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

Aaron B. Phillips *aphillips@cmu.edu*

Violetta Cavalli-Sforza *violetta@cs.cmu.edu*

Language Technologies Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15217, U.S.A.

Abstract: We describe and discuss the results of ongoing experiments that use morphological analysis in the context of Example-Based Machine Translation. The goal is to increase the coverage of our training examples so as to capture things that are not directly seen in the training text. This is done through a two stage process of generalization and filtering.

Keywords: Arabic morphology, generalization, example-based machine translation

1. Introduction

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 and uses only these probabilities at runtime.

EBMT, like SMT, is limited to working with the data that occurs in its 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. If a portion of the input to be translated can be transformed into a more general class whose membership includes portions of the training corpus, many more matches can be retrieved at translation time. It is, therefore, highly desirable to form generalizations to increase the coverage of our training 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 or semi-vowels (weak consonants) – combines with a vowelling pattern to form a stem. Affixes representing information such as person, number, gender and case 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 a context that demands specific inflections.

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, such as 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 over-generalize and produce invalid translations.[1] 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 "metadata" to select the best possible translation. In the absence of an exact translation, a generalized translation will still, hopefully, be largely correct and is superior to no translation.

In this paper, we begin by briefly describing the EBMT engine and the multi-engine MT system in which it is used. Then we address, in turn, the two major components of this research – how the words are stemmed/generalized, and what information about the original word is used to filter translations – and provide some preliminary results. We conclude by discussing some of the issues that are still facing us and our current directions of work.

2. The Panlite Multi-Engine MT System and the EBMT Engine

Several EBMT approaches have been developed over the last several years (Carl and Way, 2003; Somers, 1999), some of which explicitly consider morphology. In our research we employ the Pangloss-Lite (Panlite) MT system (Frederking and Brown, 1996), which was used as the translation engine in DIPLOMAT (Frederking, Rudnicky and Hogan, 1997), a rapid-deployment speech-to-speech MT project.

Panlite is a hybrid, multi-engine MT system, with a strong empirically-based (example-based) core (Brown, 2000a). Given a sentence to translate, each engine provides scored translation(s) for either the full sentence or fragments of the sentence. Translation candidates are placed in a chart as 'edges' covering the input or some part of it. One component of the system, the Language Modeler (LM) decoder, uses statistical knowledge of the target language and heuristics to select or piece together from the chart the best scoring translation(s) that cover the entire input. The Panlite system supports the integration of widely different MT engines, but provides three built-in engines in addition to the LM: an Example-Based MT (EBMT) Engine, a Glossary Engine, and a Dictionary Engine.

The EBMT Engine, at its simplest, translates by matching new input to previously seen examples of translation pairs. The essential 'training' data for the EBMT Engine is a parallel corpus of translation pairs (phrases or sentences). The system does not 'learn' in the traditional machine learning sense. Its training consists of preprocessing the parallel data and building an index so as to make retrieval of any part of the source text and corresponding part of the target text as fast as possible when the system is translating new input. The preprocessing also includes determining the correspondence between fragments of parallel source and target sentences. This computation could be performed at translation time, but it is more efficient to perform it at indexing time and to store its results in the index.

At runtime (translation time), the input sentence to be translated and its fragments are matched against the source-side of the indexed training corpus, with some flexibility in determining what is considered to be a good match and some control over the extent of the search for candidate matches. Candidate translations produced from the target side of the indexed corpus are scored and posted with their score to the chart, which stores the translations provided by all engines. If the EBMT Engine cannot find a match for the entire input sentence, it tries to match all possible multi-word input fragments and posts to the chart

[1] Over-generalization also occurs, among other reasons, because Arabic text normally lacks diacritics, which distinguish between different parts of speech and different derivational forms (e.g. different measures of a verb), so that the same undiacriticized form can actually map to substantially different meanings. In an example-based approach to machine translation, which employs little if any syntactic or semantic knowledge, there is little recourse for this problem, except for relying on context to select example matches that carry the appropriate part of speech and meaning for the input we are attempting to match.

what it believes to be the corresponding translations.

