The TÜBİTAK-UEKAE Statistical Machine Translation System for IWSLT 2010

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 report on our participation in the IWSLT 2010 evaluation campaign. Similar to previous years, our submitted systems are based on the Moses statistical machine translation toolkit. This year, we also experimented with hierarchical phrase-based models. In addition, we utilized automatic minimum error-rate training instead of manually-guided tuning. We focused more on the BTEC Turkish-English task and explored various experimentations with unsupervised segmentation to measure their effects on the translation performance. We present the results of several contrastive experiments, including those that failed to improve the translation performance.

1. Introduction

Our submissions in previous IWSLT evaluation campaigns mainly focused on reducing the number of out-of-vocabulary words during decoding using techniques such as lexical approximation and phrase-table augmentation [1]. Our focus this year shifts to improving the morphological segmentation, particularly in the BTEC Turkish-English task, so as to increase the system's translation performance.

We start with describing the common experiments and components of the developed systems for all the tasks in Section 2. Then we present the several morphological preprocessing schemes for Turkish-English translation in Section 3, emphasizing unsupervised methods of sub-word segmentation. In Section 4, the task-specific preprocessing for each of BTEC Arabic-English, DIALOG Chinese-English and English-Chinese, and TALK English-French tasks are explained. The experimental results and discussion of each proposed method is presented at the place of their description.

2. Common system development

We used the open-source statistical machine translation toolkit Moses [2] for training the translation models and for decoding in our primary submissions. The N-gram target language models were trained using the SRI language modeling toolkit [3]. We used 4-gram language models since they gave the best performance in our 2009 systems [4]. All the system training and decoding was performed on lowercased and punctuation-tokenized data. After decoding, we restored the case information using the Moses recaser scripts. All the BLEU scores reported in this paper are based on cased, punctuated system outputs and references.

Table 1: *Comparison of %BLEU scores for the developed Moses-based and Joshua-based systems for the BTEC and DIALOG tasks (CRR input condition for CE and EC)*

System	Devset	%BLEU	Testset	%BLEU
TE_Joshua	dev1	58.16	dev2	53.60
TE_Moses		64.62		**59.46**
AE_Joshua	dev6	46.54	dev7	46.32
AE_Moses		48.00		**47.38**
CE_Joshua	dev8+9	33.51	DIALOG	35.04
CE_Moses		34.37		**42.28**
EC_Joshua	dev3	34.63	DIALOG	29.17
EC_Moses		36.47		**29.54**

In addition, this year we used the minimum-error rate training method [5] for tuning the parameters of the log-linear models using the flexible open-source tool Z-MERT [6].

This year we also tried hierarchical phrase-based models [7] using the open-source training and decoding toolkit Joshua [8], which has been used successfully in recent translation systems [9]. The results are shown in Table 1. Since our experience with the Joshua toolkit has been limited, it might have contributed to sometimes significantly lower scores in these experiments.

3. Morphological preprocessing for Turkish-English

3.1. Morphological problems in Turkish-English MT

Turkish is an agglutinative language where words can carry several morphemes in the form of suffixes. For example, Fig. 1 shows the morphological decomposition of the Turkish word *yapamayacaksan* and the morpheme-based alignment to its English translation. Even though there are a total of about 150 distinct lexical suffixes in Turkish, the number of possible word forms are practically unlimited, causing

yap	+a	+ma	+yacak	+sa	+n
do	be able to	not	will	if	you

'if you will not be able to do'

Figure 1: *Morphemes and their glosses for the Turkish word* yapamayacaksan

data sparseness at the word-level. As a result, statistical machine translation involving Turkish requires special attention to morphology.

In our 2009 systems, we investigated three approaches to dealing with the morphology of Turkish, namely linguistic morphological segmentation, unsupervised segmentation, and lexical approximation, with the former performing the best. This year we again experimented with the former two methods, investigating ways to improve the unsupervised performance.

