Unsupervised Learning of Arabic Stemming using a Parallel Corpus

Monica Rogati[†]
Computer Science Department,
Carnegie Mellon University
mrogati@cs.cmu.edu

Scott McCarley
IBM TJ Watson
Research Center
jsmc@watson.ibm.com

Yiming Yang
Language Technologies Institute,
Carnegie Mellon University
yiming@cs.cmu.edu

Abstract

This paper presents an unsupervised learning approach to building a non-English (Arabic) stemmer. The stemming model is based on statistical machine translation and it uses an English stemmer and a small (10K sentences) parallel corpus as its sole training resources. No parallel text is needed after the training phase. Monolingual, unannotated text can be used to further improve the stemmer by allowing it to adapt to a desired domain or genre. Examples and results will be given for Arabic , but the approach is applicable to any language that needs affix removal. Our resource-frugal approach results in 87.5% agreement with a state of the art, proprietary Arabic stemmer built using rules, affix lists, and human annotated text, in addition to an unsupervised component. Task-based evaluation using Arabic information retrieval indicates an improvement of 22-38% in average precision over unstemmed text, and 96% of the performance of the proprietary stemmer above.

1 Introduction

Stemming is the process of normalizing word variations by removing prefixes and suffixes. From an information retrieval point of view, prefixes and suffixes add little or no additional meaning; in most cases, both the efficiency and effectiveness of text processing applications such as information retrieval and machine translation are improved.

Building a rule-based stemmer for a new, arbitrary language is time consuming and requires experts with linguistic knowledge in that particular language. Supervised learning also requires large quantities of labeled data in the target language, and quality declines when using completely unsupervised methods. We would like to reach a compromise by using a few inexpensive and readily available resources in conjunction with unsupervised learning.

Our goal is to develop a **stemmer generator** that is *relatively language independent* (to the extent that the language accepts stemming) and is *trainable using little, inexpensive data*. This paper presents an unsupervised learning approach to non-English stemming. The stemming model is based on statistical machine translation and it uses an English stemmer and a small (10K sentences) parallel corpus as its sole training resources.

A *parallel corpus* is a collection of sentence pairs with the same meaning but in different languages (i.e. United Nations proceedings, bilingual newspapers, the Bible). Table 1 shows an example that uses the Buckwalter transliteration (Buckwalter, 1999). Usually, entire documents are translated by humans, and the sentence pairs are subsequently aligned by automatic means. A small parallel corpus can be available when native speakers and translators are not, which makes building a stemmer out of such corpus a preferable direction.

[†]Work done while a summer intern at IBM TJ Watson Research Center

Arabic	English
m$rwE Altqryr	Draft report
wAkdt mmvlp zAmbyA End ErDhA lltqryr An bldhA y$hd tgyyrAt xTyrp wbEydp Almdy fy AlmydAnyn AlsyAsy wAlAqtSAdy	In introducing the report, the representative of Zambia emphasised that her country was undergoing serious and far-reaching changes in the political and economic field.

Table 1: A Tiny Arabic-English Parallel Corpus

We describe our approach towards reaching this goal in section 2. Although we are using resources other than monolingual data, the unsupervised nature of our approach is preserved by the fact that no direct information about non-English stemming is present in the training data.

Monolingual, unannotated text in the target language is readily available and can be used to further improve the stemmer by allowing it to adapt to a desired domain or genre. This optional step is closer to the traditional unsupervised learning paradigm and is described in section 2.4, and its impact on stemmer quality is described in 3.1.4.

Our approach (denoted by UNSUP in the rest of the paper) is evaluated in section 3.1 by comparing it to a proprietary Arabic stemmer (denoted by GOLD). The latter is a state of the art Arabic stemmer, and was built using rules, suffix and prefix lists, and human annotated text. GOLD is an earlier version of the stemmer described in (Lee et al.,).

The task-based evaluation section 3.2 compares the two stemmers by using them as a preprocessing step in the TREC Arabic retrieval task. This section also presents the improvement obtained over using unstemmed text.

