# The TÜBİTAK-UEKAE Statistical Machine Translation System for IWSLT 2009

*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 our Arabic-to-English and Turkish-to-English machine translation systems that participated in the IWSLT 2009 evaluation campaign. Both systems are based on the Moses statistical machine translation toolkit, with added components to address the rich morphology of the source languages. Three different morphological approaches are investigated for Turkish. Our primary submission uses linguistic morphological analysis and statistical disambiguation to generate morpheme-based translation models, which is the approach with the better translation performance. One of the contrastive submissions utilizes unsupervised subword segmentation to generate non-linguistic subword-based translation models, while another contrastive system uses word-based models but makes use of lexical approximation to cope with out-of-vocabulary words, similar to the approach in our Arabic-to-English submission.

## 1. Introduction

We report on our participation in the Arabic-to-English and Turkish-to-English BTEC translation tasks of IWSLT 2009. We developed systems that make use of morphology to improve upon the word-based baselines. To alleviate the out-of-vocabulary (OOV) word problem, we experimented with the lexical approximation technique in both tasks. We also developed two Turkish-to-English systems that morphologically analyze the Turkish words, both in a supervised and unsupervised manner. We were able to achieve significant improvements over the word-based baseline when translating from both languages.

In the following, we describe our methods of handling the morphology of Turkish and Arabic in sections 2 and 3, respectively. The general system architecture common to both tasks is explained in section 4, followed by the results and discussion in section 5.

## 2. Coping with Turkish morphology

Turkish is an agglutinative language where words can carry several morphemes in the form of suffixes. For example, (1) shows the morphological decomposition of the Turkish word '*yapamayacaksan*' and the morpheme-based alignment to its English translation. When aligning with a parallel English text, Turkish morphology creates problems in traditional word alignment approaches since the morphologies are asymmetric. Furthermore, vast number of word forms in Turkish cause data sparseness at the word-level. Even though there are a total of about 150 distinct suffixes in Turkish, the particular morpheme sequence in a given word may be unseen in the training corpus. As a result, statistical machine translation involving Turkish requires special attention to Turkish morphology.

| | | | | | | |
|---|---|---|---|---|---|---|
| | yap | +a | +ma | +yacak | +sa | +n |
| (1) | *do* | *be able to* | *not* | *will* | *if* | *you* |
| | *'if you will not be able to do'* | | | | | |

In the development of our Turkish-English machine translation system for IWSLT 2009, we investigated three approaches to dealing with the morphology of Turkish, described in the following subsections.

### 2.1. Using a morphological analyzer (primary submission)

We applied linguistic morphological analysis to separate the words into roots and morphemes in the Turkish texts, before both the training and the decoding steps. We used the finite-state morphological analyzer by Kemal Oflazer [1]. The morphological parses were disambiguated using the statistical disambiguator of Sak *et al.* [2].

Some of the morphological features produced by the morphological analyzer do not have a counterpart in English. For example, in (2) the accusative marker in Turkish is not aligned to any of the words in the English translation. This suggests that removing some Turkish morphemes could help automatic alignment.

| | | | | | | |
|---|---|---|---|---|---|---|
| | bu | +nu | anla | +m(a) | +ıyor | +um |
| (2) | *this* | *??* | *understand* | *not* | *do* | *I* |
| | *'I do not understand this'* | | | | | |

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 plural noun suffix '*+s*' is not separated in the English corpus. Therefore keeping some Turkish morphemes attached to the root could be favorable.

Because of these reasons, we post-processed the morphological analyzer output to selectively merge or delete some morphemes. For example, the accusative and the imperative markers were deleted from the Turkish corpus. On the other hand, the type-3 infinitive and the "as-if" markers were attached to their roots. Examples for these morphemes are shown in Figures 1 and 2, respectively. The decisions are defined on the morpheme vocabulary and are static for all occurrences of those morphemes.

| | | | |
|---|---|---|---|
| o | +nu | git | +NULL |
| *it* | *ACC.* | *go* | *IMP. (sing.)* |
| bu adres | +i | dön | +NULL |
| *this address* | *ACC.* | *turn* | *IMP. (sing.)* |
| ekmek | +i | gir | +in |
| *(the) bread* | *ACC.* | *input* | *IMP. (pl.)* |
| bu düğme | +yi | çağır | +ın |
| *this button* | *ACC.* | *call* | *IMP. (pl.)* |
| (a) | | (b) | |

