AppTek Turkish-English Machine Translation System Description for IWSLT 2009

Selçuk Köprü

AppTek, Inc.
METU, Ankara, TR
skopru@apptek.com

Abstract

In this paper, we describe the techniques that are explored in the AppTek system to enhance the translations in the Turkish to English track of IWSLT09. The submission was generated using a phrase-based statistical machine translation system. We also researched the usage of morpho-syntactic information and the application of word reordering in order to improve the translation results. The results are evaluated based on BLEU and METEOR scores. We show that the usage of morpho-syntactic information yields 3 BLEU points gain in the overall system.

1. Introduction

In the IWSLT09 evaluation campaign, we focused on the Turkish to English BTEC task. This paper describes the statistical machine translation system that is employed to generate the submissions for IWSLT09. We used a Turkish morphological analyzer and chart parser in the preprocessing step of the statistical machine translation (SMT) system.

There exists only a few studies that deal with Turkish SMT [1, 2, 3, 4]. The ones that are available take Turkish as the target language. There are important and interesting challenges in this field that take root from the nature of the Turkish language. We see this evaluation campaign as an important opportunity that will boost research on Turkish SMT.

The rest of the paper is organized as follows: In the following section, an overview of the baseline SMT is given. Next, we give information on Turkish syntax and morphology in Section 3 from an SMT point of view. Section 4 describes the applied preprocessing steps in detail. Next, the results are presented briefly in Section 5. Finally, Section 6 concludes the paper.

2. The Baseline System

The baseline system is trained on lowercased and tokenized data without any factors. A 5-gram language model with Kneser-Ney smoothing is built using the IRSTLM toolkit [5]. The language model is quantized and compiled in a memory mapped model in order to allow for space savings and quicker upload of the model.

The core of the baseline system is based on the Moses toolkit [6]. The alignments are created using GIZA++ [7]. The maximum length of phrases that are entered into the phrase table is set to 15. All other parameters to the Moses training script are kept at their default values and no factors are used. The phrase table and the reordering table are binarized after the training.

The default weights in the configuration file are tuned and optimized with minimum error rate training (MERT) by using `devset1` data. Weight optimization is carried out on BLEU scores [8].

In order to complete the entire SMT system, we trained a recaser using the 20K BTEC `train` data and Moses. The translation that is obtained for `devset2` is recased and detokenized before evaluating the quality based on the BLEU metric.

3. Challenges on Turkish-English SMT

In this section we briefly explore the challenges that prevent the construction of successful SMT systems as in other language pairs, such as English-German. The divergence of Turkish and English puts a rocky barrier in building a prosperous machine translation system. Morphological and syntactic preprocessing is important in order to converge the two languages.

The basic word order in Turkish is SOV but it is highly flexible if compared to the rigid SVO word order in English. It is possible to encounter any kind of word order variation in Turkish except VOS. Similar facts about the grammatical differences between both languages make English-Turkish a difficult language pair. In Turkish, the head element is in the final position in the phrase. The main constituent is usually preceded by the modifiers and specifiers. If compared to English, the main constituent is moved to the beginning of the phrase as in (1). The differences in the word orders require to come up with effective reordering solutions in the training data.

(1) senin masanın üstünde
you-GEN table-POSS above
'on your table'

The subject in a Turkish sentence can be dropped if it is a

pronoun because verbs are always marked with PERSON and NUMBER information. This property introduces a hard complication into the translation process. The missing subject has to be generated in the English text at a sentence initial position as shown in (2).

(2) gazeteyi okudum
newspaper-ACC read-PAST-1SG
'I read the newspaper'

The predicative structure in a Turkish sentence can be a copula. The verb *'be'* in English has to be generated in an appropriate form for translating copulative constructions as shown in (3).

(3) Ahmet müdür
Ahmet manager
'Ahmet is the manager'

Turkish is a highly agglutinative language with a rich set of suffixes. Inflectional and derivational productions introduce a big growth in the number of possible word forms. The richness in morphology introduces many challenges to the translation problem both to and from Turkish.

In general, nouns in Turkish inflect for person (1, 2 and 3), number (plural:PL and singular:SG), case (nominative:NOM, accusative:ACC, dative:DAT, genitive:GEN, ablative:ABL, locative:LOC and instrumental:INST). Definiteness of a noun is implied with the accusative case marker. Nouns also show the possessor POSS agreement.

