
AR-EN	without	44,164
	with	49,318
JP-EN	without	64,145
	with	71,435

4.1. Phrase table augmentation

The trained translation model is represented in a "phrase table" where all the bi-phrases extracted with the *grow-diag-final-and* heuristic [13] are stored along with their translation model parameters. However, there may be some source-language words in the training corpus without a one-word entry in the phrase table. To avoid out-of-vocabulary treatment of these words in previously unseen contexts, we appended them to the list of phrases extracted by the *Moses* `phrase-extract` module. Specifically, we extracted one-word bi-phrases for these missing source vocabulary words by selecting their target candidates from *GIZA++* word alignments whose lexical translation probabilities were above a relative threshold. Table 4 shows the size of the phrase table before and after this process.

Table 4: Phrase table size before and after augmentation

Corpus	AR-EN	JP-EN
Source vocabulary size	18,571	12,669
Number of entries in the original phrase table	408,052	606,432
Number of source vocabulary words without a one-word entry in the original phrase table	8,035	6,302
Number of one-word bi-phrases added to the phrase table	21,439	23,396
Number of entries in the augmented phrase-table	429,491	629,828

4.2. Parameter tuning

We manually tuned the various system parameters to maximize the BLEU scores. Our general observation was that different test data favored different set of parameters. In order to select a robust set of parameters for our system, we performed experiments varying multiple parameters at the same time. Then we ordered the resulting BLEU scores, and rather than selecting the *argmax* of the distribution, we selected the *mode* in a "desirable interval". Here, the desirable interval can be defined as "top 50%", "top 25%", etc. That is, for each parameter, the value that appears the most in the "desirable interval" of the ordered scores is selected. Note that if the "desirable interval" is narrowed all the way down to the 1-best BLEU score, this operation becomes equivalent to selecting the *argmax*.

5. Run-time lexical approximation

A major obstacle for corpus-based machine translation systems when dealing with morphologically rich languages is the high out-of-vocabulary (OOV) rate due to the large number of forms a word can appear in, especially when the training corpus size is small. Unless processed specially, these words are output without being translated, hurting the translation quality. Last year's workshop underscored the importance of a system's capability to deal with input sentences containing phrases never seen before [3].

In an attempt to translate the OOV words in the source texts, we "approximate" them whenever possible with a best-guess replacement from the training set vocabulary as follows. Given the training set vocabulary **V** and a pre-defined word-feature function *f*(), for each OOV word w_{oov} in the source text a replacement word *w** is found by:

$$\mathbf{V}_{oov} \overset{def}{=} \{ w \in \mathbf{V} : f(w) = f(w_{oov}) \} \qquad (1)$$

$$w^* = \underset{w \in \mathbf{V}_{oov}}{\text{argmax}\, freq(\text{ argmin } dist(w_{oov}, w))} \qquad (2)$$

where *dist*() is the Levenshtein distance between two strings where substitution has twice the cost of insertion and deletion, and *freq*() is the frequency of a word in the training set.

In the first step, a set of candidate approximations $\mathbf{V}_{oov}$ is generated by identifying all the vocabulary words that have a common feature *f* with the OOV word in question. In the next step, the candidate with the least edit distance from the

OOV word is selected. In case of a tie, the more frequently occurring candidate is chosen.

When decoding an Arabic input sentence, we apply lexical approximation twice. In the first pass, the feature function returns the morphological root of the word according to the Buckwalter Arabic Morphological Analyzer (BAMA).

For words unrecognized by BAMA, we apply a second lexical approximation pass in which the feature function returns the "skeletonized" version of the word where all the vowels and diacritics are removed. Our skeletonization is similar to the procedure used inside BAMA before looking up a word in its lexicon, except that we also treat Arabic character "ALIF" and its variants as vowels.

For Japanese input, the lexical approximation feature function returns all the right-truncations of a word by removing a character from the right end iteratively. Taking into account the fact that Japanese is an agglutinative language with mostly suffixes, this feature function can be considered as a crude and overly-aggressive approximation to a Japanese morphological root estimator.

Tables 5 and 6 show the reduction in OOV words and the resulting performance improvement by using the above techniques. Because of its higher OOV rate, the AR-EN translation benefits more than JP-EN. On the other hand, a higher percentage of OOV words are replaced in the Japanese data set because of the greedy feature function, which can match two words even if they only have their first characters in common.