1.1 Arabic details

In this paper, Arabic was the target language but the approach is applicable to any language that needs affix removal. In Arabic, unlike English, both prefixes and suffixes need to be removed for effective stemming. Although Arabic provides the additional challenge of infixes, we did not tackle them because they often substantially change the meaning. Irregular morphology is also beyond the scope of this paper. As a side note for readers with linguistic background (Arabic in particular), we do not claim that the resulting stems are units representing the entire paradigm of a lexical item. The main purpose of stemming as seen in this paper is to conflate the token space used in statistical methods in order to improve their effectiveness. The *quality* of the resulting tokens as perceived by humans is not as important, since the stemmed output is intended for computer consumption.

1.2 Related Work

The problem of *unsupervised stemming or morphology* has been studied using several different approaches. For Arabic, good results have been obtained for plural detection (Clark, 2001). (Goldsmith, 2001) used a minimum description length paradigm to build Linguistica, a system for which the reported accuracy for European languages is cca. 83%. Note that the results in this section are not directly comparable to ours, since we are focusing on Arabic.

A notable contribution was published by Snover (Snover, 2002), who defines an objective function to be optimized and performs a search for the stemmed configuration that optimizes the function over all stemming possibilities of a given text.

Rule-based stemming for Arabic is a problem studied by many researchers; an excellent overview is provided by (Larkey et al.,).

Morphology is not limited to prefix and suffix removal; it can also be seen as mapping from a word to an arbitrary meaning carrying token. Using an LSI approach, (Schone and Jurafsky,) obtained 88% accuracy for English. This approach also deals with irregular morphology, which we have not addressed.

A parallel corpus has been successfully used before by (Yarowsky et al., 2000) to project part of speech tags, named entity tags, and morphology information from one language to the other. For a parallel corpus of comparable size with the one used in our results, the reported accuracy was 93% for French (when the English portion was also available); however, this result only covers 90% of the tokens. Accuracy was later improved using suffix trees.

(Diab and Resnik, 2002) used a parallel corpus for word sense disambiguation, exploiting the fact that different meanings of the same word tend to be translated into distinct words.

2 Approach

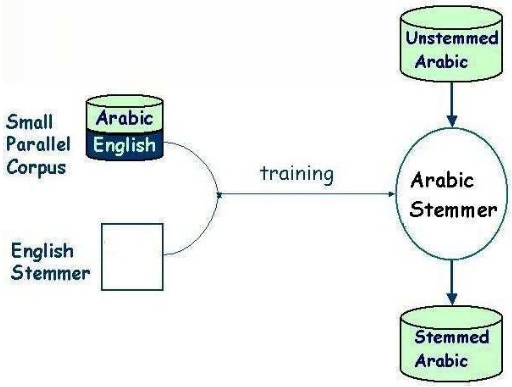

Figure 1: Approach Overview

Our approach is based on the availability of the following three resources:

- a small parallel corpus
- an English stemmer
- an optional unannotated Arabic corpus

Our goal is to train an Arabic stemmer using these resources. The resulting stemmer will simply stem Arabic without needing its English equivalent.

We divide the training into two logical steps:

- **Step 1:** Use the small parallel corpus
- **Step 2:** (optional) Use the monolingual corpus

The two steps are described in detail in the following subsections.

2.1 Step 1: Using the Small Parallel Corpus

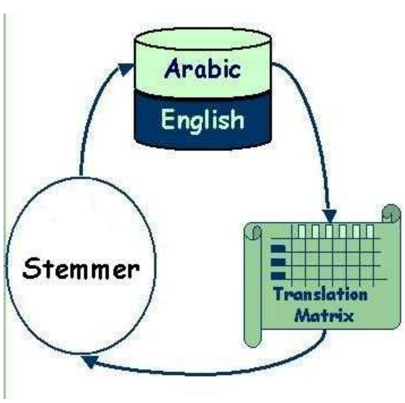

Figure 2: Step 1 Iteration

In Step 1, we are trying to exploit the English stemmer by stemming the English half of the parallel corpus and building a translation model that will establish a correspondence between meaning carrying substrings (the stem) in Arabic and the English stems.

