Improving Example Based Machine Translation Through Morphological Generalization and Adaptation

Aaron B. Phillips
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
aphillips@cmu.edu

Violetta Cavalli-Sforza
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
violetta@cs.cmu.edu

Ralf D. Brown
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
ralf@cs.cmu.edu

Abstract

Example Based Machine Translation (EBMT) is limited by the quantity and scope of its training data. Even with a reasonably large corpus, we will not have examples that cover everything we want to translate. This problem is especially severe in Arabic due to its rich morphology. We demonstrate a novel method that exploits the regular nature of Arabic morphology to increase the quality and coverage of machine translation. Through the use of generalization and rewrite rules, we are able to recover the English translation of phrases that do not exist in the training corpora. Furthermore, this system shows improvement in BLEU even with a training corpus of 1.4 million sentence pairs.

Introduction

As the world grows more interconnected, the need for translating other languages into English becomes more pervasive. In particular, the current stage of world affairs has cast a spotlight on Arabic. As a result, in the last few years Arabic has become the major focus of many machine translation projects. This focus on Arabic has resulted in many rich resources available for the language. We now have a GigaWord Arabic corpus, several million sentence pairs of bilingual text, and a handful of morphological analyzers. However, the presence of these tools does not mean that we have solved the problem of integrating them together and building an effective translation platform.

One of the key features of the Arabic language, in sharp contrast to English, is its rich morphology. At the core of an Arabic word is a root -- a sequence of three consonants (more rarely four) -- that indicates a general concept or class of words. For example, the triconsonantal root k-t-b refers to writing and is used in the words that are translated in English as 'writing', 'book', 'letter', 'library', 'school', 'typewriter', and 'dictation'. Words are formed by combining the root with a different vowel pattern to form a stem.[1] Thus, *kAtib* (كاتب) is 'writer' and *kitAb* (كِتاب) is 'book'.[2] Analysis of these stems is complicated by the fact that most vowels are omitted in normal writing. Arabic words must also be conjugated, so in addition to the vowel pattern, affixes are attached to represent information such as number, person, or gender. On the other hand, English has very little morphology and one English word will often have several different representations in Arabic.

Data-driven machine translation is typically limited by the quantity and scope of its training data. This is problematic in Arabic due to its complex morphology. If we consider every combination of a stem and morphological affixes to be a separate word, the vocabulary of our training corpus will be quite large. A large vocabulary means that the number of occurrences of each word is quite low. Even with a reasonably large corpus, we will not have examples that cover everything we want to translate.

What we propose is a novel approach that exploits the regular nature of Arabic morphology to increase the quality and coverage of data-driven machine translation. Although the training data may lack an exact phrasal match, one can still derive the proper translation of many phrases by combining information from the corpus with an understanding of Arabic morphology. This is performed by looking at Arabic phrases as a whole, allowing inexact matching of morphological features, and adapting the English translation. For example, if we have a phrase where the noun and adjectives are marked for definiteness, we allow it to match the corresponding indefinite phrase (and possibly make a correction by inserting "the" into the English translation). Our method consists of three main parts: 1) generalization, 2) filtering and adaptation, and 3) rescoring.

Previous Work

Although our approach is new, there have been several attempts to apply an understanding of Arabic morphology to machine translation. (Nießen and Ney, 2000; Young-Suk Lee, 2004; Sadat and Habash, 2006; Zollmann et al., 2006) all present techniques that select a morphological analysis and split the source text in a manner that closely reflects the English translation. The latter three specifically address techniques for Arabic. (Sadat and Habash, 2006) provides additional insight by studying the effect of different morphological preprocessing decisions and the size of the training data. They conclude that an English-like segmentation scheme works well in Arabic on small data sets, but for large datasets minimal segmentation -- splitting only conjunction clitics and particles -- should be performed. Conceptually, the most similar work to ours is that of (Nießen and Ney, 2004) and (Yang and Kirchhoff, 2006). (Nießen and Ney, 2004) describes a statistical translation system that uses a generalized hierarchical lexicon with morphological features for German-English translation. (Yang and Kirchhoff, 2006) take this one step further and describe a statistical system that generalizes over the phrase table through a back-off model with examples in German-English and Finnish-English transla-

[1] Sometimes the term root and stem are used interchangeably. In this paper a root refers to only consonants and a stem is the result of applying a vowel pattern. Neither term includes affixes.
[2] All Romanized Arabic text follows the Buckwalter transliteration.

tion. However, this latter work only uses stemming and compound splitting. To the best of our knowledge, all published research that addresses morphology in Arabic machine translation has pursued segmentation. Furthermore, most of the approaches above also were only evaluated with 100,000 sentence pairs or fewer.