*Figure 1*: Examples of deleted Turkish morphemes. (a) The accusative marker. (b) The imperative marker, which does not have an overt form for 2nd person singular.

| | | | |
|---|---|---|---|
| yavaş | +ça | dal | +ış |
| *slow* | *AS-IF* | *to dive* | *INF-3* |
| *'slowly'* | | *'diving'* | |
| dikkatli | +ce | uç | +uş |
| *careful* | *AS-IF* | *to fly* | *INF-3* |
| *'carefully'* | | *'flight'* | |
| hızlı | +ca | sat | +ış |
| *quick* | *AS-IF* | *to sell* | *INF-3* |
| *'quickly'* | | *'sale'* | |
| sıkı | +ca | bin | +iş |
| *tight* | *AS-IF* | *to board* | *INF-3* |
| *'tightly'* | | *'boarding'* | |
| (a) | | (b) | |

*Figure 2*: Examples of Turkish morphemes attached back to the root. (a) The as-if marker. (b) The type-3 infinitive marker.

The decision to leave a morpheme as a separate unit, to merge with the previous morpheme, or to delete was made based on bilingual human judgments so as to match the English units (i.e., words) as good as possible. This approach is similar to the method reported in [3], although in the opposite translation direction.

### 2.2. Using unsupervised morphological segmentation (contrastive-1)

Development of a morphological analyzer requires lots of manual work and linguistic expertise. Thus a morphological analyzer for a language may not always be readily available. Furthermore, this year's IWSLT evaluation campaign encourages using only the provided resources so that the evaluation is one of the methods of machine translation rather than the resources. Therefore, we also investigated using an *unsupervised* morphological analyzer, called Morfessor [4], which is publicly available. Morfessor uses the minimum description length (MDL) principle to find an optimal subword segmentation of a given corpus in the form of a root-and-morpheme vocabulary. 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 [4]), and the algorithm converged to a model as output that defines a segmentation or non-segmentation for each word in the vocabulary. Figure 3 shows an example segmentation model output by Morfessor. Note that the fairly frequent word 'anladım' is left unsegmented while the other less frequent variants are segmented in terms of a smaller "codebook" of morphs induced from the entire corpus.

| Word | Count | Morfessor's segmentation |
|---|---|---|
| anladı | 1 | anladı<br>*understood* |
| anladım | 13 | anladım<br>*I understood* |
| anladın | 3 | anladı +n<br>*understood you (sing.)* |
| anladınız | 1 | anladı +nız<br>*understood you (pl.)* |
| anladıysam | 1 | anladı +ysa +m<br>*understood if I* |

*Figure 3*: Sample segmentation found by Morfessor.

After thus training a segmentation model, we segmented the Turkish side of the training corpus by replacing each word with its segmentation according to this model, 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 test input.

#### *2.2.1. 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 (154 words in dev1 and 181 words in dev2) to be left unsegmented by the model. However, the roots and/or some of the morphemes of the OOV words in the test sets may be previously seen in the training corpus. To be able to take advantage of this correlation, we experimented with including the test corpus (in this experiment, both dev1 and dev2) when training the segmentation model. As a result, *all* the input words were now proposed a segmentation or non-segmentation according to the learned model. However, the translation performance was slightly degraded as shown in Table 1, so we decided not to include the test sentences in the segmentation model training. The reason could be the unnecessary segmentation of the OOV words in the test set (Figure 4), which tend to be segmented into smaller, more frequent morphs since the unsegmented word frequency is very low (i.e., not seen in the original training set).

*Table 1*: Comparison of % BLEU scores with and without including the test set in Morfessor training

| Datasets used in segmentation training | Dev1 | Dev2 |
|---|---|---|
| Train only | **60.02** | **56.48** |
| Train + dev/test | 58.23 | 53.78 |

| | | |
|---|---|---|
| tarak<br>*'comb'* | → | ta + rak |
| ilkbahar<br>*'spring'* | → | ilk + bahar |
| kahvehane<br>*'coffehouse'* | → | kahve + ha + ne |
| saks<br>*'Saks'* | → | sak + s |

*Figure 4*: Segmentations found by Morfessor for some OOV words in the test input.