Turkish verbs inflect for person (1, 2 and 3), number (plural:PL and singular:SG), voice (active:ACT and passive:PASS), tense (past:PAST, present:PRES, future:FUT and aorist:aor), mood (indicative:IND, conditional:COND and necessitative:NECC) and polarity (positive:POS and negative:NEG). The verb can also denote many other aspectual information like ability, continuation and completion etc. The infinitive (INF) form is also marked with a special morpheme. The sample sentence in (4) demonstrates medium density inflection on a noun and a verb.

(4) arabalarımızla geldiler
car-PL-POSS-INST come-PAST-3PL
'They came with our cars'

Another challenge that is worth mentioning in this section is the richness in the ambiguity in Turkish morphological analysis. Derivational morphology is an important tool to generate new words. It is very common that certain wordforms can be analyzed in multiple ways.

4. Preprocessing

4.1. Morphological Analysis

We have used a morphological analysis (MA) system that employs a finite state transducer (FST) augmented with unification based feature structures (FS). The MA system is explained in detail in [9]. In the system, manually crafted Turkish morphology rules are compiled into finite state machines.

```
cam|cam|N|N|cam
kenarına|kenar|N|N.POSS.2SG.DAT|kenar+ın+a
kenarına|kenar|N|N.POSS.3SG.DAT|kenar+ı+na
yakın|yakın|ADJ|ADJ|yakın
bir|bir|QFR|QFR|bir
masaya|masa|N|N.DAT|masa+ya
geçmek|geç|V|V.INF|geç+mek
istiyoruz|iste|V|V.AOR.1PL|iste+iyor+uz
```

Table 1: Morphological analysis output for a sample sentence.

These analysis rules are used to derive the uninflected form of an inflected word. The rule definition is composed of morpheme categories bundled with regular expression formalism.

Figure 1 depicts the simplified verb rule and the feature structures that are associated to the morpheme categories. The final feature structure that represents the input verb is built by unifying the morpheme feature structures.

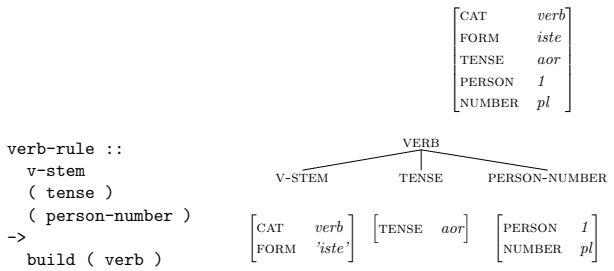

Figure 1: Morphological analysis rule for the verb and the unifying feature structures.

At the end of the analysis, the MA system is capable of producing output in Moses factored format. The example output for the sentence in (5) is given in Table 1.

(5) cam kenarına yakın bir masaya geçmek istiyoruz
window side-POSS-3SG-DAT- close a table-DAT migrate-INF want-AOR-1PL
'We want to move to a table close to the window'

4.2. Syntactic Analysis

Morphological analysis is carried out at the word context and it is not enough to resolve the ambiguities. Therefore, we have performed syntactic analysis in order to find out which of the MA outputs are selected in the parse tree. For example, in Table 1, there exist two different analyses for the word *kenarına, ('to his side' vs. 'to your side')*. This ambiguity can be resolved only in the phrase context.

We used a syntactic analyzer that utilizes a chart parser in which the rules modeling the source language grammar are

augmented with feature structures. The parser is presented in detail in [10]. The Turkish grammar is implemented manually using Lexical Functional Grammar (LFG) paradigm. The primary data structure to represent the features and values is a directed acyclic graph (dag). The system also includes an expressive Boolean formalism, used to represent functional expressions to access, inspect, or modify features or feature sets in the dag. Complex feature structures (e.g., lists, sets, strings) can be associated with lexical entries and grammatical categories using inheritance operations. Unification is used as the fundamental mechanism to integrate information from lexical entries into larger grammatical constituents.

A sample parse tree and the corresponding feature structures are shown in Figure 2. For simplicity, many details and feature values are not given. The attribute-value matrix containing the information originated from the lexicon and the information extracted from morphological analysis is shown on the leaf levels of the parse tree in the figure. The final feature structure corresponding to the root node is built during the parsing process in cascaded unification operations specified in the grammar rules.