At times, pieces of the input to be translated cannot be matched against any previously seen source sentences, so there will be holes in the translations produced by the EBMT system and it is useful to back off to the Dictionary Engine to obtain translations for single words on the source side. The dictionary used by the Dictionary Engine may be statistical (automatically computed from the parallel corpus) or manually refined or developed. Experience with our EBMT system shows that a statistical dictionary usually works best for alignment, but a manually refined dictionary provides better word-for-word translations at runtime. Both can be used in the Panlite MT system. Finally, a Glossary Engine can supplement the translations provided by EBMT with perfectly aligned human-supplied translations for words and phrases.

The EBMT Engine provides generalization capabilities above and beyond the work described in this paper (Brown, 1999; Brown, 2000b). One can define classes of words or phrases in the source language and corresponding translation in the target language that are semantically similar and syntactically interchangeable. These equivalence classes allow the training examples to work in a broader range of situations, since unseen input may be matched to more examples if fragments generalize to the same classes.

3. Major Research Components: Generalization and Filtering

In order to increase coverage, we first perform generalization and then filter our results to limit over-generalization. The generalization that we perform in these experiments currently does not use the generalization framework described above. Rather, the generalization is done simply by preprocessing the input of the EBMT system. We use the output of a morphological analyzer to reduce the Arabic vocabulary (by selecting the stem, lemmaID, or a cluster) and then attach metadata describing the original form of the word. This metadata is then used at runtime by the filtering process.

3.1. Generalization

For our work 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. Table 1 below gives an example of all the various morphological parses for the word *wkAlp* (وكالة) "agency".

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, though it is not guaranteed 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. For example, the analysis of the word *wkAlp* as shown above has three possible stems: *wkAl*, *kAl*, and *Al*. Of these possible stems, *wkAl* is used twenty one times, *kAl* is used seven times, and *Al* is only used three times. Thus, using the method described above we select *wkAl* as the canonical stem for the word *wkAlp*.

Stem	*LemmaID*	*Analysis*
wkAl	*wikAlap_1*	*wikAl/NOUN ap/NSUFF_FEM_SG*
wkAl	*wikAlap_1*	*wikAl/NOUN ap/NSUFF_FEM_SG+u/CASE_DEF_NOM*
wkAl	*wikAlap_1*	*wikAl/NOUN ap/NSUFF_FEM_SG+a/CASE_DEF_ACC*
wkAl	*wikAlap_1*	*wikAl/NOUN ap/NSUFF_FEM_SG+i/CASE_DEF_GEN*
wkAl	*wikAlap_1*	*wikAl/NOUN ap/NSUFF_FEM_SG+N/CASE_INDEF_NOM*
wkAl	*wikAlap_1*	*wikAl/NOUN ap/NSUFF_FEM_SG+F/CASE_INDEF_ACC*
wkAl	*wikAlap_1*	*wikAl/NOUN ap/NSUFF_FEM_SG+K/CASE_INDEF_GEN*
wkAl	*wikAlap_1*	*wakAl/NOUN ap/NSUFF_FEM_SG*
wkAl	*wikAlap_1*	*wakAl/NOUN ap/NSUFF_FEM_SG+u/CASE_DEF_NOM*
wkAl	*wikAlap_1*	*wakAl/NOUN ap/NSUFF_FEM_SG+a/CASE_DEF_ACC*
wkAl	*wikAlap_1*	*wakAl/NOUN ap/NSUFF_FEM_SG+i/CASE_DEF_GEN*
wkAl	*wikAlap_1*	*wakAl/NOUN ap/NSUFF_FEM_SG+N/CASE_INDEF_NOM*
wkAl	*wikAlap_1*	*wakAl/NOUN ap/NSUFF_FEM_SG+F/CASE_INDEF_ACC*
wkAl	*wikAlap_1*	*wakAl/NOUN ap/NSUFF_FEM_SG+K/CASE_INDEF_GEN*
wkAl	*wikAlap_2*	*wikAl/NOUN ap/NSUFF_FEM_SG*
wkAl	*wikAlap_2*	*wikAl/NOUN ap/NSUFF_FEM_SG+u/CASE_DEF_NOM*
wkAl	*wikAlap_2*	*wikAl/NOUN ap/NSUFF_FEM_SG+a/CASE_DEF_ACC*
wkAl	*wikAlap_2*	*wikAl/NOUN ap/NSUFF_FEM_SG+i/CASE_DEF_GEN*
wkAl	*wikAlap_2*	*wikAl/NOUN ap/NSUFF_FEM_SG+N/CASE_INDEF_NOM*
wkAl	*wikAlap_2*	*wikAl/NOUN ap/NSUFF_FEM_SG+F/CASE_INDEF_ACC*
wkAl	*wikAlap_2*	*wikAl/NOUN ap/NSUFF_FEM_SG+K/CASE_INDEF_GEN*
kAl	*kAl~_1*	*wa/CONJ kAl~/ADJ ap/NSUFF_FEM_SG*
kAl	*kAl~_1*	*wa/CONJ kAl~/ADJ ap/NSUFF_FEM_SG+u/CASE_DEF_NOM*
kAl	*kAl~_1*	*wa/CONJ kAl~/ADJ ap/NSUFF_FEM_SG+a/CASE_DEF_ACC*
kAl	*kAl~_1*	*wa/CONJ kAl~/ADJ ap/NSUFF_FEM_SG+i/CASE_DEF_GEN*
kAl	*kAl~_1*	*wa/CONJ kAl~/ADJ ap/NSUFF_FEM_SG+N/CASE_INDEF_NOM*
kAl	*kAl~_1*	*wa/CONJ kAl~/ADJ ap/NSUFF_FEM_SG+F/CASE_INDEF_ACC*
kAl	*kAl~_1*	*wa/CONJ kAl~/ADJ ap/NSUFF_FEM_SG+K/CASE_INDEF_GEN*
Al	*|lap_1*	*wa/CONJ+ka/PREP |l/NOUN ap/NSUFF_FEM_SG*
Al	*|lap_1*	*wa/CONJ+ka/PREP |l/NOUN ap/NSUFF_FEM_SG+i/CASE_DEF_GEN*
Al	*|lap_1*	*wa/CONJ+ka/PREP |l/NOUN ap/NSUFF_FEM_SG+K/CASE_INDEF_GEN*