*2.2.2. Utilizing allomorphs*

The unsupervised segmentation scheme described up to here does not use any external data resource or linguistic knowledge. However, by incorporating minimal linguistic knowledge and very small additional effort, it is possible to improve the translation performance. In particular, due to vowel and consonant harmony rules in Turkish, morphemes can have many surface forms (called allomorphs) depending on the orthography of the appended stem. Figure 5 shows some of the possible surface forms of the Turkish past tense suffix. However, the algorithm utilized by Morfessor is by default unaware of allomorphs. By manual input of suffix equivalences during learning, Morfessor can utilize this information when counting and estimating morpheme boundaries and distributions. This results in a more informed and better segmentation, evident from the improved translation performance in Table 2.

| | | |
|---|---|---|
| kal<br>*stay* | **+dı**<br>*PAST* | *'stayed'* |
| git<br>*go* | **+ti**<br>*PAST* | *'went'* |
| gör<br>*see* | **+dü**<br>*PAST* | *'saw'* |
| koş<br>*run* | **+tu**<br>*PAST* | *'ran'* |

*Figure 5*: Allomorphs of the past tense morpheme resulting from phonetic harmony rules in Turkish.

*Table 2*: Effect of leveraging allomorphs during segmentation training on the final translation % BLEU scores

| | Dev1 | Dev2 | *Test 2009* |
|---|---|---|---|
| Using surface forms | 60.02 | 56.48 | *53.76* |
| Using allomorphs | **60.22** | **57.38** | ***53.95*** |

**2.3. Using lexical approximation (contrastive-2)**

As an alternative to morphological segmentation, we investigated the usefulness of the lexical approximation approach we had previously used in IWSLT 2007 [5] and 2008 [6], which is also used in our Arabic-to-English submission this year. In this approach, the corpus and the translation models remain word-based; however, a morphological analyzer may be utilized internally to compute a similarity feature between words based on their shared roots and morphemes.

In lexical approximation, a small subset of replacement candidates $\mathbf{V}_{oov}$ is extracted from the source vocabulary $\mathbf{V}$ by identifying words sharing a common feature $f$ with the OOV word in question $w_{oov}$ (Eq. 1).

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

The final choice of the replacement word $w^*$ among multiple candidates is made via shortest edit distance, with corpus frequency as the tie-breaker (Eq. 2).

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

In applying lexical approximation to the Turkish-English task, we used as the feature $f(.)$ the root found by the morphological analyzer described in section 2.1.

## 3. Arabic-specific system features

The architecture of our Arabic-to-English translation system is similar to our 2008 submission [6]. We applied an orthographical normalization to all training and test corpora, which was originally motivated to adapt the training corpus to the automatic speech recognition (ASR) outputs. Since the ASR outputs never contained any of the eight Arabic characters [ٓ ٔ ٕ ٖ ٗ ٘ ٙ ٚ], we removed all occurrences of these from the training set. Also, the alif variants [ٱ أ إ ٲ] were used much less frequently in the ASR outputs; in particular, [ٱ ٲ] never appeared at the beginning of a word. So, we normalized all occurrences of ٱ and the word-initial occurrences of أ and ٲ to ا. Aside from the significant performance improvement on the "ASR" translation task, this normalization also slightly benefited the "clean" task performance, therefore we adopted it into our system architecture.

We applied a two-pass lexical approximation for Arabic OOV words, utilizing a different feature function $f(.)$ in each pass. In the first pass, the feature function returns the morphological root(s) of the word according to Buckwalter Arabic Morphological Analyzer [7]. For words that were not recognized in the first pass and still are OOV, a second lexical approximation pass is applied where the feature function now returns a "skeletonization" of the word obtained by removing all the vowels and diacritics.

Table 3 shows that the gains due to orthographical normalization and lexical approximation have large overlap (also discussed in [6]). Lexical approximation yields significant performance improvement in all test sets, while orthographical normalization benefited little over lexical approximation or even degraded for the 2009 test set.

*Table 3*: Arabic-English translation % BLEU scores with different system components enabled. ON: Orthographical normalization, LA: Lexical approximation.

| System | Dev6 | Dev7 | *Test 2009* |
|---|---|---|---|
| Baseline | 35.33 | 44.43 | *43.55* |
| LA only | 48.01 | 48.24 | ***49.84*** |
| ON only | 45.68 | 47.53 | *43.40* |
| ON + LA | **48.15** | **48.69** | *49.08* |

## 4. System development common to both tasks

We used the open-source statistical machine translation toolkit Moses [8] for training the translation models and for decoding. An N-gram English language model was trained using the SRI language modeling toolkit [9]. 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 script. All the BLEU scores reported in this paper are by default computed with the cased, punctuated system outputs.