Our approach relies on the strength of Example Based Machine Translation and performs fuzzy matching of morphological features across entire phrases. Using morphologically similar phrases is especially helpful in language pairs such as Arabic-English where word order is different at a phrasal level. The morphological generalization technique is an extension of the method described in (Phillips and Cavalli-Sforza, 2006). It is similar to lemmatization except that it explicitly allows for ambiguous morphological analysis. In order to facilitate a transition to other languages, our system does not require a morphological analyzer that outputs a unique analysis. Ambiguity is explicitly permitted and resolved by the context of the phrasal match. Our approach is novel in that we go beyond simple phrasal backoff and include rewrite rules that dynamically account for any differences in morphological features. The rewrite rules adapt the English translation in order to more closely match the source text. Lastly, we demonstrate that our technique scales well with the size of training data and provides additional improvement even with a corpus of 1.4 million sentence pairs.

Panlite and the EBMT Engine

In our research we employ CMU's Example-Based MT (EBMT) engine (Brown, 2000a) which was developed as part of the The Pangloss-Lite (Panlite) MT system (Frederking and Brown, 1996). Given an input sentence, the EBMT engine retrieves lexical matches from an indexed corpus. These examples, as well as word-for-word translations from a bilingual lexicon, are stored in a lattice from which the final translation is extracted with the help of a language model using a search process equivalent to the decoder in a statistical MT system.

BAMA

In order to analyze the Arabic text, we use the Buckwalter Arabic Morphological Analyzer (BAMA). This analyzer identifies all possible combinations of stems and affixes for a word. For each analysis the stems and affixes are annotated with the morphological features they represent. Each stem is also associated with a lemmaID (which groups together stems with similar meanings) and an English gloss. All analyses are context insensitive and do not indicate their likelihood. BAMA returns these analyses as an XML document. Due to the nature of XML, analyzing a few megabytes of text results in hundreds of megabytes of output. To speed up processing, we modified BAMA to create a more compact output that could be read directly by our EBMT system.

Generalization

Our goal is to find an example in the training corpus that matches the Arabic source text regardless of morphological inflection. In order to achieve this goal we generalize every word in the corpus (retaining information about its original form). We replace the surface form of each word with a token that is the same across all morphological inflections.

Conceptually, we want to replace the surface form of a word with a token that corresponds to its meaning. Recall from the introduction that Arabic words are composed of a stem and affixes. The meaning of most Arabic words is contained within the stem; thus, a simple approach to this is to stem each word.[3] Using the analysis shown in Figure 1 of *wktAby*, we would remove the affixes *w* and *y*.

	Prefix		**Stem**		**Suffix**
Surface	*w*		*ktAb*		*y*
Voweled	*wa*	+	*kitAb*	+	*iy*
Gloss	'and'		'book'		'my'

Figure 1: One Analysis of *wktAby* (وكتابي)

As alluded to earlier, Arabic text is usually written without short vowels. This is why Figure 1 omits the bold letters in the surface form: *w**a**k**i**tAb**i**y*. In addition, some letters that look very similar are often mistakenly interchanged. For example, ي is sometimes written as ى (a separate letter). This results in a large amount of ambiguity when determining a proper morphological analysis. Indeed, the surface form *lnHl* (لنحل) has 47 different solutions according to the BAMA. A more realistic case would be our example from earlier, *wktAby,* which still has 13 different morphological analyses as shown in Figure 2. Not only is there ambiguity in determining the correct morphological analysis, but it is often difficult to determine the correct stem. Within the 13 analyses for *wktAby* there are still 3 possible stems with 4 different meanings.

