AppTek's *APT* Machine Translation System for IWSLT 2010

Evgeny Matusov *Selçuk Köprü*

AppTek Inc.
Aachen, Germany
ematusov@apptek.com

AppTek Inc.
Ankara, Turkey
skopru@apptek.com

Abstract

In this paper, we describe AppTek's new *APT* machine translation system that we employed in the IWSLT 2010 evaluation campaign. This year, we participated in the Arabic-to-English and Turkish-to-English BTEC tasks. We discuss the architecture of the system, the preprocessing steps and the experiments carried out during the campaign. We show that competitive translation quality can be obtained with a system that can be turned into a real-life product without much effort.

1. Introduction

In the IWSLT 2010 evaluation campaign, AppTek participated in the Arabic-to-English and Turkish-to-English BTEC tasks. This paper describes AppTek's new *APT* machine translation (MT) system that constitutes the core of AppTek's commercial MT products starting September 2010. We used this high-quality and efficient state-of-the-art MT engine to generate the submissions for the IWSLT evaluation this year. Our experimental efforts focused mainly on using novel reordering models, combining different alignment approaches and on utilizing multiple morphological segmentations of source words.

While aiming at the best possible translation quality, we decided not to include any steps in the translation pipeline which would make a real-time application of the translation system impossible. We did not perform re-scoring of the N-best lists nor system combination of different systems or system variants. This allowed us to create a MT system that produced our primary submission (ranked 3rd and 4th in terms of BLEU on the Arabic-to-English and Turkish-to-English BTEC tasks, respectively) with the translation speed of 12 words per second[1].

The rest of the paper is organized as follows. In the following section, details about the preprocessing steps for the language pairs is given. Next, we describe the baseline system that is trained for the BTEC evaluation tasks. In section 4, we discuss the experiments and their evaluations. Finally, we conclude the paper in section 5.

[1]Using a single core of a 3.0 GHz CPU.

2. Preprocessing

In this section we briefly explain the preprocessing steps carried out for Arabic-to-English and Turkish-to-English statistical MT (SMT). The data preparation steps play an important role in the success of the overall MT performance. This role can be understood better if the inflectional structure of Arabic morphology and the agglutinative structure of Turkish morphology is taken into account. The word formation in English is not as productive as in Arabic or as in Turkish. This kind of morphological divergence obligates us to perform a reasonable amount of preprocessing for the success of the MT system.

In the following, we describe our approach to the morphological analysis of both languages. We use the same in-house developed morphology analysis toolkit for this purpose. Following the language-independent explanations, we discuss the language-specific details in the preprocessing steps for Arabic and Turkish.

2.1. Morphological analysis

Our approach to morphological analysis is explained in detail in [1] with focus on Arabic. However, the same approach and toolkit is used for all languages that we deal with. The approach employs finite-state transducers (FST) enriched with unification of *feature structures*. The FST is not visible to the grammar developer who writes the morphological analysis rules in *Lexical Functional Grammar* (LFG) formalism. Manually crafted rules similar to regular expression patterns are compiled into FST.

The exploited LFG formalism allows the declaration and formulation of linguistic knowledge in a concise manner. The system also includes an expressive Boolean formalism, used to represent functional equations to access, inspect or modify features or feature sets attached to the morphemes. Complex feature structures (e.g. lists, sets, strings, and conglomerate lists) 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. The same framework has been used successfully for inflectional and agglutinative languages.

2.2. Arabic preprocessing

In the preprocessing step of Arabic, a full-fledged morphological analyzer and a part-of-speech (POS) tagger were exploited.

2.2.1. Arabic morphological segmentation

The entire Arabic morphology is divided into two main categories in the implementation of the analysis rules: verbal categories and nominal categories. For the verbal categories, we analyze the following grammatical features in Arabic: tense (*perfective*, *imperfective*), number (*singular*, *dual*, *plural*), person (*1*, *2*, *3*), gender (*masculine*, *feminine*), mood (*indicative*, *subjunctive*, *imperative*, *jussive*), voice (*active*, *passive*) and modality (*future*). Personal pronouns can be attached to the end as suffixes. Additionally, there exist coordination and subordination prefixes attached to verbs. Nouns in Arabic inflect for number (*singular*, *dual*, *plural*), definiteness (*definite*, *indefinite*), case (*nominative*, *accusative*, *genitive*), person (*1*, *2*, *3*), gender (*masculine*, *feminine*). Similar to verbs, personal pronouns can be attached to the end as suffixes but with a different functional role this time.