Although we used 3-gram target language models in our systems, using 4-gram models as suggested by one of the reviewers result in better performance. Table 4 shows the effect of N-gram model order on the performance of our primary submission.

*Table 4*: Effect of increasing N-gram language model order on the % BLEU scores of our primary submission

| LM Order | Arabic-English | | | Turkish-English | | |
|---|---|---|---|---|---|---|
| | Dev6 | Dev7 | *Test 2009* | Dev1 | Dev2 | *Test 2009* |
| 3 | **49.61** | 50.52 | *49.33* | 62.59 | 59.86 | *55.82* |
| 4 | 49.50 | 50.91 | ***50.38*** | 63.31 | **60.33** | ***57.24*** |
| 5 | 49.60 | **51.18** | *50.34* | **63.48** | 60.27 | *56.90* |

Similar to our 2007 and 2008 systems, we made use of phrase table augmentation. For source vocabulary words that are not included in the phrase table as a result of the phrase extraction process, this technique adds single-word phrase pairs derived from GIZA++[10]-produced lexical alignments to the phrase table. Hence, every word in the training source vocabulary is guaranteed to be provided translation hypotheses by the translation model during decoding. For some words, forcing the model to propose hypotheses as such may have the negative effect of generating incorrect translations in the output that could have been remedied by other methods (e.g., by lexical approximation in section 2.3), but in our previous experience the benefits of phrase-table augmentation outweigh the harm [5,6].

Among the provided development corpora, the two most recent sets were reserved for tuning the parameters and internal testing (devsets 1-2 for Turkish and devsets 6-7 for Arabic). The remaining corpora were used in training. Hence, for the Arabic-to-English system, devsets1-3 were also added with their 16 references as a training parallel corpus.

## 5. Results and discussion

Table 5 compares the performance of our three Turkish-English submissions (sections 2.1-2.3) on the development sets and the 2009 test set. The official evaluation results of our submitted Arabic-English and Turkish-English systems (primary and contrastive) are shown in Tables 6 and 7, respectively. Note that while the official submission of our contrastive-1 system (Table 7) was buggy, Table 5 is calculated on the correct system output. Among the three morphological approaches for Turkish, using morphological analysis customized to the translation task performed the best (primary submission). Also, the word-based lexical approximation approach performed close to unsupervised segmentation, even though it was outperformed during development experiments. Using a much larger monolingual Turkish corpus in segmentation model training might improve the accuracy of the unsupervised segmentation and its translation performance.

*Table 5*: Comparison of % BLEU scores of the developed Turkish-English systems

| System | Dev1 | Dev2 | *Test 2009* |
|---|---|---|---|
| Primary submission | **62.59** | **59.86** | ***55.82*** |
| Contrastive-1 | 60.74 | 57.97 | *53.27* |
| Contrastive-2 | 57.94 | 55.69 | *53.45* |
| Baseline | 56.58 | 54.24 | *52.51* |

### *5.1.1. Target language morphological analysis*

We also experimented with applying morphological analysis on the target language (English). In the scenario of section 2.2, we also investigated applying a similar unsupervised segmentation to the English side of the parallel texts as well. On one hand, there is some morphology encoded in the English words (such as the plural or gerund markers) favoring morphological analysis; but on the other hand, many words are in their root form and risk degrading the system performance if wrongly segmented by the morphological analyzer. For example, Figure 6 shows example segmentations found by Morfessor for the low-frequency words on the English training set.

| | Frequency | Segmentation |
|---|---|---|
| (a) | 1 | far + m |
| | 1 | pal + m |
| | 2 | to + m |
| | 1 | tour + is + m |
| | 1 | wor + m |
| (b) | 1 | wor + e |
| | 1 | wor + se |
| | 1 | wor + ship |

*Figure 6*: Examples of false segmentation on the English corpus. The non-linguistic morpheme "m" is proposed by Morfessor for the words in (a). Note that all the remaining parts are regular words by themselves, except for "wor", which is another proposed morpheme that accounts for other low-frequency, unrelated words in (b).