There is also a problem with stem changes, such as in the common broken plurals, which are formed by modifying the stem rather than adding a plural affix. From our example we know that *kitAb* is book, but the plural form is *kutub*.

Generally, a human can determine the correct word by the meaning and context of a sentence. However, this is a difficult task for a computer. Though there are analyzers that look at the surrounding context and select the most likely analysis (Habash and Rambow, 2005), we would prefer to not make such decisions early on and potentially remove good candidates.[4] Instead, we preserve the ambiguity of the analysis and allow the system to select the best one at runtime. Furthermore, not requiring a more advanced analyzer makes the system's requirements fairly low and the transition to another language much easier.

Without a more sophisticated morphological analyzer, we cannot strip the affixes because we do not always know which part of the word is the stem. Even if we could identify the stem, that is not always good enough because sometimes the stem changes form. Thus we need a level

[3] If we wanted even more generalization, we could use the root of each word, but many words that share a root are only vaguely related. We tried a few experiments with this initially, but did not pursue the approach further.

[4] Preliminary experiments with early ambiguity removal showed worse performance.

lemmaID	Morphological Analysis							
kitAbiy~_1	*wa* ‘and’	+	*kitAbiy~* ‘writing/written’					
kitAbiy~_1	*wa* ‘and’	+	*kitAbiy~* ‘writing/written’	+	*u* [def.nom.]			
kitAbiy~_1	*wa* ‘and’	+	*kitAbiy~* ‘writing/written’	+	*a* [def.acc.]			
kitAbiy~_1	*wa* ‘and’	+	*kitAbiy~* ‘writing/written’	+	*i* [def.gen.]			
kitAbiy~_1	*wa* ‘and’	+	*kitAbiy~* ‘writing/written’	+	*N* [indef.nom.]			
kitAbiy~_1	*wa* ‘and’	+	*kitAbiy~* ‘writing/written’	+	*K* [indef.gen.]			
kitAb_1	*wa* ‘and’	+	*kitAb* ‘book’	+	*ayo* ‘two’ [acc.]			
kitAb_1	*wa* ‘and’	+	*kitAb* ‘book’	+	*ayo* ‘two’ [acc.]	+	*ya* ‘my’	
kitAb_1	*wa* ‘and’	+	*kitAb* ‘book’	+	*ayo* ‘two’ [gen.]			
kitAb_1	*wa* ‘and’	+	*kitAb* ‘book’	+	*ayo* ‘two’ [gen.]	+	*ya* ‘my’	
kitAb_1	*wa* ‘and’	+	*kitAb* ‘book’	+	*iy* ‘my’			
kut~Ab_1	*wa* ‘and’	+	*kut~Ab* ‘village school’’	+	*iy* ‘my’			
kAtib_1	*wa* ‘and’	+	*kut~Ab* ‘authors/writers’	+	*iy* ‘my’			

Figure 2: All Morphological Analyses of *wktAby* (وكتابي).

of abstraction higher than the stem. For every word in the lexicon, BAMA provides a lemmaID.[5] The lemmaID is a canonical form that represents a distinct semantic sense. For example, the lemmaID for *kitAb* ‘book’ and *kutub* ‘books’ is “kitAb_1”. Whereas using the lemmaID does reduce the ambiguity, often a single word can still be analyzed as having several different meanings (and thus different lemmaIDs).

Therefore, we abstract above the level of a single lemmaID by forming clusters that represent several lemmaIDs. We assign each lemmaID to be a member of one cluster. Our goal is to have all lemmaIDs that can be derived from an Arabic word exist in the same cluster. Then we will be able to tag each Arabic word with a token representing that cluster. Because the cluster represents several lemmaIDs, it maintains the ambiguity of the analyses. Every possible analysis of the Arabic word results in a lemmaID that is present in that cluster. However, this does not mean that every lemmaID present in the cluster is a valid analysis of the Arabic word. The clusters are not unique to each word; rather, they are shared by many words. Thus, there may be extra lemmaIDs present in a cluster (which will be filtered out at run time).