The output of the morphological analyzer is processed further to achieve better alignments with the English translations. The basic idea is to detach morphemes on the Arabic side, assuming that they correspond to English words. The morphologically segmented corpus is created by identifying prefixes, suffixes, as well as the definite article, detaching them from the rest of the word concerned, and marking them with symbol "_".

2.2.2. Arabic POS tagging

An efficient Hidden Markov Model (HMM) based POS tagger is utilized in the Arabic reordering model (see Section 3.3). Despite the morphological complexity of Arabic, our POS tagging approach is a data driven approach and does not utilize any morphological analyzer or a lexicon as many other Arabic POS taggers. This makes the tagger very efficient and valuable to be used in other applications. The obtained accuracy results are still comparable to alternative Arabic POS taggers. It achieves an accuracy of 95.6% using a standard tag set of 17 tags on a test file sampled randomly from the ATB corpus [2] containing 984 sentences and over 12 K tokens of which 5.6 K are unique. The tags predicted by this tagger were used for the POS-based reordering model described in Section 3.3.

2.3. Turkish preprocessing

The main focus in Turkish preprocessing is on morphological analysis because of the agglutinative nature of the language. In Turkish, the surface forms of the words are built by suffixation of inflectional and derivational morphemes. FST-based systems like the one explained in section 2.1 are successful in modeling the morphotactics and morphophonemic rules of Turkish. Turkish words have a clear but complex morphotactic structure. The complexity introduces a very productive word formation scheme. For example, circular constructions can appear in a single word. Moreover, distinctive morphophonemic rules, e.g. harmony, epenthesis, duplication, etc., contribute further to the complexity of word formation. Inflectional and derivational productions introduce a big growth in the number of possible word forms. Naturally, only inflectional morphology is aimed to be analyzed. The richness in morphology introduces many challenges to the translation problem both to and from Turkish. The grammatical categories handled in the morphological analysis are similar to the categories handled in our system of previous year [3]. Morphological disambiguation is performed by syntactic analysis of the sentence. At the end of the syntactic analysis, the parser outputs the MA hypotheses that are selected in the winning parse tree. The morpheme boundaries are marked in the parser output with symbol "_". The morphemes are detached from each other to reduce the out-of-vocabulary (OOV) rate and to achieve better alignments. The detached suffixes are normalized in order to reduce the number of different forms that result because of the morphophonemic rules in Turkish.

3. Baseline MT system

3.1. Statistical phrase-based approach

AppTek's baseline MT system is a state-of-the-art phrase-based statistical translation system similar to [4] and [5]. In this system, a target language translation $e_1^I = e_1 \dots e_i \dots e_I$ for the source language sentence $f_1^J = f_1 \dots f_j \dots f_J$ is found by maximizing the posterior probability $Pr(e_1^I | f_1^J)$. This probability is modeled directly using a log-linear combination of several models. The best translation is found with the following decision rule:

$$\hat{e}_1^{\hat{I}} = \operatorname*{argmax}_{I, e_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \quad (1)$$

The model scaling factors λ_m for the features h_m are trained with respect to the final translation quality measured by an automatic error criterion [6]. The baseline system includes an n-gram language model, a phrase translation model, and a word-based lexicon model as the main features. The latter two models are used in both directions: $p(f|e)$ and $p(e|f)$. The phrase-based and word-based models are estimated from the training data using word alignments trained with GIZA++ [7]. For phrase extraction, we use the implementation in the Moses statistical MT toolkit [4]. Further log-linear model features include a word penalty and a phrase penalty, as well as the target n-gram language model (LM). Finally, reordering models are also used as features.