In our experiments, training a segmentation model for the English side and using it in system training did not provide a clear improvement over leaving the English corpus as words, as shown in Table 8.

*Table 8*: Effect of segmenting the English corpus via Morfessor on the translation % BLEU scores

| Segmented corpus | Dev1 | Dev2 | *Test 2009* |
|---|---|---|---|
| Turkish only | **60.02** | 56.48 | ***53.76*** |
| Turkish and English | 59.74 | **56.76** | *53.45* |

One reason for performance degradation could be the added complexity of generating roots and morphemes at the decoder output, since the output tokens are more in number and the reordering search space is much larger. Another reason could be the errors in English morphological segmentation. Using a linguistic-based English morphological

analyzer could generate a more accurate segmentation and might more consistently improve the translation performance, especially if the source-side is also analyzed with the same approach, as done in [3].

## 6. Acknowledgement

We would like to thank Kemal Oflazer from Sabancı University for providing the Turkish morphological analyzer.

## 7. References

[1] K. Oflazer, "Two-level description of Turkish morphology", *Literary and Linguistic Computing*, vol. 9, no. 2, pp. 137-148, 1994.

[2] H. Sak, T. Güngör, and M. Saraçlar, "Morphological disambiguation of Turkish text with perceptron algorithm", in *CICLing 2007*, LNCS vol. 4394, pp. 107-118, 2007.

[3] K. Oflazer, "Statistical machine translation into a morphologically complex language", *in CICLing 2008*, LNCS vol. 4919, pp. 376-387, 2008.

[4] 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.

[5] C. Mermer, H. Kaya, and M.U. Doğan, "The TÜBİTAK-UEKAE Statistical Machine Translation System for IWSLT 2007", *Proc. of the International Workshop on Spoken Language Translation (IWSLT)*, Trento, Italy, pp. 176-179, 2007.

[6] C. Mermer, H. Kaya, O.F. Gunes, and M.U. Doğan, "The TÜBİTAK-UEKAE Statistical Machine Translation System for IWSLT 2008", *Proc. of the International \Workshop on Spoken Language Translation (IWSLT)*, Hawaii, USA, 2007.

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

[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, 2007.

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

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

*Table 6*: Official automatic evaluation results for the submitted Arabic-English system

| | bleu | meteor | f1 | prec | recall | wer | per | ter | gtm | nist |
|---|---|---|---|---|---|---|---|---|---|---|
| case+punc | 0.4933 | 0.7327 | 0.7670 | 0.7845 | 0.7501 | 0.3634 | 0.3308 | 30.410 | 0.7410 | 7.6512 |
| no_case+no_punc | 0.4712 | 0.6866 | 0.7229 | 0.7482 | 0.6994 | 0.4236 | 0.3754 | 34.865 | 0.7105 | 7.6827 |

*Table 7*: Official automatic evaluation results for the submitted Turkish-English systems

| case+punc | bleu | meteor | f1 | prec | recall | wer | per | ter | gtm | nist |
|---|---|---|---|---|---|---|---|---|---|---|
| primary | **0.5582** | **0.8120** | **0.8328** | 0.8396 | **0.8262** | **0.3267** | **0.2676** | **25.219** | **0.7792** | **8.6018** |
| contrastive1 | 0.5112 | 0.7500 | 0.8008 | **0.8529** | 0.7547 | 0.3737 | 0.3204 | 28.985 | 0.7317 | 6.8455 |
| contrastive2 | 0.5345 | 0.7647 | 0.8015 | 0.8312 | 0.7737 | 0.3486 | 0.2989 | 27.611 | 0.7496 | 7.6529 |
| no_case+no_punc | bleu | meteor | f1 | prec | recall | wer | per | ter | gtm | nist |
| primary | **0.5385** | **0.7763** | **0.8008** | 0.8122 | **0.7897** | **0.3721** | **0.2932** | **29.029** | **0.7649** | **9.0226** |
| contrastive1 | 0.4927 | 0.7028 | 0.7573 | **0.8200** | 0.7035 | 0.4335 | 0.3585 | 33.444 | 0.7105 | 6.7275 |
| contrastive2 | 0.5132 | 0.7256 | 0.7659 | 0.8023 | 0.7326 | 0.4026 | 0.3368 | 31.872 | 0.7238 | 7.6772 |