Table 5: OOV reduction in Arabic development set after run-time Lexical Approximation (LA)

	devset4		*devset5*	
	# of OOVs	BLEU	# of OOVs	BLEU
Original	661	**24.91**	795	**20.59**
After LA#1	185	**25.33**	221	**21.22**
After LA#2	149	**25.56**	172	**21.51**

Table 6: OOV reduction in Japanese development set after run-time Lexical Approximation (LA)

	devset4		*devset5*	
	# of OOVs	BLEU	# of OOVs	BLEU
Original	119	**23.68**	169	**20.44**
After LA	10	**23.84**	17	**20.69**

It should be noted that the selected replacement word is not guaranteed to be correct, either due to the looseness of the feature function or of the selection process, or because the word is indeed different from all of the words in the training data.

6. Results and discussion

Tables 7 and 8 respectively show the characteristics of the evaluation sets and the official BLEU scores of our submitted system on those sets. Under the clean transcript condition, both Japanese-to-English and Arabic-to-English systems perform reasonably well. The performance is lower as expected for the ASR output conditions. However, the drop in the Arabic-to-English task is larger than that in the Japanese-to-English task.

Table 7: OOV statistics of the evaluation sets

		Clean transcript	ASR output
Number of segments		489	
Avg. number of tokens per segment (EN)		7.7	
Number of source OOV words	AR	424 (14.1%)	374 (15.9%)
	JP	35 (0.9%)	18 (0.5%)

Table 8: Official BLEU scores of the submitted systems

Input conditions	Clean transcript	ASR output
AR-EN	**49.23**	**36.79**
JP-EN	**48.41**	**42.69**

6.1. Clean transcript vs. ASR output

Common to both language directions, the possible causes of the translation performance drop in the ASR output condition and their remedies can be listed as follows:

- Automatic speech recognition introduces erroneous words in the source. Recognition errors could in theory be reduced using techniques better than always selecting the 1-best recognition output, since the speech recognizer's "lattice accuracy" is significantly better than its 1-best accuracy, especially for Arabic [3].
- The ASR output lacks punctuation. Automatic punctuation prediction, whether it is done before, during, or after translation, will inevitably introduce punctuation errors. Better punctuation modeling could be researched.
- We performed parameter tuning only on the clean transcripts of the development sets, favoring the clean transcript condition. The system could be made more robust by tuning with the ASR outputs as well.
- Source-side punctuation insertion also favors clean transcript condition since the translation models as a result are trained on perfectly punctuated texts. They perform the best when the input is also perfectly punctuated. The system could be trained without punctuation on the source side, e.g., as done in [14].

6.2. AR-EN vs. JP-EN

In last year's workshop, the recognition accuracy of the Arabic data was lower than that of Japanese [3], which, if also true for this year's evaluation sets, could be the main source of the observed performance drop discrepancy between the two language tracks. Furthermore, the following factors additionally amplify the clean-to-ASR translation performance drop in the AR-EN task:

- Table 9 suggests that the performance gain of lexical approximation is sensitive to recognition errors. Without lexical approximation, the performance drop from clean to ASR output condition is less severe.

Table 9: BLEU scores for the Arabic evaluation data before and after run-time lexical approximation (LA)

	Clean transcript	ASR output	Clean-to-ASR degradation
Original source	**38.48**	**31.82**	17.3%
After LA	**49.23**	**36.79**	25.3%
Improvement	27.9%	15.6%	-

- Recognition errors also make the automatic punctuation insertion less reliable, as the latter is trained on clean transcripts.

Another reason for poorer automatic punctuation insertion performance in Arabic than in Japanese is the higher difficulty of the task due to the number of punctuation types that need to be differentiated, as explained in Section 3, leading to a bigger drop in performance.

6.3. *devsets4-5* vs. evaluation set

A comparison of Tables 5 and 9 reveals a dramatic variation in the improvement obtained with the lexical approximation technique on the evaluation and development sets. This variation can be attributed to the conspicuously different characteristics of the two data sets, which is already evident by the significant gap in segment lengths (*cf.* Tables 1 and 7) and absolute BLEU scores (*cf.* Tables 5 and 8). We performed further analysis to pinpoint which data set characteristic could be primarily responsible for the effectiveness of lexical approximation.