Table 1. Possible Analyses of the word *wkAlp* (وكالة)

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 typically more general than just the stemmed word because they can encapsulate multiple stems. However, lemmaID classes are still relatively small and a given surface form can be part of multiple lemmaID classes, so even the lemmaID is not general enough. Looking at the word *wkAlp* again, we would select the lemmaID "wikAlap_1" because it is used the most frequently. This lemmaID spans the stems *wikAl* and *wakAl*. (In this case, once you remove the vowels the stems are the same.) Table 2 below provides as an example a few lemmaIDs and the stems they cover:

LemmaID	**wikAlap_1**	
wkAl	wikAl	agency
wkAl	wakAl	agency
wkAl	wikAl	agencies
wkAl	wakAl	agencies

LemmaID	**wikAlap_2**	
wkAl	wikAl	proxy

LemmaID	**{ibon_1**	
<bn	{ibon	son;junior (Jr.)
<bn	{ibon	daughter
>bnA'	>abonA'	sons;children
>bnA'	>abonA'	sons;children
>bnA&	>abonAW	sons;children
>bnA}	>abonA}	sons;children

LemmaID	**mut~aka>_1**	
mtk>	mut~aka>	support;prop
mtk&	mut~akaW	support;prop
mtk}	mut~aka}	support;prop
mtk|	mut~aka|	supports;props
mtk}	mut~aka}	supports;props
mtk	mut~aka	supports;props

LemmaID	**mut~aka>_2**	
mtk>	mut~aka>	cushion;couch
mtk&	mut~akaW	cushion;couch
mtk}	mut~aka}	cushion;couch
mtk|	mut~aka|	cushions;couches
mtk}	mut~aka}	cushions;couches
mtk	mut~aka	cushions;couches

Table 2. Examples of LemmaIDs from BAMA's Stem Dictionary

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. This method of clustering is quite simple. If two words share any possible analyses (by analysis we mean whatever the smallest unit we are looking at is, such as stems) then we declare that they are in the same cluster together. Given the analyses in Table 3 below, we cluster "Hjmy" and ">Hjm" together in one cluster and "qAsmy" and "tqAsm" together in another cluster. These words are clustered together because they share at least one stem in common. In this way, all morphological variants of a word belong to one cluster. 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.