The phrase-based search consists of two parts. First, those contiguous phrases in the source sentence are identified which have translation candidates in the phrase table. This phrase matching is done efficiently using an algorithm based on the work of [5]. The second phase is the source cardinality-synchronous search (SCSS) implemented with dynamic programming. The goal of the search is to find the most probable segmentation of the source sentence into K non-empty non-overlapping contiguous blocks, select the most probable permutation of those blocks, and choose the best phrasal translations for each of the blocks at the same time. The concatenation of the translations of the permuted blocks yields a translation of the whole sentence.

3.2. Reordering penalty model

In a recent attempt to improve the word order of the generated translations, we introduced a *run-based* penalty model for reordering [8]. In related work, the basic costs for reordering a phrase in the SMT search are usually linear in the distance [9]. More complex models have been introduced, but in most cases they extend the simple distance-based distortion model.

In [8], we argue that the distance-based model is not the best choice even for the basic reordering model in statistical MT. The absolute distance is usually not a good indicator if a reordering should take place or not. Also, the distance-based model can penalize linguistically very improbable reorderings less than the other, more reasonable reorderings. For instance, the model penalizes "jumps" back to the monotonic translation path, which is not reasonable in most situations. We decided to abandon the distance-based model completely, and replace it with a model that assigns a penalty for each new deviation from the monotonic translation path. A *run* is a contiguous sequence of covered source word positions in a partial translation hypothesis. At any point in the search, we allow at maximum m runs. If m is set to 1, the search becomes monotonic. A penalty λ_{nr} is added to the total translation costs only if the new hypothesis that extends a previous one by translating a candidate phrase $\tilde{f}$ has a higher number of runs than the previous hypothesis. We define this to be the case when the left neighbor of the first word in $\tilde{f}$ has not yet been covered. In such cases, a "new" non-monotonicity is introduced. To further distinguish between the runs, in practice we use three penalties: one for short-range reordering, when the new run is started from a position that is no more than 3 positions away from the last covered position in the hypothesis; one for medium-range reordering when the starting position of the new run is between 4 and 7 positions from the last covered position; and one for long-range reordering, used for jumps of more than 7 positions which start a new run. An optimal set of values for the 3 penalties can favor or discourage some of these reordering types. For example, short-range reorderings may be more likely for language pairs such as Arabic and English, and they can get a lower penalty, whereas long-range reorderings may be penalized more strongly.

In addition to the three run-based penalties, we introduce a feature that measures the length of the sentence part that has been translated non-monotonically in the current hypothesis. The feature is used to penalize reorderings which involve many words, as opposed to skipping e. g. 1-2 words for later translation. Details are described in [8].

3.3. POS-based and lexicalized reordering models

The run-based penalties described in the previous section do not depend on the actual words which start a new run. Therefore, we decided to introduce another model to capture this dependency. We make the event of opening a new run of a certain type t ($t \in \{$short, medium, long-range$\}$) depend on the word that actually starts the new run, and on the word that was translated last in the current MT hypothesis. Alternatively, we use POS tags instead of the actual words.

The model is estimated on the word-aligned training data. Given a word alignment that is a function of source word positions with all source words aligned, there is a unique way of reordering the source words so that the alignment with the target sentence becomes monotonic [10]. When comparing the reordered source sentence with the original one, it is straightforward to identify each word f that starts a new run of type t, as well as the word f' before it in the reordered sentence. From all training source sentences we extract tuples (t, f, f') and estimate the probabilities $p(t|f, f')$ which we then use as an additional feature in the translation process. The probabilities are estimated with proper smoothing (backing-off) similar to the estimation of LM probabilities. In the future, we plan to improve the estimation of this model using maximum entropy techniques.