Table 10 shows that a significant number of the evaluation set segments (167 out of 489) have at least one reference which is a perfect match with a training segment. However, out of these 167 segments, only 19 have the source segment exactly the same as in the training set. Therefore the remaining 148 segments represent a potential to obtain a perfect match and a big improvement in BLEU score if the "proper" source modification is applied. For example, the source difference may be due to a Arabic spelling variation or a morphological variation that map to the same word form in English. When such variations result in an out-of-vocabulary source word, it is possible for the lexical approximation method to find a replacement that results in a perfect match.

Table 10: Exact matches between the training set and various test sets

Number of segments	Devset4	Devset5	Evaluation set
Exact match of at least one reference with a segment in the training set	12	4	167
Exact match of the source with a segment in the training set	1	0	19
Total	489	500	489

7. Conclusion and future work

We have presented our Arabic-to-English and Japanese-to-English statistical machine translation systems based on publicly-available software. We described our modifications to automatic punctuation insertion and translation model generation. We introduced run-time lexical approximation as an effective method to cope with out-of-vocabulary words in the input. Three word-level feature functions were proposed. The Arabic-to-English translation benefited the most from lexical approximation due to its high OOV rate. Overall, our system obtained the highest BLEU scores among participants for both Japanese-to-English and Arabic-to-English under the clean transcript condition. The system's performance degradation under the ASR output condition was investigated.

A priority in the future will be to make the system more robust to ASR output conditions, namely, possible recognition errors and lack of punctuation. Making use of the N-best/lattice ASR outputs, including ASR outputs when tuning the system, and better punctuation handling and modeling are some of the future work towards this goal.

8. References

[1] Akiba, Y., Federico, M., Kando, N., Nakaiwa, H., Paul, M., Tsuji, J., "Overview of the IWSLT04 Evaluation Campaign", in *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2004, pp. 1-12.

[2] Eck, M. and Hori., C., "Overview of the IWSLT 2005 Evaluation Campaign", in *Proc. of the International Workshop on Spoken Language Translation*, Pittsburgh, USA, 2005, pp. 11-32.

[3] Paul, M., "Overview of the IWSLT 2006 Evaluation Campaign", in *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2006, pp. 1-15.

[4] IWSLT 2007, http://iwslt07.itc.it/, 2007.

[5] Kikui, G., Sumita, E., Takezawa, T., and Yamamoto, S., "Creating Corpora for Speech-to-Speech Translation", in *Proc. of the EUROSPEECH03*, Geneve, Switzerland, 2003, pp. 381-384.

[6] Och, F. Z. and Ney, H., "A Systematic Comparison of Various Statistical Alignment Models", *Computational Linguistics*, vol. 29, no. 1, 2003, pp. 19-51.

[7] Brown, P.F., Della Pietra, S.A., Della Pietra, V.J. and Mercer, R.L., "The Mathematics of Statistical Machine Translation: Parameter Estimation", *Computational Linguistics*, vol. 19, no. 2, 1993, pp. 263-311.

[8] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. and Herbst, E., "Moses: Open Source Toolkit for Statistical Machine Translation", *The 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, demonstration session, Prague, Czech Republic, June 2007.

[9] Stolcke, A., "SRILM – an extensible language modeling toolkit", in *Proc. International Conference on Spoken Language Processing*, vol. 2, Denver, USA, 2002, pp. 901-904.

[10] Buckwalter Arabic Morphological Analyzer Version 1.0. Linguistic Data Consortium Catalog: LDC2002L49.

[11] IWSLT 2006 Evaluation Campaign, http://www.slc.atr.jp/IWSLT2006/downloads/case+punc_tool_using_SRILM.instructions.txt, 2007.

[12] WMT 2007, The Second ACL Workshop on Machine Translation, http://www.statmt.org/wmt07/baseline.html, 2007.

[13] Koehn, P., Axelrod, A., Mayne, A.B., Callison-Burch, C., Osborne, M. and Talbot, D., Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation, in *Proc. of the International Workshop on Spoken Language Translation*, Pittsburgh, PA, 2005.

[14] Mauser, A., Zens, R., Matusov, E., Hasan, S., and Ney, H., "The RWTH Statistical Machine Translation System for IWSLT 2006 Evaluation", in *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2006, pp. 103-110.