Word	*Stems*
Hjmy	Hjmy, Hjm
>Hjm	>Hjm, Hjm
qAsmy	qAsmy, qAsm
tqAsm	tqAsm, qAsm

Example Clusters:

Hjm : Hjmy Hjm >Hjm
tnZyr : tnZyry tnZyr
smE : smEy smEAn >smE tsmE smE
fDA} : fDA}y fDA} wDA} DA}n
byr : byr kbyr byrty lbyryA lbyry byry byrA
dbr : tdbr >dbr dbry dbr
qAsm : tqAsm qAsm qAsmy

Table 3. Clustering Example

The results from each of the methods are displayed in Figure 1 below. We are especially encouraged by the strong improvements in three and four gram matches. These longer matches are more important because they contain more context and lead to more accurate translations.

	2-Gram Matches	*3-Gram Matches*	*4-Gram Matches*
UTF8	0.842	0.565	0.190
Stemmed	0.914	0.638	0.259
LemmaID	0.922	0.664	0.281
Clustering	0.950	0.755	0.393

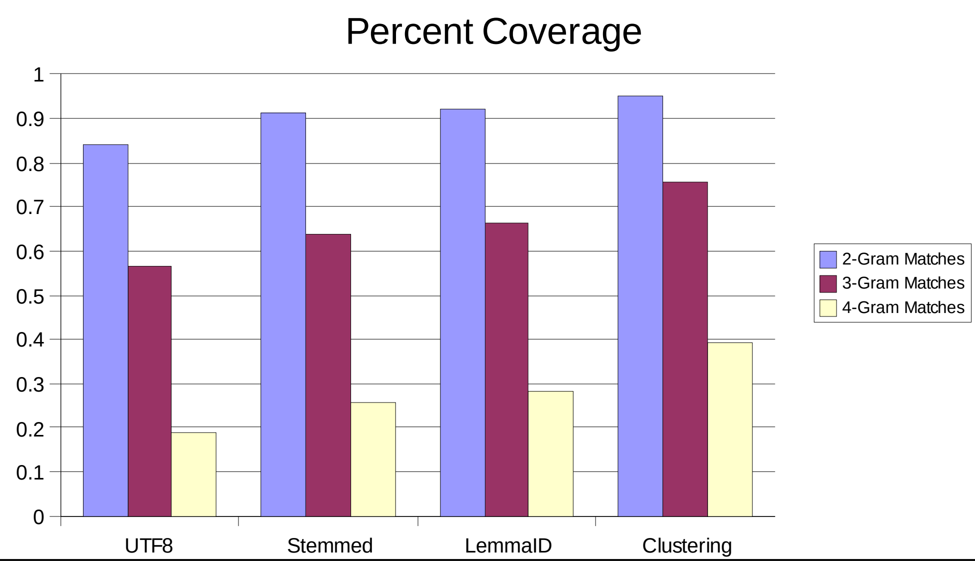

Figure 1. N-gram coverage results for different generalization methods.

3.2. Filtering

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

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 with the morphological features of each example in our corpus. The multiple analyses produced by BAMA make this difficult, since we do not know which features are correct. We could also merge all the possible morphological features and see what percentage of them match, but neither of these context-insensitive approaches worked very well in our experiments.

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 have a near-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 in the same cluster 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. This decreases the over-generalization by filtering out matches that do not share any analyses in common.

Consider the sentence "5 ملايين طن قمح روسي لمصر" ("5 *mlAyyn Tn qmH rwsy lmSr*"0 "5 million tons of Russian wheat to Egypt". First, each word in the sentence is matched to a cluster (some of which contain many words and some only contain a single word). The clusters for this sentence are shown in Table 4 below. If we are training the EBMT engine on this text, then the cluster is used by the EBMT engine to index each word. We also store metadata consisting of the original surface form of the word and all possible alternate stems in the EBMT index. When we attempt to translate this sentence we do the reverse process. First, for each word, we look up all examples that share the same cluster. Then we determine if any of the examples have the same surface form, which is the most precise match we can make. Failing any surface form matches, we check to see if any of the examples have an alternate stem in common. This filtering process allows us to match only the closest example(s) from our training corpus.