Theoretically, the number of lemmaIDs contained within each cluster is unimportant. However, in practice we do not want to have very large clusters. The number of lemmaIDs in each cluster is proportional to how many examples we must look for in our corpus at run time. Unfortunately, the morphological ambiguity in Arabic can be quite extreme compared to western languages. Thus, our goal of having all lemmaIDs that can be derived from an Arabic word exist in the same cluster is not feasible because some clusters are simply too large.[6] For efficiency, we relax this restriction and form clusters of lemmaIDs such that most analyses of an Arabic word result in lemmaIDs that occur together in one cluster. At runtime the system is limited to looking up matches using the lemmaIDs contained in one cluster. If two lemmaIDs are possible analyses of an Arabic word and we have failed to place them in the same cluster, then we will only be able to look for matches using one of the lemmaIDs. It will not make the system worse than if no generalization was present, but the system will not perform at its full potential.

In order to build the clusters described above, we rephrase the problem as a graph clustering problem. First we plot all the lemmaIDs on a graph. Then we analyze every Arabic word in BAMA's lexicon. For each word we build a fully connected graph of the set of lemmaIDs that are possible analyses of the word. If a connection between two lemmaIDs already exists, then we increment the weight of the connection. See the example in Figure 3. To further ensure accuracy, we also adjust the weighted graph by the unigram probability of each lemmaID as calculated from

[5] Our system is not dependent on BAMA, but if stems can change form (as they do in Arabic) then the analyzer must provide some canonical form for the stem.

[6] Our initial system did exactly that and was very, very, slow. In particular, there was one cluster that contained over 2,000 lemmaIDs and occurred frequently. This large cluster was the result of some two letter Arabic roots and corresponding lemmaIDs being possible, but unlikely, analyses of a wide range of words.

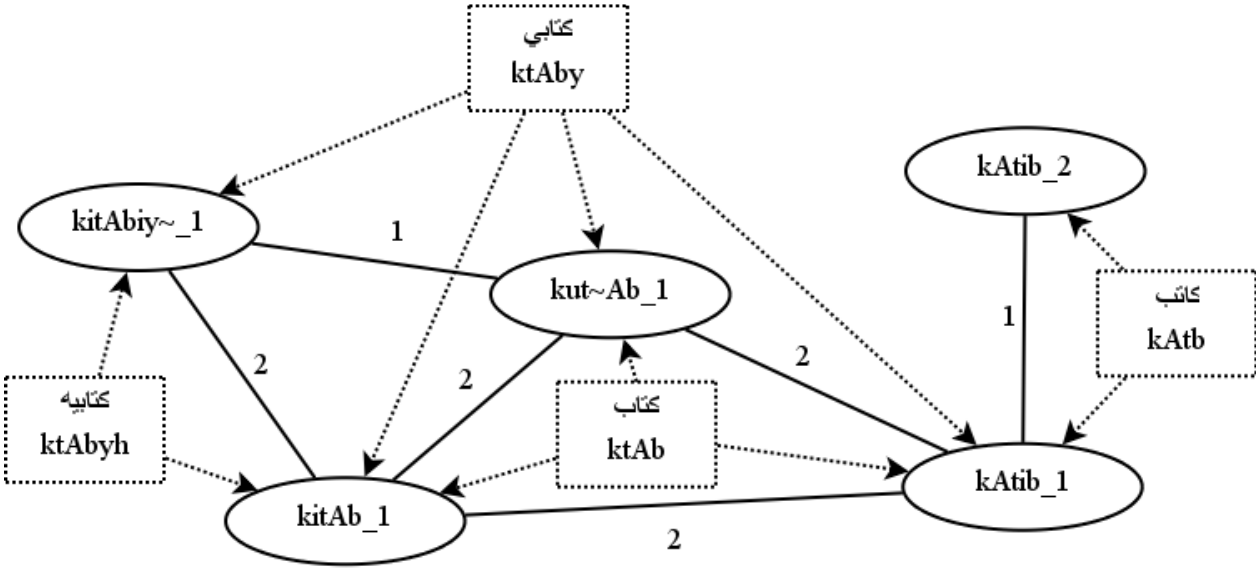


Figure 3: Example of Clustering. Each ellipse represents a lemmaID. The dotted boxes and lines are not part of the graph. Rather, they are provided to illustrate to the reader the Arabic words that result in this graph structure.

the LDC Arabic Treebank. The actual clustering is done using a technique developed by (van Dongen, 2000) and the freely available MCL toolkit.[7] This algorithm randomly walks through the graph to determine areas of high connectivity. Based on parameters given to MCL, we are able to adjust the required amount of interconnectivity and thus the size of the clusters.