For our purposes, a *translation model* is a matrix of translation probabilities p(Arabic stem| English stem) that can be constructed based on the small parallel corpus (see subsection 2.2 for more details). The Arabic portion is stemmed with an initial guess (discussed in subsection 2.1.1)

Conceptually, once the translation model is built, we can stem the Arabic portion of the parallel corpus by scoring all possible stems that an Arabic word can have, and choosing the best one. Once the Arabic portion of the parallel corpus is stemmed, we can build a more accurate translation model and repeat the process (see figure 2). However, in practice, instead of using a harsh cutoff and only keeping the best stem, we impose a probability distribution over the candidate stems. The distribution starts out uniform and then converges towards concentrating most of the probability mass in one stem candidate.

2.1.1 The Starting Point

The starting point is an inherent problem for unsupervised learning. We would like our stemmer to give good results starting from a very general initial guess (i.e. random). In our case, the starting point is the initial choice of the stem for each individual word. We distinguish several solutions:

- **No stemming.**

  This is not a desirable starting point, since affix probabilities used by our model would be zero.

- **Random stemming**

  As mentioned above, this is equivalent to imposing a uniform prior distribution over the candidate stems. This is the most general starting point.

- **A simple language specific rule - if available**

  If a simple rule is available, it would provide a better than random starting point, at the cost of reduced generality. For Arabic, this simple rule was to use *Al* as a prefix and *p* as a suffix. This rule (or at least the first half) is obvious even to non-native speakers looking at transliterated text. It also constitutes a surprisingly high baseline.

2.2 The Translation Model [*]

We adapted Model 1 (Brown et al., 1993) to our purposes. Model 1 uses the concept of *alignment* between two sentences **e** and **f** in a parallel corpus; the alignment is defined as an object indicating for each word e_i which word f_j generated it. To obtain the probability of an foreign sentence **f** given the English sentence **e**, Model 1 sums the products of the translation probabilities over all possible alignments:

$$Pr(\mathbf{f}|\mathbf{e}) \sim \sum_{\{a\}} \prod_{j=1}^{m} t(f_j|e_{a_j})$$

The alignment variable a_i controls which English word the foreign word f_i is aligned with. $t(f|e)$ is simply the translation probability which is refined iteratively using EM. For our purposes, the translation probabilities (in a *translation matrix*) are the final product of using the parallel corpus to train the translation model.

To take into account the weight contributed by each stem, the model's iterative phase was adapted to use the sum of the weights of a word in a sentence instead of the count.

2.3 Candidate Stem Scoring

As previously mentioned, each word has a list of substrings that are possible stems. We reduced the problem to that of placing two separators inside each Arabic word; the "candidate stems" are simply the substrings inside the separators. While this may seem inefficient, in practice words tend to be short, and one or two letter stems can be disallowed.

An initial, naive approach when scoring the stem would be to simply look up its translation probability, given the English stem that is most likely to be its translation in the parallel sentence (i.e. the English stem aligned with the Arabic stem candidate). Figure 3 presents scoring examples before normalization.

English Phrase:	the *advisory* committee
Arabic Phrase:	Alljnp *AlAst$Aryp*

Task: stem *AlAst$Aryp*

Choices	Score
AlAst$Aryp	0.2
*Al***Ast$Aryp**	0.7
*Al***Ast$Ary***p*	0.8
*AlA***st$Ary***p*	0.1
⋮	⋮

Figure 3: Scoring the Stem

However, this approach has several drawbacks that prevent us from using it on a corpus other than the training corpus. Both of the drawbacks below are brought about by the small size of the parallel corpus:

- *Out-of-vocabulary words*: many Arabic stems will not be seen in the small corpus
- *Unreliable translation probabilities* for low-frequency stems.