Word	Cluster	Metadata
mlAyyn	mlAyyn	(SURFACE mlAyyn) (ALT_STEM malAyiyn)
Tn	wT	(SURFACE Tn) (ALT_STEM Tan~) (ALT_STEM Tun~)
qmH	qmH	(SURFACE qmH) (ALT_STEM qamoH) (ALT_STEM qam~aH)
rwsy	rws	(SURFACE rwsy) (ALT_STEM ruws) (ALT_STEM ruwsiy~)
lmSr	mSr	(SURFACE lmSr) (ALT_STEM muSir~) (ALT_STEM miSor)

**Table 4. Clusters and metadata associated with the sentence "5 ملايين طن قمح روسي لمصر" ("*5 mlAyyn Tn qmH rwsy lmSr*")
"5 million tons of Russian wheat to Egypt"**

4. Remaining Issues

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. However, 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.

In Figure 2 below, there are two charts describing the EBMT output for the sentence "عزة ابراهيم يستقبل مسؤولا اقتصاديا سعوديا" ("*Ezp AbrAhym ystqbl msWwlA AqtSAdyA sEwdyA*") "Izzat Ibrahim meets a Saudi economic official". The chart on the top (with the Arabic characters) depicts our original system. The chart on the bottom (using the Buckwalter transliteration) depicts what happens when we use the methods described in this paper for generalization and filtering. (The input sentence in the bottom chart looks slightly different because each word has been replaced with a cluster.) Aside from the proper name "Izzat Ibrahim", all the entries in the top chart are single word translations. The bottom chart shows that we are now able to retrieve the bigram "economic official". This is important because the word for word translation would not get the proper ordering of "economic" and "official". The bottom chart also shows some variants of the single word translations ("economic" and "economy"; "officials" and "official") which are a direct result of the morphological generalization. In this case they are unnecessary because we already have a bigram covering that span of text, but often they are helpful. Lastly, as a result of increasing our coverage, extraneous translations can appear in the chart as well. "Official will" appears in the generalized chart below, but is not related to the text we are translating. This arc is likely the result of "official" properly generalizing, but also having a poor alignment such that "[ya]sotaqobil" ([ي]ستقبل) is aligned to "will" (recall that stems are being used, hence the prefix ي (*"ya"*) is not in the chart. It should be noted that while we want to limit spurious translations, it is normal to have some. This is why we employ a language model to help select the best translation candidates.

5. Conclusion and Current Work

In this paper we described a method for generalizing Arabic text specifically in the context of Example Based Machine Translation. We demonstrated three different techniques for generalization, of which the clustering technique we developed gives the greatest coverage. We also explained how we prevented over-generalization in our system by filtering the results. Although we can see substantial improvement in the translation candidates we obtain, we have so far been unable to improve actual translation quality. Current work is addressing this problem by focusing on the language model that selects the final translation candidates to use. We are also continuing to investigate to what extent we can further refine the filtering process by using additional morphological information.