3.4. Multiple segmentation paths

In Section 2 it was described that Arabic suffixes and prefixes (including articles) or Turkish suffixes can be detached from the rest of the word in order to reduce the vocabulary size and for better word alignment with the corresponding English sentence (morphological segmentation). It is not easy to determine what type of segmentation, if any, is best for machine translation quality. Therefore, we decided to consider multiple segmentations in the MT search. AppTek's *APT* MT engine is capable of translating paraphrases of each given token sequence. This functionality is achieved by allowing the decoder to match extra phrases associated with a particular sequence of source sentence positions. The algorithm is very similar to word lattice translation algorithms presented in [11] and [12]. To make use of this algorithm for our task, we took the source sentence with the finest morphological segmentation (the one resulting in the largest number of tokens). We then defined a paraphrase for each token sequence that started with a prefix (and thus could be joined with the next token) or ended with a suffix (and thus could be attached to the previous token). The case where there was both prefix and suffix or multiple affixes was also considered. An example of the paraphrases that we defined is given in Figure 1. Here, the token sequence that is replaced with the concatenation of 2 (or more) tokens is defined using the `<span>` tag. Multiple spans can be specified for the same

```
<s>
şehir
merkez   <span label="merkezi" len="2"/>
         <span label="merkezine" len="3"/>
_i       <span label="ine" len="2"/>
_ne
ne
kadar
?
</s>
```

Figure 1: *Example of alternative segmentations encoded in the APT input file.*

token sequence.

In the search, all of the segmentations specified are equally probable. In order for the MT system to actually match the joined words, besides using the morphologically segmented word-aligned training corpus for phrase extraction we also extracted phrases from the same training data without any segmentation and from the same data with only those words segmented which occur less than 100 times. The latter approach is somewhat similar to the CoMMA method tried in [13].

3.5. Language models

The IRSTLM Toolkit [14] was used for training the English language models for both Arabic-to-English and Turkish-to-English translation tasks. We applied improved Kneser-Ney smoothing in the training process. Only the target side of the provided bilingual training data was taken for the LM estimation, no external data was used. The English text was preprocessed before training the LMs. The preprocessing includes tokenization and replacement of all numbers with a special tag. In LM and translation model training, we true-cased the English side of the bilingual corpus: all words whose lowercase form was more frequent were converted to lowercase, excluding a closed list of manually specified exceptions. Thus, no true-casing step was necessary in post-processing of translation hypotheses, except for making each letter that started a sentence an uppercase letter.

In Section 4.1 it will be described how the best n-gram size for the LM was chosen experimentally.

3.6. Additional system features

Besides the models described above, we also experimented with a number of other enhancements to our system. In order to facilitate better word choice, we added the sentence-level inverse IBM model 1 scores as additional feature. The model is expressed with the following equation:

$$h_{\text{inv-ibm1}}(f_1^J, e_1^I) = \prod_{i=1}^{I} \left[\frac{1}{J} \sum_{j=1}^{J} p(e_i|f_j) \right] \quad (2)$$

This model has been used also by [15, 16]. The advantage of the model as compared to the standard IBM model 1 is that it can be used to score partial hypotheses directly in the search, since there is no dependency on the full target sentence.

Table 1: *Examples of improved sentence structure when OOV words are disambiguated with their POS tags.*

Baseline	May I have another UNK green?
+ OOV handling	May I have another green UNK_N?
Reference	May I have some more green beans?
Baseline	UNK dangerous this project is quite as the economy that depends quickly.
+ OOV handling	This project is quite dangerous to as that depends economy is UNK_V quickly.
Reference	The project is quite risky as economic circumstances change quickly.

For each sentence in the test data which fully matched a source sentence in the training data, we replaced the statistical translation of the sentence with the "translation memory", i. e. with the corresponding target language translation from the training data. This was done to ensure that at least the test sentences which have been observed in training will get a translation that is 100% correct. However, experiments showed that the automatic error measures degrade when we use the translation memory (see Section 4.2).

Finally, we tried a number of things on the Arabic-to-English task which try to tackle the problem of out-of-vocabulary words. The quite large number of OOVs that we observed when translating the development data can be explained by the small amount of parallel sentence-aligned data that we were given for training. We tried the following to deal with the OOVs:

- We replaced some OOVs with in-vocabulary words whose edit distance in letters to the OOV word was equal to 1. On devset7, out of 100 unknown types a replacement was found for 79. This has improved translations of a number of sentences, when the replacement word was a different full form of the same base form. In most cases, however, the replacement word was completely semantically unrelated to the original word. Overall, we observed no improvement in BLEU and decided not to use this technique for our final submission.