We can avoid these issues if we adopt an alternate view of stemming a word, by looking at the prefix and the suffix instead. Given the word, the choice of prefix and suffix uniquely determines the stem. Since the number of unique affixes is much smaller by definition, they will not have the two problems above, even when using a small corpus. These probabilities will be considerably more reliable and are a very important part of the information extracted from the parallel corpus. Therefore, the score of a candidate stem should be based on the score of the corresponding prefix and the suffix, in addition to the score of the stem string itself:

$$score(\text{“}pas'') = f(p) \times f(a) \times f(s)$$

where a = Arabic stem, p = prefix, s=suffix

When scoring the prefix and the suffix, we could simply use their probabilities from the previous stemming iteration. However, there is additional information available that can be successfully used to condition and refine these probabilities (such as the length of the word, the part of speech tag if given etc.).

[*]Note that the algorithm to build the translation model is not a "resource" per se, since it is a language-independent algorithm.

English Phrase:	the *advisory* committee
Arabic Phrase:	Alljnp *AlAst$Aryp*

<u>Task</u>: stem *AlAst$Aryp*

Choices	Score
$AlAst\$Aryp$	0.8
Al$Ast\$Aryp$	0.7
Al$Ast\$Ary$	0.6
AlA$st\$Ary$**p**	0.1
$\vdots$	$\vdots$

Figure 4: Alternate View: Scoring the Prefix and Suffix

2.3.1 Scoring Models

We explored several stem scoring models, using different levels of available information. Examples include:

- Use the stem translation probability alone

  $score = t(a|e)$

  where a = Arabic stem, e = corresponding word in the English sentence

- Also use prefix (p) and suffix (s) conditional probabilities; several examples are given in table 2.

Probability conditioned on	Scoring Formula
the candidate stem	$t(a|e) \times \frac{p(p,s|a)+p(s|a)\times p(p|a)}{2}$
the length of the unstemmed Arabic word (len)	$t(a|e) \times \frac{p(p,s|len)+p(s|len)\times p(p|len)}{2}$
the possible prefixes and/or suffixes	$t(a|e) \times p(s|S_{possible}) \times p(p|P_{possible})$
the $first$ and $last$ letter	$t(a|e)\times p(s|last)\times p(p|first)$

Table 2: Example Scoring Models

The first two examples use the joint probability of the prefix and suffix, with a smoothing back-off (the product of the individual probabilities). Scoring models of this form proved to be poor performers from the beginning, and they were abandoned in favor of the last model, which is a fast, good approximation to the third model in Table 2. The last two models successfully solve the problem of the empty prefix and suffix accumulating excessive probability, which would yield to a stemmer that never removed any affixes. The results presented in the rest of the paper use the last scoring model.

2.4 Step 2: Using the Unlabeled Monolingual Data

This optional second step can adapt the trained stemmer to the problem at hand. Here, we are moving away from providing the English equivalent, and we are relying on learned prefix, suffix and (to a lesser degree) stem probabilities. In a new domain or corpus, the second step allows the stemmer to learn new stems and update its statistical profile of the previously seen stems.

This step can be performed using monolingual Arabic data, with no annotation needed. Even though it is optional, this step is recommended since its sole resource can be the data we would need to stem anyway (see Figure 5).

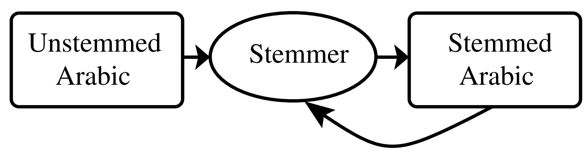

Figure 5: Step 2 Detail

Step 1 produced a functional stemming model. We can use the corpus statistics gathered in Step 1 to stem the new, monolingual corpus. However, the scoring model needs to be modified, since $t(a|e)$ is no longer available. By removing the conditioning, the first/last letter scoring model we used becomes

$$score = p(a) \times p(s|last) \times p(p|first)$$

The model can be updated if the stem candidate score/probability distribution is sufficiently skewed, and the monolingual text can be stemmed iteratively using the new model. The model is thus adapted to the particular needs of the new corpus; in practice, convergence is quick (less than 10 iterations).