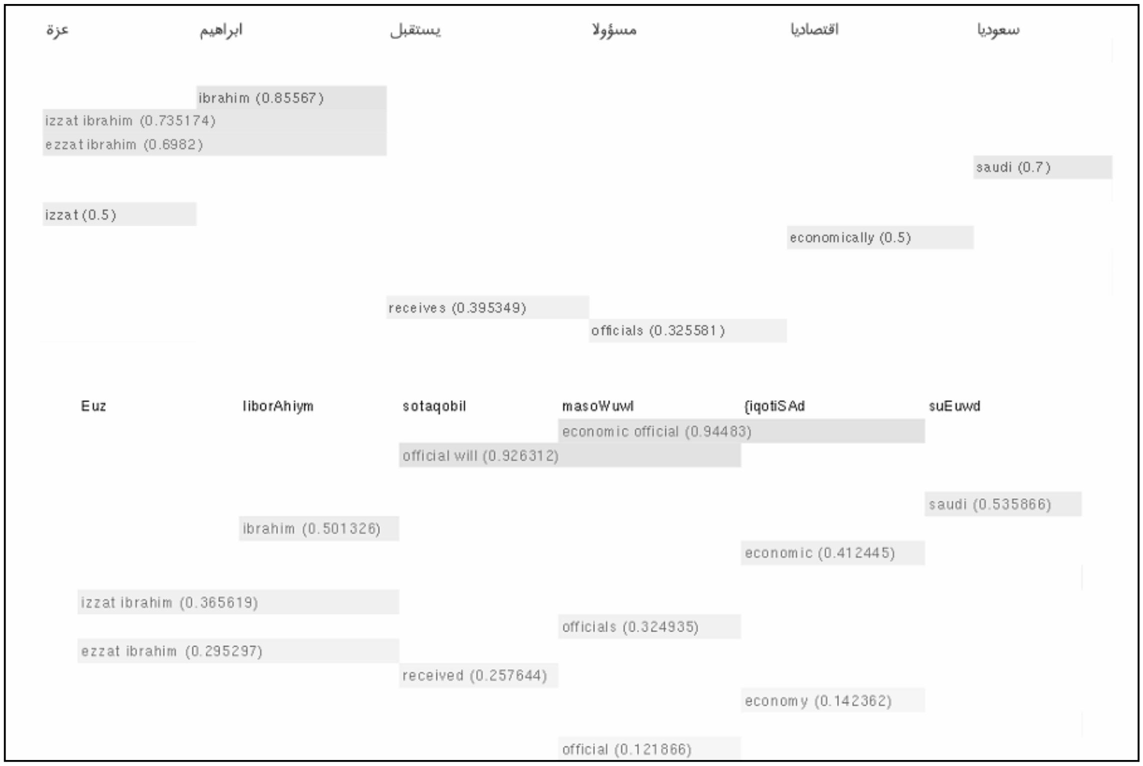

Figure 2. Two charts for the sentence
"عزة ابراهيم يستقبل مسؤولا اقتصاديا سعوديا"
("*Ezp AbrAhym ystqbl msWwlA AqtSAdyA sEwdyA*")
"Izzat Ibrahim meets a Saudi economic official"
The top chart is our original system and the bottom chart uses generalization

References

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

Brown, R. 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 (TMI-99), Chester, UK, August, pp. 22-32.

Brown, R. D. (2000a). Example-Based Machine Translation at Carnegie Mellon University. In *The ELRA Newsletter*, European Language Resources Association, vol 5:1, Jan-Mar.

Brown, R. D. (2000b). Automated Generalization of Translation Examples. In *Proceedings of the Eighteenth International Conference on Computational* Linguistics (COLING-2000), Saarbrücken, Germany, August, pp. 125-131.

Brown, R. D. (2004). A Modified Burrows-Wheeler Transform for Highly-Scalable Example-Based Translation", in *Machine Translation: From Real Users to Research*, Proceedings of the 6th Conference of the Association for Machine Translation (AMTA-2004), Washington, D.C., USA, September/October 2004, pp. 27-36. Springer, Lecture Notes in Artificial Intelligence, Volume 3265, ISSN 0302-9743.

Buckwalter, T. (2004). Buckwalter Arabic Morphological Analyzer Version 2.0. LDC2004L02 (http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2004L02)

Carl, M. and Way, A. (2003). Recent Advances in Example-Based Machine Translation. Dordrecht, The Netherlands: Kluwer Academic Publishers.

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* (AMTA-96), Montreal, Canada, pp. 268–272.

Frederking, R., Rudnicky, A., and Hogan, C. (1997). Interactive Speech Translation in the DIPLOMAT Project. Presented at the Spoken Language Translation Workshop at the 35th Meeting of the Association for Computational Linguistics (ACL-97), Madrid, Spain.

Somers, H. (1999). Example-based Machine Translation. *Machine Translation* 14(2), pp. 113-158.

Trujillo, A. (1999). Translation Engines: Techniques for Machine Translation. London, Great Britain: Springer-Verlag.