These clusters are then used to transform the text so that we can look up all morphological forms of a word with one token. Each word in the text is replaced by a token -- the name of the cluster to which it belongs. In addition, we annotate the text with information about the original form of the word and its possible morphological features from BAMA. This is important because we use the information later to identify the best translations. This process is external to the EBMT system so that a different morphological analyzer could be used and so that it could be easily applied to a different language. We perform this process on both the training text and the evaluation text.

Filtering & Adaptation

At runtime the EBMT engine looks for examples that will yield the correct English translation. This is done by searching for Arabic matches based on the clustering described above. When we retrieve an example, we know it is only a possible match. The clusters are often broad and over-generalize. We need to determine which examples in our training data are true morphological generalizations and which ones are noise created by the clustering. Additionally, we may have to adapt the translation of some of the examples we find in the corpus if the morphological features differ.

We iterate through each possible morphological analysis of the text to be translated and all matches from the corpus. For each pair of analyses we determine how similar they are. If the surface forms are equal, then we have a perfect match. (This would be the same as if we were running the system without morphology.) If the surface forms are not equal, then we determine if we have a valid generalization by comparing lemmaIDs. If any word in the example does not share a possible lemmaID with the corresponding word in the text to be translated, the example is discarded. Additionally, it should be pointed out that we are only looking at phrases (2 or more words), so there is also some context that must be the same.

If there are matching lemmaIDs in both the source text and the example, then we compare the morphological features. To be a valid translation candidate, some morphological features such as part of speech and person must be the same. It is necessary that these features do not change in order to recover a proper English translation.

Other features such as gender and case are allowed to be different. Most of the time these features, while different in Arabic, have the same realization in English. Even if the realization is occasionally different, it results in an acceptable English translation. The EBMT system is probabilistic and in the end a language modeler will select the most likely combination of phrases. It is usually better to get a long phrasal match that is mostly correct than resort to word by word translation.

Additionally, we allow matches with morphological features that alter the English translation, if the change is easily defined. When this occurs, the system dynamically alters the English translation via a rewrite rule to match the change in morphology. For example, one of the most common adjustments made is to account for the prolific use of initial *wa* in Arabic. The prefix *wa* is identified by BAMA as a conjunction clitic, and our system marks the phrase with the morphological feature CONJ. Sometimes *wa* is translated as ‘and’, but frequently it is extraneous and it is dropped from the English translation. In its un-

[7] Available at http://micans.org/mcl/.

voweled form, *wa* appears simply as *w*. Although we may not have seen the phrase *wAlktAb Alqdym* (والكتاب القديم) we may have seen *AlktAb Alqdym* (الكتاب القديم) translated as 'the old book'. In this case we do not know if 'and' (due to the presence of *w*) should appear in the English translation. Thus we put both 'the old book' and 'and the old book' into the lattice and rely on the language modeler to select the correct phrase in context. Similarly, the system will add or remove words from the English translation to account for the presence or absence of prepositions and definite markers in the Arabic text.

Allowing generalization to take place over morphological features that alter the English translation and then recovering a valid English translation through rewrite rules significantly enhances our coverage. Although there are a handful of morphological features that do not usually change the English translation (the ones we do not modify as explained previously), they do not occur with as much frequency. It is very common to find two translations that are subtly different -- where the only difference is that one uses a different preposition or is marked as definite and the other is not. Fortunately, these changes can easily be captured and adjusted for with rewrite rules. The rewrite rules allow the largely statistical system to tap into human knowledge of how morphological features change the translation. They allow our system to increase coverage and generate examples that do not exist in the corpus.