3.2. Using a morphological analyzer (primary submission)

We preprocessed the Turkish texts both in training and decoding using linguistic morphological analysis to separate the words into their roots and morphemes. We used the finite-state morphological analyzer by Kemal Oflazer [10]. The morphological parses were disambiguated using the statistical disambiguator of Sak et al. [11].

In our submission for IWSLT 2009, through analysis of the morphologically-segmented and disambiguated training corpus we designed some post-processing rules that selectively merge or delete certain morphemes with the aim of matching the morphologies of the two languages. This year we again used these rules. To summarize, we removed those Turkish morphemes that do not have an overt form in English, e.g., the accusative marker and the imperative verb form. In addition, since morphological analysis is only applied on the Turkish side, there is some over-segmentation relative to the English side, e.g., the noun plural suffix, the infinitive type-3 verb form (as in *to sell* $\rightarrow$ *sale* and *to fly* $\rightarrow$ *flight*), and the "as if" marker (usually corresponding to the "+*ly*" suffix that generates adverbs). Such morphemes were attached to their roots on the Turkish side. These decisions are applied on the morpheme vocabulary and are static for all occurrences of those morphemes. For more explanation and examples, please refer to our 2009 system description paper [4].

The described methodology of utilizing a morphological analyzer + disambiguator + rules was also adopted for Turkish by Bisazza and Federico [12] and Oflazer and Durgar El-Kahlout [13].

Faced with the decision on a development set for Turkish-English, since there were only two development sets provided, we could not test on an independent set to see which of the two devsets result in better generalization of the tuned parameters. So, when decoding the testsets, we decided to use the arithmetic average of the two sets of parameters tuned on dev1 and those tuned on dev2 (without knowing beforehand whether this method would be better than tuning with either of the devsets). Table 2 shows that this method indeed resulted in a better performance than choosing one of the devsets (dev1) for parameter tuning.

Table 2: *Comparison of %BLEU scores of tuning with dev1 versus averaging parameters tuned with dev1 and those tuned with dev2*

Tuning method	dev1	dev2	iwslt09	iwslt10
dev1	64.62	59.46	56.40	53.32
(dev1+dev2)/2	62.92	60.37	**57.63**	**54.05**

Comparison of this method (named here "linguistic+manual") to the word-based baseline (along with two unsupervised methods described in the next section) is presented later in Section 3.4, where it is seen to yield the best performance. This result is consistent with the findings of last year's experiments [4].

3.3. Using unsupervised morphological segmentation

Developing a morphological analyzer requires linguistic expertise and extensive manual effort. Even then, constant updating of the vocabulary is necessary as new words emerge, or when faced with domain-specific terminology. Moreover, the optimal segmentation of a morphologically-complex word might differ depending on the target language's morphology (e.g., a distant vs. a close language pair) or the size of the training corpus (the common wisdom in SMT community is that less segmentation is better when more training data are available). Furthermore, the IWSLT evaluation campaigns encourage using only the provided resources. With these motivations, we also extensively investigated using an unsupervised morphological segmentation tool, called Morfessor [14], which is publicly available. Morfessor utilizes the minimum description length (MDL) principle to search for the optimal sub-word segmentation of a given corpus. The segmentations in this model are static in that all the occurrences of a word are assumed to be segmented in the same manner regardless of the context.

We used the supplied BTEC training corpus as input to Morfessor version 1.0 (also called the "baseline" model in [14]), and obtained a (deterministic) segmentation for each word in the vocabulary. We segmented the Turkish side of the training corpus by replacing each word with its segmentation, and the resulting corpus was paired with the word-based English corpus to train the translation model. In decoding, the same segmentation model was also applied to the Turkish input.

The performance of Morfessor-baseline on the development sets and the 2009 and 2010 test sets is shown in Section 3.4.