3 Results

3.1 Unsupervised Training and Testing

For unsupervised training in Step 1, we used a small parallel corpus: 10,000 Arabic-English sentences from the United Nations(UN) corpus, where the English part has been stemmed and the Arabic transliterated.

For unsupervised training in Step 2, we used a larger, Arabic only corpus: 80,000 different sentences in the same dataset.

The test set consisted of 10,000 different sentences in the UN dataset; this is the testing set used below unless specified.

We also used a larger corpus (a year of Agence France Press (AFP) data, 237K sentences) for Step 2 training and testing, in order to gauge the robustness and adaptation capability of the stemmer. Since the UN corpus contains legal proceedings, and the AFP corpus contains news stories, the two can be seen as coming from different domains.

3.1.1 Measuring Stemmer Performance

In this subsection the accuracy is defined as agreement with GOLD. GOLD is a state of the art, proprietary Arabic stemmer built using rules, suffix and prefix lists, and human annotated text, in addition to an unsupervised component. GOLD is an earlier version of the stemmer described in (Lee et al.,). Freely available (but less accurate) Arabic light stemmers are also used in practice.

When measuring accuracy, all tokens are considered, including those that cannot be stemmed by simple affix removal (irregulars, infixes). Note that our baseline (removing Al and p, leaving everything unchanged) is higher that simply leaving all tokens unchanged.

For a more relevant task-based evaluation, please refer to Subsection 3.2.

3.1.2 The Effect of the Corpus Size: How little parallel data can we use?

We begin by examining the effect that the size of the parallel corpus has on the results after the first step. Here, we trained our stemmer on three different corpus sizes: 50K, 10K, and 2K sentences. The high baseline is obtained by treating *Al* and *p* as affixes. The 2K corpus had acceptable results (if this is all the data available). Using 10K was significantly better; however the improvement obtained when five times as much data (50K) was used was insignificant. Note that different languages might have different corpus size needs. All other results

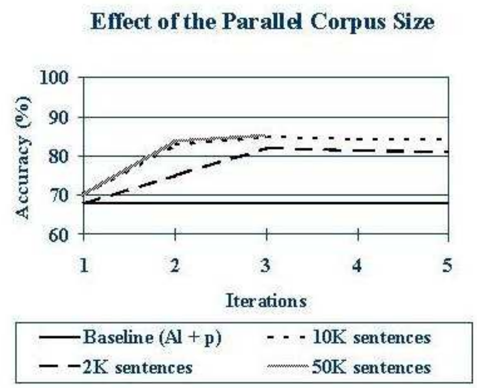

Figure 6: Results after Step 1 : Corpus Size Effect

in this paper use 10K sentences.

3.1.3 The Knowledge-Free Starting Point after Step 1

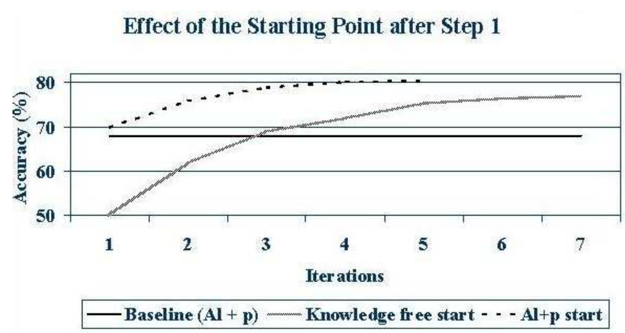

Figure 7: Results after Step 1 : Effect of Knowing the *Al+p* rule

Although severely handicapped at the beginning, the knowledge-free starting point manages to narrow the performance gap after a few iterations. Knowing the Al+p rule still helps at this stage. However, the performance gap is narrowed further in Step 2 (see figure 8), where the knowledge free starting point benefitted from the monolingual training.