Not Allowed	Allowed without Modifications	Allowed with Modifications
Part of Speech Aspect Voice	Gender (not nouns) Case (only acc. and nom.) Mood	Definiteness Negativity Poss. Pronouns Nom. Pronouns Conjunctions Prepositions *Person *Number *Acc. Pronouns

*Currently Not Implemented

Figure 4: Types of Morphological Generalizations

Figure 4 shows the features over which the current system is able to generalize. As noted in the diagram there are still some features that are slightly more complex for which we have not yet had time to write the code. Number, for example, is fairly simple in the singular and plural, but Arabic has a dual number that might appear as a plural or some permutation of 'both' or 'two of' in the English text.

Scoring

The EBMT engine returns a heuristic score for each example that roughly represents its alignment quality. This score is then modified by a weight corresponding to how much generalization was required for the example to match. A perfect match obviously needs to receive more weight than a heavily generalized match. The scores across all examples that resulted in the same target text are summed together and then divided by the total of all examples to produce a probability for each translation.

Proper scoring of the generalizations plays a critical role in the system. Initially, the system had a predefined weight for each morphological feature and the scoring was multiplicative. Thus, a morphological generalization that had a change in definiteness and gender would be the product of the weights for definiteness and gender. However, the performance of this system was suboptimal.

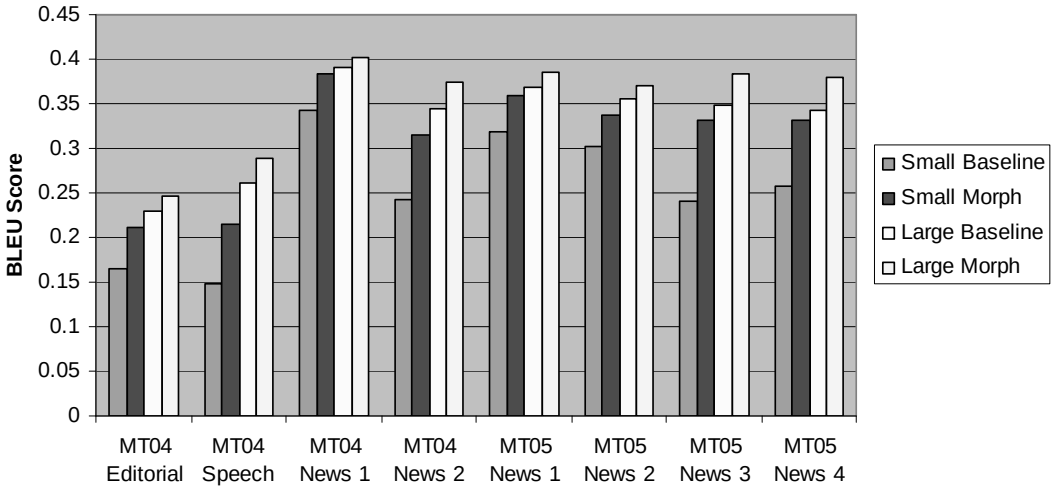
We ran the system over parts of the 2003 NIST MT Evaluation (MT03) data and compared the generalized matches to four reference translations. An example was marked as correct if its translation was found in the reference translations and incorrect if it was not present in any of the reference translations. Short matches (those containing a single word or whose total length was less than 10 characters) were excluded from analysis. Then we ran tests to see how often each generalization was correct and the results were surprising. Figure 5 is a partial summary of the morphological generalizations performed on trigram examples retrieved from the corpus. The percentage next to each line indicates how often the generalization was correct. Each set of parenthesis indicates the morphological generalizations (if any) performed at that position in the phrase. Figure 5 illustrates that removing a preposition in the first word of a trigram is much more likely to be correct than removing a preposition later in the phrase. Thus, we needed to score differently based on the location of the generalization. Furthermore, the score for a combined value does not appear to be related to its individual scores. Inserting or removing a definite article at the beginning of a trigram has approximately the same probability. However, inserting a definite article and removing a preposition at the beginning of a trigram is much more likely to result in a correct translation than removing a definite article and removing a preposition at the same location.

11% () () (Preposition Removal)
13% () (Preposition Removal) ()
81% (Preposition Removal) () ()
34% (Definite Insertion) () ()
31% (Definite Removal) () ()
48% (Preposition Removal + Definite Insertion) () ()
23% (Preposition Removal + Definite Removal) () ()

Figure 5: Percent Correct by Generalization.