Table 3: *Comparison of %BLEU scores with and without postprocessing allomorphs in Morfessor output*

Representation	dev1	dev2	iwslt09	iwslt10
Surface forms	59.41	54.42	**52.15**	**49.83**
Allomorphs	59.53	55.28	51.57	48.93

3.3.1. Utilizing allomorphy

Morfessor does not use any linguistic knowledge in its model. However, by incorporating minimal linguistic knowledge in the form of allomorphy (the same lexical morpheme appearing in different surface forms depending on the stem it is attached to), one might expect to improve the translation performance. To test this hypothesis, we used a setup as follows. The segmentation model is trained and the corpus segmented as before using Morfessor. Then, all the allomorphic letters in all the suffixes are mapped to their base letter, (e.g., [ı, i, u, ü] are all mapped to H etc.), hoping that equivalences between variants of the same lexical morphemes are in this way captured. This postprocessing is not applied to the stems. The resulting corpus is fed to the SMT training (or decoding) phase.

Table 3 shows that, even though small improvements in development sets (and even in last year's experiments) were observed, we did not obtain the expected improvements in this year's tests. It is possible that imposing allomorphy externally after the segmentation is learned has a negative effect on the performance. A better method of handling allomorphy could be to use this linguistic knowledge during segmentation learning inside Morfessor (though the new segmentation method would no longer be truly unsupervised).

3.3.2. Including the test set in segmentation training

The segmentation model trained as such can only segment those words seen in the training corpus. This results in all of the out-of-vocabulary words in the test sentences to be left unsegmented by the model. This might be desirable if the OOV word is a proper noun, for example. However, especially in morphologically-rich languages, a significant portion of the OOV words are due to previously unseen root+morpheme combinations, even though these roots and morphemes might be seen in other contexts in the training corpus.

Therefore we tried segmenting the test set using Viterbi decoding as described in [15], which searches for the maximum-probability segmentation of the test set given the model. However, one drawback of this method is that it forces the resulting segmented test set to consist entirely of the roots and morphemes in the segmentation model (or individual letters if no such segmentation is impossible). Therefore all the OOV words are forcibly segmented, leading to incorrect segmentations for some of them, e.g., proper nouns. This method (named here "train+viterbi") is compared in Table 4 against segmenting only the in-vocabulary words using the mapping learned in the segmentation model (named here "train").

Table 4: *Comparison of %BLEU scores with different segmentation methods for the test set*

Method	dev1	dev2	iwslt09	iwslt10
train	**59.53**	**55.28**	51.57	48.93
train+viterbi	58.14	52.92	51.73	48.35
train+test	59.23	53.23	**52.00**	**50.53**

Last year to address this problem we experimented with including the test corpus when training the segmentation model. With this method, all the words in the testset are now proposed a segmentation (or non-segmentation) according to the learned model. The performance is listed in Table 4 as "train+test".

In last year's experiments, the performance of "train+test" were inferior to the "train" method on the development sets, so it was not used in our primary submission. However, Table 4 shows that it actually performs much better in iwslt09 and iwslt10 test sets. The main degrading factor in this segmentation method is the unnecessary segmentation of especially the rare OOV words (such as proper nouns) in the test set, which tend to be segmented into smaller, more frequent morphs. We suspect that differences in the distribution of such OOV words between the development and test sets might be the reason for this performance discrepancy. Also note that re-training was performed after the test sets are released and all four sets in Table 4 were appended to the training set.

Implementation-wise, the "train+test" method requires knowing the testset beforehand. Although the experiment described here took less than one minute on a standard computer, re-training the segmentation model in an online setup on a sentence-by-sentence basis might not be practical. But for applications where the input sentences can be processed in batch, this once-per-batch training step might be acceptable.

3.3.3. Parallel search and Gibbs sampling

In this section, we propose two new search methods as an alternative to the search algorithm that Morfessor uses in segmentation training. Morfessor's original search algorithm [15] processes all words in the vocabulary one-by-one (in random order), computing for each word the posterior probability of the existing model after each possible binary segmentation (splitting) of the word. The highest-scoring split (or non-split) is accepted. The process is repeated iteratively until convergence. This search algorithm is a greedy algorithm where the costs of the next search points are affected by the decision in the current step. This leads to a sequential search and does not lend itself to parallelization.