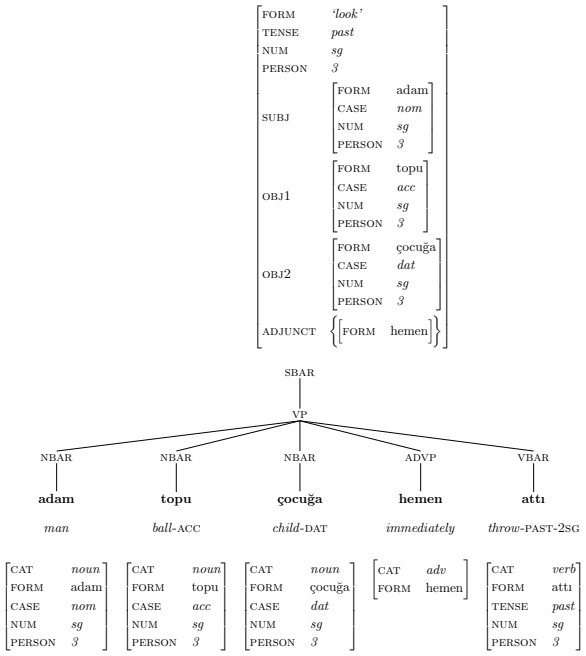

Figure 2: Syntactic analysis and the unifying feature structures.

4.3. Suffix Splitting

At the end of the syntactic analysis, the parser outputs the MA hypotheses that are selected in the winning parse tree. The stems and the suffix boundaries are marked in the parser output. The suffixes are detached from the stem using the boundary information. Next, the suffixes have to be normalized in order to reduce the number of different forms in the corpus. For example, the final form for the sample sentence in (6a) is shown in (6b).

(6) a. bu düğmeyi dikebilir misin ?
this button- sew- INTER- ?
ACC ABIL 2SG
'Can you sew this button?'

b. `bu düğme +yI dik +AbIl +Ir mI +sIn ?`

In Turkish, the suffixes are formed according to vowel harmony rules: The type of the vowel in the suffix must be of the same type of the last vowel in the word stem. That is, a word with a *front* vowel in the last syllable can be followed by a suffix which has *front* vowels only. Similarly, *back* vowels can be followed only by *back* vowels. This fact can be seen in (7a): The word '*fare*' (*mouse*) takes the *-ler* plural marker because the last vowel *'e'* is a front vowel. However, the word '*insan*' (*human*) takes the *-lar* plural marker because the last vowel *'a'* is a back vowel. As a result, the same suffix can have different surface forms when attached to different words because of vowel harmony. The vowels in the suffixes are normalized in order to reduce the number of different suffixes. For example, the two distinct plural markers are normalized as `-lAr` as shown in (7b). Capital letters in the suffixes represent normalized forms.

(7) a. fareler ve insanlar
mouse-PL and human-PL
'(Of) mice and men'

b. `fare +lAr ve insan +lAr`

Table 2 shows the word count statistics for the training corpus. The first two rows in the table show the numbers for the original corpus. The third row, labeled as `tr.split1`, represents the Turkish corpus where all suffixes are split. In `tr.split1`, the number of unique words drops down to 7151, however the average sentence length increases to 10.54. Next, in order to equalize the average sentence length, we only split words that have a frequency less than a threshold value which is calculated to be 24. The `tr.split2` represents the corpus where only the less frequent words are split. In this case, the average sentence length in Turkish gets close to the length in English. The system parameters are retuned for the `split1` and `split2` experiments.

4.4. Word Reordering

Although there are many word reordering possibilities between Turkish and English as explained in Section 3, we only reordered the Turkish infinitive marker. For example, the sentence in (8a) is reordered as shown in (8b).

		# of lines	# of words	avg. sent. length	unique words
1	en	19972	189K	9.46	7182
2	tr	19972	140K	7.05	17265
3	tr.split1	19972	210K	10.54	7151
4	tr.split2	19972	189K	9.45	7822

Table 2: Word count statistics for the training corpus.

(8) a. okumak istiyorum
read-INF want-PRES-1SG
'I want to read'

b. `oku +mAk iste +yor +Im`
`+mAk oku iste +yor +Im`

5. Results and Discussions

In this section we present the evaluation scores that are obtained after applying the preprocessing steps. The `devset2` file is used as the test file and it contains 16 reference translations. The results are listed in Table 3. The same preprocessing steps that are applied to the training data are also applied to the test input sentences. The `baseline` system is the one that is described in Section 2. Unfortunately, the preprocessing steps did not help to improve the BLEU scores. Splitting all suffixes (`split1`) resulted in 1 point drop in the BLEU score and splitting less frequent words (`split2`) resulted in 4 point drop in the BLEU score. The last row in Table 3 shows the scores of our rule-based MT system. The rule-based system is not trained on the training data, therefore, it is solely a *constraint* system. The scores of the rule-based MT system is only listed for comparison purposes.