This demonstrated that the scoring needs to be based off some training data and have different probabilities for each possible combination of morphological features. The probabilities will not be exact, and by the nature of our tests looking for exact string matches, they are likely to be somewhat low. However, the final score for a translation usually depends on many different generalizations, so an inexact weight for one generalization will not greatly affect the overall scores.

The final score for each translation is an interpolation of its score for each type of generalization. The interpolation weights are what we determine from training data, with non-generalized examples given a weight of 1.0. Conceptually, we can think of the system as having different cor-



	MT04 Editorial	MT04 Speech	MT04 News 1	MT04 News 2	MT05 News 1	MT05 News 2	MT05 News 3	MT05 News 4
Small Baseline	0.1640	0.1480	0.3431	0.2429	0.3191	0.3014	0.2409	0.2574
Small Morph	0.2114	0.2153	0.3841	0.3153	0.3585	0.3363	0.3315	0.3306
Small Improv.	28.90%	45.47%	11.95%	29.81%	12.35%	11.58%	37.61%	28.44%
Large Baseline	0.2293	0.2604	0.3911	0.3436	0.3680	0.3550	0.3482	0.3434
Large Morph	0.2466	0.2897	0.4027	0.3741	0.3856	0.3696	0.3836	0.3803
Large Improv.	7.54%	11.25%	2.97%	8.88%	4.78%	4.11%	10.17%	10.75%

Figure 6: System Performance in BLEU with and without Morphological Generalization.

pora for each type of generalization and combining together the examples it finds from each corpora.

Results

For the evaluation, we built one system with a small training corpus and another system with a large training corpus. The large dataset consisted of newswire text and sections of the UN Arabic-English Parallel Text (LDC2004E13).[8] The small dataset excluded the UN corpus and half of the newswire text. The large dataset was approximately 1.4 million sentences pairs while the small dataset contained 50,000 sentence pairs. When data is scarce, we expected the morphological processing to bring large gains. However, we wanted to see if our system could still contribute when very large amounts of training data were available.

Each system was evaluated on the 2004 and 2005 NIST MT Evaluation data sets (MT04 and MT05). Each of these datasets contains four human reference translations. MT04 contains editorial, speech, and news genres, but nearly half of it is news. We desired to evaluate our system several times to determine when morphological generalization is beneficial. As such, we split MT04 by genre, but also divided the news into two parts -- one from Xinhua News Agency and the other from Agence France Press. MT05 contains only news text also from Xinhua News Agency and Agence France Press; we split this data set into four chunks that were approximately the same size as the MT04 chunks. Document boundaries were preserved in all the splits and the chunks range in size from 251 sentences to 387 sentences. Splitting the data in this fashion allowed us to perform multiple evaluations while maintaining enough sentences to have meaningful results.

Parameters controlling aspects of the system such as the number of translation candidates, length ratio, reorder penalty, language model weighting, and more were tuned on the MT03 dataset. The tuning process evaluated 26 random starting points and then maximized the best starting point through hill-climbing. To help ensure that a local maximum was not found inadvertently, the procedure was done twice on each data set. The parameters were tuned for the baseline system which does not include the morphological generalizations. Weights for the morphological generalizations were determined separately from the MT03 dataset. However, the system that included the morphological generalizations used the same tuned parameters as the baseline system.

The results of our evaluation are shown in Figure 6. In both the small and large dataset we see a healthy improvement over the baseline that is statistically significant. That the relative improvement for the small system is higher than the large system is to be expected. In the large data scenario, the morphological framework is not needed as frequently as the corpus contains more of the phrases we are trying to translate. However, the important part is that the morphological framework does still im-

[8] Newswire includes AFA, AFP, ANN, ASB, eTIRR, Ummah 2006, and Xinhua.

prove the baseline between 3-11%. This proves our earlier conjecture that due to the nature of Arabic morphology, even with a large corpus, there will be many phrases we have not seen before.