In the first search alternative [16], named here "batch-

Table 5: *Comparison of %BLEU scores with different segmentation search algorithms*

Search algorithm	dev1	dev2	iwslt09	iwslt10
original	59.41	54.42	**52.15**	**49.83**
batch-update	59.22	53.61	50.68	48.55
stochastic	59.09	54.55	51.90	48.60

update", the segmentation decisions for individual words are stored but are not applied until the end of an iteration. In this way, all cost calculations can be performed independently and in parallel. Since the model is not updated at every decision, the search path generally differs from that in the sequential search and hence results in a different final segmentation.

The second alternative search strategy, named here "stochastic", is closely related to Gibbs sampling. Instead of the greedy model updates at each processed word, the segmentation decision for a word is sampled from the distribution proportional to the posterior probability of the model given the existing state of segmentation for the rest of the words. Note that these two proposed methods are not mutually exclusive and they can co-exist in a segmentation scheme.

Comparison of the translation performance of the two methods against the original Morfessor search algorithm is presented in Table 5. Neither method improves upon the original baseline in either test set, though "stochastic" comes closer to the performance of the original search.

3.3.4. Segmentation training with monolingual out-of-domain corpus

In this section, we explore whether using a large monolingual corpus can reduce data sparsity of Turkish word forms and hence improve the segmentation. We experiment with using a large Turkish monolingual corpus to see whether a better segmentation can be learned. The additional corpus, which consists of about 40 Mwords with a vocabulary size of about 500 K, is gathered from Turkish news sites on the web, so it is out of domain for the BTEC task.

In the first experiment (named here as "+mono"), we simply merge the BTEC corpus with the additional monolingual corpus and train Morfessor. In the second experiment (named here as "+mono(flat)"), we set the frequencies of all the words in the vocabulary to 1. This latter method has been reported to result in more satisfactory segmentations in some applications [14], especially with large corpora, because in Morfessor's model it is more costly to split frequently-occurring words than rarely-occurring words. As corpus size increases, even the morphologically complex words start occurring frequently, resulting in not being segmented. As a result, training Morfessor on "types" rather than on "tokens" is sometimes found to match linguistic segmentation more closely. Since our additional monolingual corpus is quite large, we also experimented with this flat-vocabulary method. But we first cut-off the singletons in the out-of-domain corpus before merging the two vocabularies, mainly for text noise reduction.

Table 6: *%BLEU scores with and without added monolingual out-of-domain corpus for segmentation training*

Corpus	tuning	dev1	dev2	iwslt09	iwslt10
btec	dev1	59.41	54.42	**52.15**	**49.83**
+mono	dev1	55.88	50.49	49.17	46.09
+mono(flat)	dev1	58.98	53.53	50.69	48.87
+mono	dev2	53.60	53.46	50.31	47.01
+mono(flat)	dev2	56.89	56.54	51.08	49.66

The results are shown in Table 6. Using an out-of-domain monolingual corpus did not help the translation performance in our experiments, though training on types is found to be more effective than training on tokens in this case.

For this experiment only, we also compared tuning on devset1 versus devset2. Table 6 shows that tuning on devset2 consistently gave the better test set performance.

3.3.5. Morfessor Categories-MAP

Up to here, the unsupervised segmentation experiments are conducted using Morfessor-baseline, which employs a fairly simple segmentation model where the induced morphs are assumed to be independent of their context. A more advanced model called Morfessor Categories-MAP [14] probabilistically assigns each induced morph to one of prefix, stem, or suffix classes. In an observed corpus of words segmented into morphs, the transitions between classes and the emissions of morphs from a given class are modeled in a hidden Markov model (HMM) framework.

The performance of this segmentation model, named here as "Morfessor-catmap", is compared in Section 3.4. It exceeds the performance of Morfessor-baseline, but still falls short of supervised segmentation.

3.4. Comparison of methodologies

To summarize the results in this section, Table 7 compares the performances from the described supervised segmentation scheme and two unsupervised segmenters (Sections 3.2-3.3) against the word-based baseline (i.e., no segmentation). It is seen that supervised segmentation gives by far the best translation performance.