	System	BLEU	METEOR
1	baseline	51.66	57.83
2	split1	50.62	63.17
3	split2	48.38	61.77
4	rule-based	20.13	26.48

Table 3: BLEU and METEOR scores for the different systems.

The METEOR scores, however, indicate the opposite of the BLEU scores. There are more than 5 points gain in the METEOR score if all suffixes are split (`split1`). The gain is close to 4 METEOR points if less frequent words are split (`split2`). Although, the BLEU scores in Table 3 indicate that the suffix splitting attempt did not pay off as expected, the disagreement between the METEOR and BLEU scores foster further investigation. In search for the reason of the BLEU score drop, we did a second round of experiments after increasing the default distortion limit from 6 to 12. The motivation behind this experiment is that the splitting increases the phrase length and this length increase might require large-scale reorderings.

The new scores with the new distortion limit value are presented in Table 4. The new scores are consistent with our expectations. The `split1` system is the best performing system both in terms of BLEU and METEOR scores. There is a 3 BLEU point gain if compared to the baseline system. In terms of METEOR scores, the difference between the baseline and the `split1` system is more than 7 points. The `split2` system did not perform better than the baseline system even with the new distortion limit value. This result indicates that it has a negative effect on the translation quality if the splitting is not performed in a consistent manner.

	System	BLEU	METEOR
1	baseline	52.33	57.97
2	split1	55.32	65.37
3	split2	50.05	62.55

Table 4: BLEU and METEOR scores after increasing the distortion limit to 12.

6. Conclusions

We have described our Turkish-to-English SMT system for the IWSLT09 task in detail. The different challenges in machine translating from Turkish to English are highlighted at the beginning of our paper. We also discussed the preprocessing steps that are implemented in the system. We have explored two different suffix splitting schemes and their effects on the BLEU and METEOR scores. The results imply that the applied preprocessing steps degraded the BLEU scores by 1 point if a small distortion limit was used. The gain from the preprocessing steps was 3 BLEU points with a higher distortion limit. The METEOR scores showed the positive effect of the morpho-syntactic preprocessing steps even with a small distortion limit. The gain was 5.3 METEOR points with a small distortion limit and it was 7.4 points with a higher distortion limit. It is important to notice that evaluation using a single metric can be misleading.

We continue to explore different preprocessing and reordering schemes in order to further improve the overall system performance. We also investigate the possibility of incorporating the experience from the processing of other morphologically rich languages like Arabic as in [11].

7. Acknowledgments

Thanks to Evgeny Matusow and Hassan Sawaf for their comments and significant support. Thanks also to the reviewers for their valuable comments.

8. References

[1] K. Oflazer and İ. D. El-Kahlout, "Exploring different representational units in English-to-Turkish Statistical MT," in *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 25–32.

[2] İ. D. El-Kahlout and K. Oflazer, "Initial explorations in English to Turkish Statistical Machine Translation," in *Proceedings of the Workshop on Statistical Machine Translation*. New York City: Association for Computational Linguistics, 2006, pp. 7–14.

[3] K. Oflazer, "Statistical Machine Translation into a morphologically complex language," in *Proceedings of the CICLing 2008 Conference on Intelligent Text Processing and Computational Linguistics*. Haifa, Israel: Springer-Verlag, 2008, pp. 376–387.

[4] C. Tantuğ, E. Adalı, and K. Oflazer, "A mt system from turkmen to turkish employing finite state and statistical methods," in *MT-Summit XI: The 11th Machine Translation Summit*. Copenhagen, Denmark: EAMT, 2007.

[5] M. Federico, N. Bertoldi, and M. Cettolo, "IRSTLM: an open source toolkit for handling large scale language models," in *Interspeech 2008*. ISCA, 2008, pp. 1618–1621.

[6] 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 *Annual Meeting of the Association for Computational Linguistics (ACL)*. Prague, Czech Republic: Association for Computational Linguistics, 2007.

[7] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.

[8] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002, pp. 311–318.

[9] S. Köprü and J. Miller, "A unification based approach to the morphological analysis and generation of arabic," in *3rd Workshop on Computational Approaches to Arabic Script-based Languages at MT Summit XII*, A. Farghaly, K. Megerdoomian, and H. Sawaf, Eds. Ottowa, Canada: IAMT, August 26th 2009.

[10] S. Köprü and A. Yazıcı, "Lattice parsing to integrate speech recognition and rule-based machine translation," in *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Athens, Greece: Association for Computational Linguistics, March 2009, pp. 469–477.

[11] F. Sadat and N. Habash, "Combination of arabic preprocessing schemes for statistical machine translation," in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 1–8.