- Observing that OOVs often break the sentence structure (e.,g., they are often translated in a wrong position, and in their desired position a semantically incorrect phrase is formed), we decided to make use of POS information when "translating" OOVs. Since we were using a morphological analyzer with a larger vocabulary, for 94% of the OOVs the POS tag had been predicted. We replaced the occurrences of these OOVs in the development data with their POS tags. In training, when extracting phrase pairs from the word-aligned parallel corpus, we extracted additional phrase pairs by replacing all occurrences of a specific POS with its tag both in the source and in the target phrase. We first added phrase pairs by replacing all nouns, then by replacing all verbs (keeping the surface forms of the nouns), and finally, by replacing all adjectives. In the search, these additional phrase pairs could be matched only by those sentence parts which contained OOVs with corresponding POS tags. Again, when translating the development data with this extended phrase table, we observed no improvement of automatic MT error measures and decided against using this technique for the final submission. However, subjectively the structure of some translated sentences with OOVs improved (see examples in Table 1). We believe that this technique could be promising for real-life dialog application, where the person speaking the target language would see an unknown word, but may guess its meaning from the correct context.

4. Experimental Results

AppTek participated in Arabic-to-English and Turkish-to-English BTEC tasks, using the supplied training data of 20K sentence pairs only. As the criterion for optimization we used the the well-established automatic measure BLEU [17]. We computed BLEU without considering case, but considering punctuation. On the test data, we report results in terms of case-sensitive BLEU and translation edit rate (TER,[18]), and the punctuation marks are included in the evaluation.

In post-processing, we detokenized the MT output. We also restored most of contractions (e. g. "do not" → "don't"). It turned out that the restoration of contractions has a very large influence on the automatic MT error measures, since most of the reference translations use them very often.

4.1. Turkish-to-English BTEC task

For the Turkish-to-English task, we used `devset2` (500 sentences) for the optimization of model scaling factors in the experiments. The other development data `devset1` (506 sentences) is used as the test set. Each development set contains 16 references. For the primary submission, we added both `devset1` and `devset2` to the training data. Before adding the development data, the longest reference translation is selected.

Turkish data is all lowercased in the final system trained for the primary submission. However, each English word is lowercased only if it appeared at least once in lowercase except for a list of abbreviations.

4.1.1. Language Model Experiments

We first estimated different *n*-gram models to observe the effect of the *n*-gram level on the BLEU scores. The system is optimized on `devset2` and tested on `devset1` as mentioned before. Table 2 shows the BLEU scores for the various *n*-gram levels for the Turkish system. For the 16 `devset1` references, the perplexity ranges between 310.53 and 720.94 and the OOV value lies in between 4.04% and 7.74%. For `devset2`, the numbers range between 412.80 and 873.51 for perplexity and between 5.41% and 8.45% for OOV.

Table 2: *Various n-gram levels and the corresponding OOV and BLEU scores for the Turkish system.*

n-gram	Opt. BLEU	Test BLEU
3	57.15	59.56
4	57.73	60.67
5	57.31	**61.55**
6	57.66	60.81
7	**57.76**	60.96
8	57.58	60.74
9	57.61	59.86

4.1.2. Word Alignment Experiments

In this part, we have explored the effect of different heuristics on symmetrization of word alignments. Various symmetrization heuristics are applied to obtain different word alignments. Systems based on those different alignments are evaluated. We also merged multiple alignments into a single system by extracting phrases from concatenation of the same parallel corpus with different word alignments (this is similar to the alignment combination in phrase table training as described in [13], but we combine different alignment heuristics).

Table 3 lists the Tr-En BTEC task BLEU scores for different methods of word alignment symmetrization. As before, the system is optimized on `devset2` and tested on `devset1`. The best BLEU score is obtained by merging all alignments produced by the different heuristics. The '`subset merged`' row in the table indicates the system where only the best 5 alignments are merged into a single system. This system achieved the best optimized BLEU score, however, it is outperformed by the '`all merged`' system on the test data. The heuristic names in Table 3 indicate the combination of IBM model 4 alignments with ACL [7], "grow-diag-final" [9], "intersection", "intersection-union" and "unify" heuristics, respectively. The "left" and "right" heuristics indicate the standard and inverse direction of the GIZA++ IBM model 4 alignment, respectively. The "berkeley" alignment is obtained with Berkeley Aligner [19]. The best individual BLEU score is obtained with the "acl" heuristic.