3.1.4 Results after Step 2: Different Corpora Used for Adaptation

Figure 8 shows the results obtained when augmenting the stemmer trained in Step 1. Two different monolingual corpora are used: one from the same domain as the test set (80K UN), and one from a different domain/corpus, but three times larger (237K AFP). The larger dataset seems to be more useful in improving the stemmer, even though the domain was different.

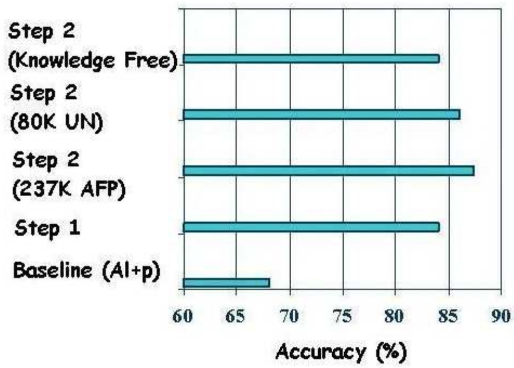

Figure 8: Results after Step 2 (Monolingual Corpus)

The baseline and the accuracy after Step 1 are presented for reference.

3.1.5 Cross-Domain Robustness

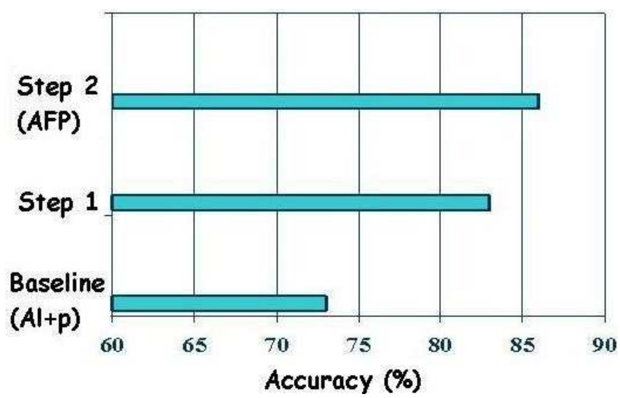

Figure 9: Results after Step 2 : Using a Different Test Set

We used an additional test set that consisted of 10K sentences taken from AFP, instead of UN as in previous experiments shown in figure 8 . Its purpose was to test the cross-domain robustness of the stemmer and to further examine the importance of applying the second step to the data needing to be stemmed.

Figure 9 shows that, even though in Step 1 the stemmer was trained on UN proceedings, the results on the cross-domain (AFP) test set are comparable to those from the same domain (UN, figure 8). However, for this particular test set the baseline was much higher; thus the relative improvement with respect to the baseline is not as high as when the unsupervised training and testing set came from the same collection.

3.2 Task-Based Evaluation : Arabic Information Retrieval

Task Description:

Given a set of Arabic documents and an Arabic query, find a list of documents relevant to the query, and rank them by probability of relevance.

We used the TREC 2002 documents (several years of AFP data), queries and relevance judgments. The 50 queries have a shorter, "title" component as wel as a longer "description". We stemmed both the queries and the documents using UNSUP and GOLD respectively. For comparison purposes, we also left the documents and queries unstemmed.

The UNSUP stemmer was trained with 10K UN sentences in Step 1, and with one year's worth of monolingual AFP data (1995) in Step 2.

Evaluation metric: The evaluation metric used below is *mean average precision* (the standard IR metric), which is the mean of average precision scores for each query. The *average precision* of a single query is the mean of the precision scores after each relevant document retrieved. Note that average precision implicitly includes recall information. *Precision* is defined as the ratio of relevant documents to total documents retrieved up to that point in the ranking.

