Aaron B. Phillips[1], Violetta Cavalli-Sforza[2]
[1]Carnegie Mellon University- USA, aphillips@cmu.edu
[2]Carnegie Mellon University- USA, violetta@cs.cmu.edu

Arabic-to-English Example Based Machine Translation Using Context-Insensitive Morphological Analysis

Example Based Machine Translation (EBMT) is a form of automated translation that uses a large corpus of previously-translated example sentences to create a translation for a new sentence. Typically the system does not have in its corpus the entire sentence to be translated. Instead, the system matches words and small phrases and stitches them together with the help of a target language model. Although EBMT uses statistical methods, it differs from Statistical Machine Translation (SMT). EBMT consults its corpus of translations at runtime, whereas SMT pre-processes the corpus to calculate the probability of a word or phrase occurring as translations, but then uses only these probabilities at runtime.

EBMT, like SMT, is limited to working with the data that occur in their corpus. The fact that many words and phrases occur with low frequencies poses a significant problem because even a large training corpus will not have examples that cover everything we want to translate. Furthermore, the accuracy of a translation is significantly enhanced when many examples of the same text are found in the corpus. It is, therefore, highly desirable to form generalizations to increase the coverage of our examples and capture things that are not directly seen in the text.

Arabic, a highly inflectional language, is particularly susceptible to data sparseness, but also lends itself well to generalization. A root (usually a series of three or four consonants) combines with a vowelling pattern to form a stem. Affixes representing information such as case, number, and gender are added to stems in order to form words. Thus, while it is unlikely that we will see in our training data all forms of an Arabic word, if we know the rules of Arabic morphology we can predict how unseen Arabic words would act.

In this research, we exploit the regular nature of Arabic morphology in order to generalize over text that we have seen and find translations of unseen text. Instead of building a corpus that uses the final form of Arabic words, we try to enhance coverage by using a more general representation of each word, which, in its simplest form, is its stem (e.g., نكتب "we write", stem: كتب). This is often safe because, in translating from Arabic to English, the same English word covers several inflected variants of the Arabic word. However, in many instances, this technique will overgeneralize and produce invalid translations. Thus, we also need to keep track of information from the original surface form that was replaced by the generalization, and pass it at along at runtime as "meta data" to select the best possible translation. In the absence of an exact translation, a generalized translation will still be largely correct and is superior to no translation.

There are two components to this research: how the words are stemmed/generalized, and what information about the original word is used to filter translations, which we will address in turn.

I. Generalization

For these experiments we used the Buckwalter Arabic Morphological Analyzer (BAMA), a context-insensitive morphological analyzer that returns all possible compositions of stems and affixes for a word. 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. Unfortunately, the missing diacritic markings of written Arabic text and other orthographical variations and errors give rise to ambiguity, so that a word can be analyzed as having originated from multiple stems with different affix segmentations. This increases the number of analyses we must look at and raises the issue of how to select the correct stem for a word.

If we had a large corpus of hand-analyzed Arabic text, we could choose the stem based on its frequency of occurrence. Lacking that, the only information we can use from BAMA is which stem gives the most analyses, which does not guarantee it to be the most frequent stem in natural text. Although less than ideal, using this initial approach to generalization gave us a reasonable boost in coverage, showing a 5% increase in the number of words covered by a phrase of 4 words or longer.

A slightly more sophisticated means of generalizing is to use the lemmaID in BAMA. The lemmaID is a rough indication of the sense of the word and covers a fairly small group of words that share a similar stem. Generalizing words by their lemmaID performed better than just taking the most frequent stem in the BAMA

analysis, but further increased coverage of phrases of 4 words or longer by only 2%. LemmaIDs are more general than just the stemmed word because they encapsulate multiple stems, but lemmaID classes are still relatively small and a given surface form can still be part of multiple lemmaID classes, and so even the lemmaID is not general enough.

To achieve greater generalization, we also experimented with clustering the stems in such a way as to have each stem map to a single generalized token and all possible stems for any given word to map to the same generalized token. In this way we are not losing information, and we keep possible analyses around for disambiguation at runtime. This technique showed a further 10% increase in coverage of phrases 4 words or longer. Nearly 40% of unseen text is part of phrases four words or longer that are found in our generalized corpus. Approximately 80% of unseen text is covered by trigram matches or better. These are substantial improvements over the 50% trigram matches and 17% four-gram matches we saw in our original text with no generalization.

II. Filtering

Having generalized to improve coverage of unseen text, we now must deal with reducing the ambiguity introduced through overgeneralization.

We initially postulated that, if we generalized by stemming the words, we could select the closest match by comparing the morphological features of the text to be translated and each example in our corpus. The multiple analyses produced by BAMA make this difficult, since we do not have a morphologically annotated corpus on which to compute the frequency of occurrence of different features. We could also keep all the possible morphological features and see what percentage of them match, but neither of these context-insensitive approaches works very well.

What does work well is to save the original surface form of the word along with all possible stems of this surface form. If two words share the same surface form, then we mostly have an exact translation and we prefer these matches over generalized ones, guaranteeing that we will have all the same matches that would occur if the text did not undergo any generalization. However, if no surface form matches exist, then we select the example(s) in the corpus that share a possible stem with the text we are attempting to translate. Recall that we are clustering the stems into groups that have some possible analysis in common. We know that every possible stem of a word will be in the same cluster, but that does not mean that the possible analyses of two different words are the same. Likely, the two words will only have one or two stems in common. By comparing the possible stems, we effectively reduce the large clusters we built to smaller classes that represent words that are truly ambiguous and could have the same stem as the word we are looking at, decreasing the overgeneralization by filtering out matches that do not share any analyses in common.

Conclusion & Current Work

As a result of generalization and filtering, the EBMT system finds more and longer examples that match the text we are attempting to translate. Before generalization, we could often translate common phrases, but many parts of sentences had to be translated word by word. As expected from the generalization, some of the matched examples have incorrect morphological features, such as the wrong tenses or number, but this is not problematic in translating from Arabic to English. Analysis of matched examples shows that the generalization preserves all the examples from the ungeneralized text and frequently includes many more examples of substantial quality.

The problem we are currently facing is that the language modeler (LM) is not good at stitching the generalized phrases back together. If we hand-stitch them, we can create much higher quality sentences than we can from the ungeneralized examples found in the corpus but, if an example has an incorrect tense or is the wrong part of speech, the LM will assign it a very low score because such combinations do not occur in natural text. Ongoing experiments are addressing this problem.

References

1. Ralf D. Brown, "Example-Based Machine Translation in the Pangloss System". In *Proceedings of the 16th International Conference on Computational Linguistics* (COLING-96), p. 169-174. Copenhagen, Denmark, August 5-9, 1996.

2. Ralf D. Brown. "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* (TMI-99), p. 22-32. Chester, UK [photos], August 1999.

3. Ralf Brown. "Example-Based Machine Translation at Carnegie Mellon University". In *The ELRA Newsletter*, European Language Resources Association, vol 5:1, January-March 2000.

4. Ralf D. Brown. "Automated Generalization of Translation Examples". In Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING-2000), p. 125-131. Saarbrücken, Germany [photos], August 2000.

5. Tim Buckwalter. Buckwalter. Arabic Morphological Analyzer Version 2.0. LDC2004L02 http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004L02)