Finally, we tested the sentence-level inverse IBM model 1 on the development data. The optimization BLEU score was 57.73 and test BLEU score was 60.59 which is almost 1 point below what we have obtained in the '`all merged`' system.

Table 3: *BLEU scores for different methods on symmetrization of word alignments in the Turkish to English BTEC task.*

Method	Optimized BLEU	Test BLEU
acl	57.18	59.87
gdf	53.75	56.41
intersection	55.80	59.27
iu	57.30	59.68
unify	52.57	55.69
left	56.04	59.65
right	54.02	56.74
berkeley	55.12	57.81
subset merged	**57.70**	61.14
all merged	57.31	**61.55**

4.1.3. Other Experiments

In this part, we report various other experiments that were carried out on the development data for the Turkish-to-English task. Table 4 lists the optimization scores and the test BLEU scores. The first experiment is about mapping the contractions in the output as explained at the beginning of this section. This increased the test BLEU score by 0.82 points from 61.55 to 62.37 as shown in the first row of Table 4.

Next, we have tested the multiple segmentation of the Turkish words as explained in Section 3.4. For this purpose, we have trained two different systems; one with splitting all morphemes and the other with no morpheme splitting at all. The phrase tables of the two systems are later merged into one in order to be able to handle alternative segmentations encoded in the *APT* input file. Unfortunately, this approach resulted in a decrease of the BLEU score by 0.63 points from 61.55 to 60.92, as shown in Table 4. This is different from the Arabic-to-English task, where a significant improvement was observed when using multiple segmentations (see Section 4.2).

The last three lines in Table 4 show the BLEU scores when the English data is truecased before the training. The primary submission is created with a system trained with truecased data and optimized on both `devset1` and `devset2`. Using truecased English data resulted in a small decrease of the BLEU score by 0.27 points from 62.37 to 62.10.

4.2. Arabic-to-English BTEC task

On the Arabic-to-English task, for most experiments we used the concatenation of devset2 (500 sentences) and devset6 (489 sentences) for optimization of model scaling factors. To track the progress of our MT system, we used devset7 (507 sentences) as a held-out test set. The rest of the available development sets were added to the training data with the longest reference translation per sentence each.

Table 4: *BLEU scores for other experiments in the Turkish to English BTEC task.*

	Optimized BLEU	Test BLEU
mapping contractions	58.22	62.37
multiple segmentations	57.93	60.92
truecased opt devset1	63.26	59.46
truecased opt devset2	60.87	62.10
truecased opt devset1+2	61.84	N/A

In our preliminary experiments, the optimal LM n-gram size in terms of BLEU turned out to be 4. We therefore used a 4-gram LM in all of the experiments reported below.

Table 5 summarizes the most important translation experiments on the Arabic-to-English task, evaluated with the official evaluation server on the IWSLT09 and IWSLT10 test sets. Lines 1-3 correspond to using a certain type of morphological segmentation both in training and translation: no segmentation, segmentation of words with frequency ≤ 100, and full morphological segmentation of Arabic prefixes, suffixes, and articles as described in Section 2. It is clear that without morphological segmentation the BLEU score drops by more than 2% absolute. Full morphological segmentation is comparable to segmentation of only infrequent words in terms of MT quality on both test sets.

For experiments in lines 1-3, we joined corpora with different word alignments prior to phrase extraction. Similarly to the experiments on the Turkish-to-English task, we took the 5 alignments which yielded the best results in terms of BLEU when used individually. This was the standard direction of the GIZA++ IBM model 4 alignment, as well as the combination of IBM model 4 alignments with the heuristics “intersection-union”, “grow-diag-final”, and ACL [7]. We also added the HMM model alignments combined with the ACL heuristic. The MT system that used only the ACL heuristic combining the IBM model 4 alignments was the best individual system. The result for this system is given in line 9 in Table 5. We see that using multiple alignments for phrase extraction only improved BLEU and TER results on the IWSLT10 set (e. g. TER from 33.6 to 32.3%). In contrast, we observed a non-significant degradation of the scores on the IWSLT09 set. Thus, the alignment combination was not as effective as on the Turkish-to-English task.