It is interesting to point out that the speech genre showed the largest improvement using both the small and large corpora. The morphological system also significantly helped out the editorial genre, but by not as much. These two data points alone are not enough to draw strong conclusions, but they do suggest that our method is particularly useful when the genre or topic differs from the training data.

From the news text we can see that the morphological processing in particular significantly boosts underperforming sections. Fluctuations across all the news results are likely due to the different news sources and document topics. Within the news genre the morphological generalization improved the mean and decreased the variance.

The most encouraging result is that in these tests the morphological processing never resulted in a lower score than the baseline. Thus, we can safely apply this method even when training on very large datasets. Furthermore, we evaluated both systems using parameters tuned for the baseline. We expect even greater gains if we tune the parameters for the system that uses morphological generalizations.

Conclusion

In conclusion, we have described a system that improves the quality of translation by generalizing over Arabic morphological features at the phrasal level. When the morphological features differ, if necessary, the system automatically alters the English translation through rewrite rules. As a result of this work, the EBMT system is now able to effectively translate Arabic phrases it has never seen before based on morphologically similar phrases. This, in effect, extends the coverage of the training corpus, which also increases accuracy of translation. Moreover, this system improves BLEU scores even when trained on 1.4 million sentence pairs. Our strategy is more effective than current approaches because morphemes are not split off, allowing us to match morphological changes anywhere in a phrase. We also appropriately handle the issue of morphological ambiguity and include rewrite rules that allow for a wider range of generalization. While this work focused only on Arabic, the rewrite rules are the only language-specific component of the system. Minimal work would be required to apply our system to another language, and this is an area we hope to explore in the future.

References

Brown, Ralf D. (1996). Example-Based Machine Translation in the Pangloss System. In Proceedings of the 16th International Conference on Computational Linguistics (pp. 169--174).

Brown, Ralf D. (1999). Adding Linguistic Knowledge to a Lexical Example-Based Translation System. In Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation (pp. 22--32).

Brown, Ralf D. (2000a). Example-Based Machine Translation at Carnegie Mellon University. In The ELRA Newsletter, European Language Resources Association, 5(1).

Brown, Ralf D. (2000b). Automated Generalization of Translation Examples. In Proceedings of the Eighteenth International Conference on Computational Linguistics (pp. 125--131).

Buckwalter, Tim. (2004). *Buckwalter Arabic Morphological Analyzer Version 2.0*. Linguistic Data Consortium, Catalog Number LDC2004L02.

Frederking, R.E. and Brown, R.D. (1996). The Pangloss-Lite Machine Translation System. In Expanding MT Horizons: Proceedings of the Second Conference of the Association for Machine Translation in the Americas (pp. 268--272).

Habash, Nizar and Owen Rambow. (2005). Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop. In Proceedings of the Conference of American Association for Computational Linguistics.

Lee, Young-Suk. (2004). Morphological analysis for statistical machine translation. In Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference.

Nießen, Sonja and Ney, Hermann. (2000). Improving SMT Quality with Morpho-Syntactic Analysis. In The 18th International Conference on Computational Linguistics.

Nießen, Sonja and Ney, Hermann. (2004). Statistical Machine Translation with Scarce Resources using Mopho-Syntactic Information. Comput. Linguist, 30(2), 181--204.

Phillips, Aaron B. and Cavalli-Sforza, Violetta. (2006). Arabic-to-English Example Based Machine Translation Using Context-Insensitive Morphological Analysis. In Journées d'Etudes sur le Traitement Automatique de la Langue Arabe.

Sadat, Fatiha and Habash, Nizar. (2006). Arabic Preprocessing Schemes for Statistical Machine Translation. In Proceedings of Human Language Technology Conference of the NAACL.

van Dongen, Stign. (2000). Graph Clustering by Flow Simulation. Ph.D. Thesis. University of Utrecht.

Yang, Mei and Kirchoff, Katrin. 2006. Phrase-Based Backoff Models for Machine Translation of Highly Inflected Languages. In Proceedings of the European Chapter of the ACL.

Zollmann, Andreas, Venugopal, Ashish and Vogel, Stephan. (2006). Bridging the Inflection Morphology Gap for Arabic Statistical Machine Translation. In Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers (pp. 201--204).