Results

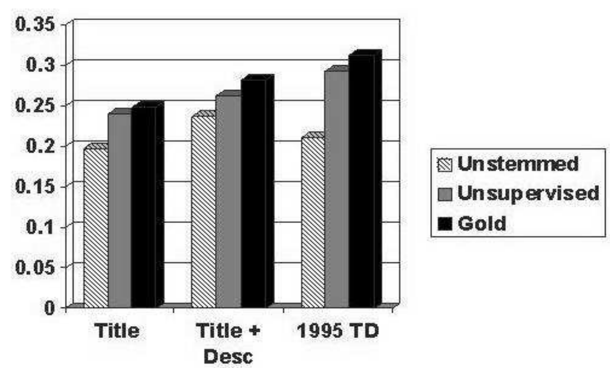

Figure 10: Arabic Information Retrieval Results

We looked at the effect of different testing conditions on the mean average precision for the 50 queries. In Figure 10, the first set of bars uses the query titles only, the second set adds the description, and the last set restricts the results to one year (1995), using both the title and description. We tested this last condition because the unsupervised stemmer was refined in Step 2 using 1995 documents. The last group of bars shows a higher relative improvement over the unstemmed baseline; however, this last condition is based on a smaller sample of relevance judgements (restricted to one year) and is therefore not as representative of the IR task as the first two testing conditions.

4 Conclusions and Future Work

This paper presents an unsupervised learning approach to building a non-English (Arabic) stemmer using a small sentence-aligned parallel corpus in which the English part has been stemmed. No parallel text is needed to use the stemmer. Monolingual, unannotated text can be used to further improve the stemmer by allowing it to adapt to a desired domain or genre. The approach is applicable to any language that needs affix removal; for Arabic, our approach results in 87.5% agreement with a proprietary Arabic stemmer built using rules, affix lists, and human annotated text, in addition to an unsupervised component. Task-based evaluation using Arabic information retrieval indicates an improvement of 22-38% in average precision over unstemmed text, and 93-96% of the performance of the state of the art, language specific stemmer above.

We can speculate that, because of the statistical nature of the unsupervised stemmer, it tends to focus on the same kind of meaning units that are significant for IR, whether or not they are linguistically correct. This could explain why the gap betheen GOLD and UNSUP is narrowed with task-based evaluation and is a desirable effect when the stemmer is to be used for IR tasks.

We are planning to experiment with different languages, translation model alternatives, and to extend task-based evaluation to different tasks such as machine translation and cross-lingual topic detection and tracking.

5 Acknowledgements

We would like to thank the reviewers for their helpful observations and for identifying Arabic misspellings. This work was partially supported by the Defense Advanced Research Projects Agency and monitored by SPAWAR under contract No. N66001-99-2-8916. This research is also sponsored in part by the National Science Foundation (NSF) under grants EIA-9873009 and IIS-9982226, and in part by the DoD under award 114008-N66001992891808. However, any opinions, views, conclusions and findings in this paper are those of the authors and do not necessarily reflect the position of policy of the Government and no official endorsement should be inferred.

References

P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer. 1993. The mathematics of machine translation: Parameter estimation. In *Computational Linguistics*, pages 263–311.

Tim Buckwalter. 1999. Buckwalter transliteration. http://www.cis.upenn.edu/~cis639/arabic/info/translit-chart.html.

Alexander Clark. 2001. Learning morphology with pair hidden markov models. In *ACL (Companion Volume)*, pages 55–60.

Mona Diab and Philip Resnik. 2002. An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 255–262, July.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. In *Computational Linguistics*.

Leah Larkey, Lisa Ballesteros, and Margaret Connell. Improving stemming for arabic information retrieval: Light stemming and co-occurrence analysis. In *SIGIR 2002*, pages 275–282.

Young-Suk Lee, Kishore Papineni, Salim Roukos, Ossama Emam, and Hany Hassan. Language model based arabic word segmentation. In *To appear in ACL 2003*.

Patrick Schone and Daniel Jurafsky. Knowledge-free induction of morphology using latent semantic analysis. In *4th Conference on Computational Natural Language Learning, Lisbon, 2000*.

Matthew Snover. 2002. An unsupervised knowledge free algorithm for the learning of morphology in natural languages. Master's thesis, Washington University, May.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2000. Inducing multilingual text analysis tools via robust projection across aligned corpora.