4. Other task-specific preprocessing

4.1. BTEC Arabic-English

In our 2008 and 2009 systems, we applied an orthographical normalization (named here as "scheme-A") to all training and test corpora, which was originally motivated by the orthographic variations in the automatic speech recognition

Table 7: *%BLEU scores of the developed Turkish-English systems each tuned on devset1*

Segmentation	dev1	dev2	iwslt09	iwslt10
Word-based	56.65	51.40	49.48	47.49
Morfessor-baseline	59.41	54.42	52.15	49.83
Morfessor-catmap	62.69	54.78	53.03	50.91
Linguistic+manual	64.62	59.46	**56.40**	**53.32**

Table 8: *Arabic-English %BLEU scores for different orthographical normalization schemes tuned on dev6 and tested on dev7*

Normalization	Dev	Test
scheme-A	46.54	**46.32**
scheme-B	38.28	43.80
none	37.89	43.00

(ASR) outputs. This year we also tried a simpler normalization used in [17], in which only the alef and ya variants are normalized (named here as "scheme-B"). We compared both of these orhtographic normalization schemes against using the corpora as is (named here as "none") in Table 8 (the experiments in this section were conducted with the Joshua toolkit). Scheme-A clearly outperformed the other methods, so it was continued to be used in further experiments and in our primary submission.

Among the provided development corpora, devset6 was used for tuning the parameters and devset7 for internal testing. The remaining devsets (1-3) were added to the training set. In our Arabic-English systems for the previous IWSLT's, all 16 references per source sentence was added to the training set (with the source sentence replicated 16 times). This year we also experimented with adding only one reference per source sentence. Table 9 compares the resulting translation performances. The latter method yielded higher BLEU scores, hence one reference per source sentence was added to the training set in our primary submission.

4.2. DIALOG Chinese-English and English-Chinese

For each of the translation directions in the DIALOG task, two separate systems were developed: one for translating the ASR outputs and one for translating the correct recognition results (CRR). Each of these systems was tuned on the respective condition of the development set. The development sets selected for parameter tuning in our primary submissions are denoted in Table 1 as "devset". Since both conditions lacked punctuation in the test input sentences, we trained an N-gram punctuation model from the training set and inserted punctuation marks using the `hidden-ngram` tool from the SRILM package [3].

Table 9: *Arabic-English %BLEU scores for different training corpora tuned on dev6 and tested on dev7*

Training corpus	Dev	Test
train	45.85	45.45
+devs1-3(16 refs)	44.99	46.12
+devs1-3(1 ref)	46.54	**46.32**

Table 10: *TALK task %BLEU scores for two methods of recasing*

Recasing method	Dev	Test
Normal recase	19.94	23.01
Recase w/ dummy	19.98	**23.50**

4.3. TALK English-French

A quick inspection of the TED training corpus revealed some unbalanced sentence-pairs in terms of the number of words. Such sentences generally had one side replicated while the other side contained the translation of a different part of the other side at each replication. Using an automatic script, 399 such unsymmetrically repeated lines were detected and removed from the training corpus.

Further inspection of the parallel corpus suggested that due to the very similar ordering between the two languages and the general faithfulness of the translations to the originals, the sentence fragments within comma boundaries were likely to be parallel as well. Based on this observation, during sentence splitting for the TED training set, we also included the comma marks as possible split points (in addition to sentence markers). As a result, the number of parallel sentences in the TED training corpus increased from about 83 K to about 103 K.

We found handling punctuation modeling during the training and decoding phases for the TED corpus to be tricky. Since the ASR outputs contained on the average about 5 times longer sequences of words per segment than the training text, we synthetically merged the training corpus in blocks of 5 segments at a time and then trained the punctuation N-gram model.

Restoration of case information was also tricky, because even the CRR condition test set segments did not always coincide with the sentence boundaries and the majority of the segments started mid-sentence. Even disabling the automatic sentence-beginning uppercasing still yielded several false uppercased segment beginnings since the recasing model had a bias for uppercasing words at segment beginnings compared to other words. To remove this bias and to treat the segment starts as any other context, we artificially inserted dummy out-of-vocabulary words at the beginning of each segment before running the recaser script. Table 10 shows that this method (named here as "recase w/ dummy") improved the cased BLEU scores over "normal recase".