The differences in the performance of our MT system on the two test tests were also clear in the next experiment. Using the ability of the *APT* MT engine to consider multiple input paths (see Section 3.4), we marked the Arabic definite article and the accusative marker as optional for translation. Thus, the decoder could choose not to translate them at all. This improved the automatic measures slightly, but only on the IWSLT 2010 set (line 4 vs. line 3 in Table 5).

In the next step, we also allowed multiple morphological segmentations in the input. For this experiment, we extracted phrases from all the 5 alignments of all 3 morphological seg-

Table 5: *AppTek's contrastive systems on the BTEC Arabic-to-English task.*

		IWSLT 09		IWSLT 10	
		BLEU [%]	TER [%]	BLEU [%]	TER [%]
1	no morphological segmentation	49.2	30.9	41.0	35.8
2	morphological segmentation for words w with $N(w) \leq 100$	52.0	29.2	45.3	32.5
3	full morphological segmentation	51.5	29.8	45.2	32.3
4	+ optionally remove article/accusative marker	51.2	30.0	45.7	32.0
5	+ multiple morphological segmentation paths	52.3	29.1	46.6	31.9
6	+ POS-based reordering model	51.5	28.9	46.0	31.8
7	+ sentence-level inverse IBM model 1	51.8	28.7	46.6	31.4
8	+ lexicalized reordering model	51.7	28.7	46.8	31.4
9	like 3., but single alignment (IBM model 4, ACL heuristic)	51.6	29.7	44.7	33.6
10	like 5., but monotonic translation	51.1	30.0	43.7	33.7
11	like 8., but no translation memory	52.2	**28.0**	**47.0**	**30.8**
12	like 11., but no restoration of contractions	50.4	29.6	46.3	32.5
	primary submission	**52.6**	29.3	45.7	32.9

mentation types (thus, our corpus for phrase extraction was effectively 15 times larger than the original corpus with 20K sentence pairs). Allowing the decoder to choose whether to detach particular prefixes and suffixes or not improved the BLEU scores significantly on both test sets (line 5 vs. line 4 in Table 5).

In all of the experiments described so far, we used the 4 run-based penalties described in Section 3.2 to select an optimal word order in translation process. As a reordering constraint, we set the maximum number of runs in an MT hypothesis to 3. To evaluate the power of our baseline reordering model, we performed a comparative experiment with monotonic translation (line 10 of Table 5). We see that the translation with reordering and run-based penalty features outperforms the monotonic baseline significantly on both test sets, e. g. by 1.8% absolute in TER on the IWSLT10 test set.

Next, we added the more complex POS-based and lexicalized reordering models described in Section 3.3. We also tested the sentence-level inverse IBM model 1. Again, the trend is different on IWSLT09 and IWSLT10. Overall, the combination of all three features yields the best results (line 8 vs. line 5 in Table 5). The measure that improves consistently is TER (e. g. by 0.5% absolute on the IWSLT10 test set). This measure is particularly sensitive to improvements in word order.

In lines 1 through 10 of Table 5 we used the "translation memory" as described in Section 3.6, replacing full sentence matches with translations from the training data. In total, 9% of the test sentences were replaced, mostly short ones like "That's right". Interestingly, when we omit this step, the results improve even further (line 11 vs. line 8 of Table 5). This indicates significant differences between the reference English translation of the same sentence in the training and in the test data. Another piece of evidence that automatic evaluation is not very reliable even with multiple references is the handling of contractions. When we fail to restore contractions, we change neither the meaning nor the structure of the sentence. Nevertheless, the MT measures degrade significantly (e. g., the BLEU score drops by 1.8% absolute on the IWSLT09 set, see line 12 of Table 5).

The result of our primary submission is given in the last line of Table 5. Unfortunately, it is worse than the best result we could achieve (lines 8/11). The reason for this is that for the final submission we re-optimized the system on 3 development sets: dev2, dev6, dev7. The idea of increasing the size of the dev set was to avoid local minima in the optimization process and improve the generalization capabilities of the system. However, in practice this only improved the BLEU score on the IWSLT09 test set, whereas both BLEU and TER degraded on the IWSLT10 set. This again shows that most of development and test sets for the IWSLT task are not similar to each other. Unfortunately, no reliable estimate of whether or not a certain new feature leads to translation quality improvement can be made on any of them.

5. Conclusions

In this paper, we described AppTek's new *APT* MT system. The system follows a statistical phrase-based approach, but utilizes comprehensive morphological analysis components to deal with rich morphology of the source languages involved – Arabic and Turkish. The biggest gains in translation quality as measured by automatic error measures are achieved by considering morphological segmentation alternatives in the MT search, and by employing novel reordering models.

The speech translation systems developed by AppTek for this year's IWSLT evaluation outperformed many of the competitor systems and at the same time had been designed to produce translations at a speed of 12 words per second or more – faster than the real time. This means that AppTek's

IWSLT systems can be seamlessly turned into a useful real-life product, such as speech translation in hand-held devices.

6. References

[1] S. Köprü and J. Miller, "A Unification Based Approach to the Morphological Analysis and Generation of Arabic," in *CAASL3: Proc. of the 3rd Workshop on Computational Approaches to Arabic-script based Languages*, Ottawa, Ontario, Canada, 2009.

[2] M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki, "The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus," in *Proceedings of the NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt, 2004, pp. 102–109.

[3] S. Köprü, "AppTek Turkish-English Machine Translation System Description for IWSLT 2009," in *Proc. of the International Workshop on Spoken Language Translation*, Tokyo, Japan, 2009, pp. 19–23.

[4] 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.

[5] R. Zens, "Phrase-based statistical machine translation: Models, search, training," Ph.D. dissertation, RWTH Aachen University, Aachen, Germany, February 2008.

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

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

[8] E. Matusov and S. Köprü, "Improving Reordering in Statistical Machine Translation from Farsi," in *AMTA 2010: The Ninth Conference of the Association for Machine Translation in the Americas*, Denver, Colorado, USA, November 2010.

[9] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in *6th Conference of the Association for Machine Translation in the Americas (AMTA 04)*, Washington DC, September/October 2004, pp. 115–124.

[10] E. Matusov, S. Kanthak, and H. Ney, "Efficient statistical machine translation with constrained reordering," in *Conference of the European Association for Machine Translation*, Budapest, Hungary, May 2005, pp. 181–188.

[11] C. Dyer, S. Muresan, and P. Resnik, "Generalizing word lattice translation," in *ACL-08: HLT*, Columbus, OH, USA, June 2008, pp. 1012–1020.

[12] E. Matusov, B. Hoffmeister, and H. Ney, "ASR word lattice translation with exhaustive reordering is possible," in *Interspeech*, Brisbane, Australia, September 2008, pp. 2342–2345.

[13] W. Shen, B. Delaney, A. R. Aminzadeh, T. Anderson, and R. Slyh, "The MIT-LL/AFRL IWSLT-2009 System," in *Proc. of the International Workshop on Spoken Language Translation*, Tokyo, Japan, 2009, pp. 71–78.

[14] 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.

[15] F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, S. Kumar, L. Shen, D. Smith, K. Eng, V. Jain, Z. Jin, and D. Radev, "A smorgasbord of features for statistical machine translation," in *Proceedings of HLT/NAACL 2004*, Boston, MA, May 2004.

[16] S. Hasan and H. Ney, "Comparison of extended lexicon models in search and rescoring for SMT," in *Proceedings of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies conference*, Boulder, Colorado, June 2009, pp. 17–20.

[17] 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*. Morristown, NJ, USA: Association for Computational Linguistics, 2002, pp. 311–318.

[18] M. Snover, B. Dorr, R. Schwartz, L. Micciula, and J. Makhoul, "A study of translation edit rate with targeted human evaluation," in *7th Conference of the Association for Machine Translation in the Americas (AMTA 06)*, Cambridge, Massachusetts, USA, August 2006, pp. 223–231.

[19] J. DeNero and D. Klein, "Tailoring Word Alignments to Syntactic Machine Translation," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 17–24.