We divided the provided development set roughly into

half, keeping the individual documents intact, for the purposes of parameter tuning on one half and internal testing on the other. So, the parameters of our primary submission was tuned on one-half of the development set.

5. Conclusions

We presented our primary submission systems for five translation tasks in IWSLT 2010. We also reported the results of extensive experimentation, some of which improved the system performance while some failed to do so. We focused more on the Turkish-English task, and particularly on the problem of unsupervised segmentation. While supervised segmentation via a morphological analyzer and manual postprocessing yielded the best translation performance, the HMM-based morpheme model showed improved translation performance compared to the baseline unsupervised segmentation.

6. Acknowledgements

We thank Kemal Oflazer for providing the Turkish morphological analyzer.

7. References

[1] C. Mermer, H. Kaya, and M. U. Doğan, "The TUBITAK-UEKAE statistical machine translation system for IWSLT 2007," in *Proc. of the International Workshop on Spoken Language Translation*, Trento, Italy, 2007.

[2] 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 *Proc. of the 45th Annual Meeting of the Association for Computational Linguistics, Companion Volume Proc. of the Demo and Poster Sessions*, Prague, Czech Republic, June 2007, pp. 177–180.

[3] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, vol. 3, 2002.

[4] C. Mermer, H. Kaya, and M. U. Doğan, "The TUBITAK-UEKAE statistical machine translation system for IWSLT 2009," in *Proc. of the International Workshop on Spoken Language Translation*, Tokyo, Japan, 2009.

[5] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, July 2003, pp. 160–167.

[6] O. F. Zaidan, "Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems ," *The Prague Bulletin of Mathematical Linguistics*, vol. 91, no. 1, pp. 79–88, 2009.

[7] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[8] Z. Li, C. Callison-Burch, C. Dyer, S. Khudanpur, L. Schwartz, W. Thornton, J. Weese, and O. Zaidan, "Joshua: An open source toolkit for parsing-based machine translation," in *Proc. of the Fourth Workshop on Statistical Machine Translation*, Athens, Greece, March 2009, pp. 135–139.

[9] L. Schwartz, "Reproducible results in parsing-based machine translation: The JHU shared task submission," in *Proc. of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, Uppsala, Sweden, July 2010, pp. 177–182.

[10] K. Oflazer, "Two-level description of Turkish morphology," *Literary and Linguistic Computing*, vol. 9, no. 2, 1994.

[11] H. Sak, T. Güngör, and M. Saraçlar, "Morphological disambiguation of Turkish text with perceptron algorithm," in *Proc. CICLing*, 2007, pp. 107–118.

[12] A. Bisazza and M. Federico, "Morphological Pre-Processing for Turkish to English Statistical Machine Translation," in *Proc. of the International Workshop on Spoken Language Translation*, Tokyo, Japan, 2009, pp. 129–135.

[13] K. Oflazer and İ. Durgar El-Kahlout, "Exploring different representational units in English-to-Turkish statistical machine translation," in *Proc. of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, June 2007, pp. 25–32.

[14] M. Creutz and K. Lagus, "Unsupervised models for morpheme segmentation and morphology learning," *ACM Transactions on Speech and Language Processing*, vol. 4, no. 1, pp. 1–34, 2007.

[15] ——, "Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0," *Computer and information science, Report A*, vol. 81, 2005.

[16] C. Mermer and A. A. Akın, "Unsupervised search for the optimal segmentation for statistical machine translation," in *Proc. of the ACL 2010 Student Research Workshop*, Uppsala, Sweden, July 2010, pp. 31–36.

[17] N. Habash, "REMOOV: A tool for online handling of out-of-vocabulary words in machine translation," in *Proc. of the 2nd International Conference on Arabic Language Resources and Tools (MEDAR)*, Cairo, Egypt, 2009.