# Parsing Arabic Dialects

Final Report – Version 1, January 18, 2006

Owen Rambow, Columbia University
David Chiang, University of Maryland
Mona Diab, Columbia University
Nizar Habash, Columbia University
Rebecca Hwa, University of Pittsburgh
Khalil Sima'an, University of Amsterdam
Vincent Lacey, Georgia Tech
Roger Levy, Stanford University
Carol Nichols, University of Pittsburgh
Safiullah Shareef, Johns Hopkins University
Contact: rambow@cs.columbia.edu

# Contents

This report summarizes work done at a Johns Hopkins Summer Workshop during the summer of 2005. The authors are grateful to Johns Hopkins, and the faculty, students, and staff at Johns Hopkins who made the summer workshop possible.

This material is based upon work supported in part by the National Science Foundation under Grant No. 0121285. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

# Chapter 1

# Introduction

## 1.1 Goals of This Work

The Arabic language is a collection of spoken dialects and a standard written language. The dialects show phonological, morphological, lexical, and syntactic differences comparable to those among the Romance languages. The standard written language is the same throughout the Arab world: Modern Standard Arabic (MSA). MSA is also used in some scripted spoken communication (news casts, parliamentary debates). MSA is based on Classical Arabic and is itself not a native language (children do not learn it from their parents but in school). Most native speakers of Arabic are unable to produce sustained spontaneous MSA. The most salient variation among the dialects is geographic; this variation is continuous, and proposed groupings into a small number of geographic classes do not mean that there are, for example, only five Arabic dialects. In addition to the geographic continuum, the dialects can vary according to a large number of additional factors: the urban/rural distinction, the Bedouin/sedentary distinction, gender, or religion.

The multidialectal situation has important negative consequences for Arabic natural language processing (NLP): since the spoken dialects are not officially written, it is very costly to obtain adequate corpora, even unannotated corpora, to use for training NLP tools such as parsers. While it is true that in unofficial written communication, in particular in electronic media such as web logs and bulletin boards, often ad-hoc transcriptions of dialects are used (since there is no official orthography), the inconsistencies in the orthography reduce the value of these corpora. Furthermore, there are almost no parallel corpora involving one dialect and MSA.

In this paper, we address the problem of parsing transcribed spoken Levantine Arabic (LA), which we use as a representative example of the Arabic dialects.[1] Our work is based on the assumption that it is easier to manually create new resources that relate LA to MSA than it is to manually create syntactically annotated corpora in LA. Our approaches do not assume the existence of any annotated LA corpus (except for development and testing), nor of a parallel LA-MSA corpus. Instead, we assume we

[1] We exclude from this study part-of-speech (POS) tagging and LA/MSA lexicon induction.

have at our disposal a lexicon that relates LA lexemes to MSA lexemes, and knowledge about the morphological and syntactic differences between LA and MSA. For a single dialect, it may seem that it is easier to create corpora than to encode all this knowledge explicitly. In response, we argue that because the dialects show important similarities, it will be easier to reuse and modify explicit linguistic resources for a new dialect, than to create a new corpus for it. The goal of this paper is to show that leveraging LA/MSA resources is feasible; we do not provide a demonstration of cost-effectiveness.

This report is organized as follows. ***Para needs work!*** After discussing related work and available corpora, we present linguistic issues in LA and MSA (Section 1.2). We then proceed to discuss three approaches: sentence transduction, in which the LA sentence to be parsed is turned into an MSA sentence and then parsed with an MSA parser (Section 5.2); treebank transduction, in which the MSA treebank is turned into an LA treebank (Section 5.3); and grammar transduction, in which an MSA grammar is turned into an LA grammar which is then used for parsing LA (Section 5.4). We summarize and discuss the results in Section 6.

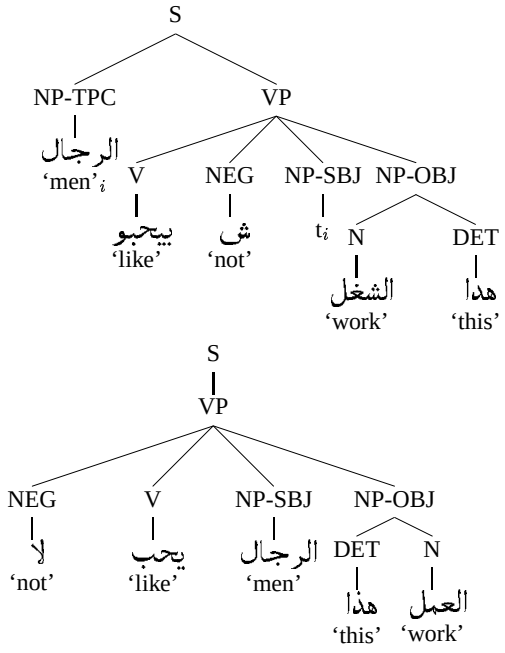


Figure 1.1: LDC-style left-to-right phrase structure trees for LA (left) and MSA (right) for sentence (1)

## 1.2 Linguistic Facts

We illustrate the differences between LA and MSA using an example:

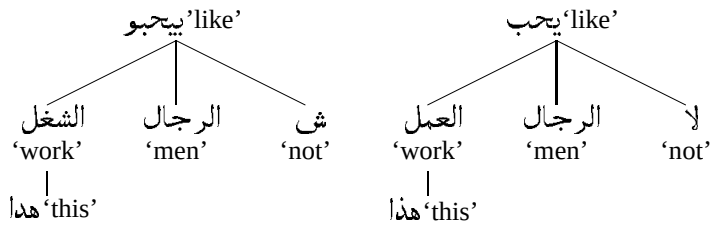


Figure 1.2: Unordered dependency trees for LA (left) and MSA (right) for sentence (1)

(1) a. الرجال بيحبو ش الشغل هدا (LA)

AlrjAl byHbw $ Al$gl hdA
the-men like not the-work this

the men do not like this work

b. لا يحب الرجال هذا العمل (MSA)

lA yHb AlrjAl h*A AlEml
not like the-men this the-work

the men do not like this work

Lexically, we observe that the word for 'work' is الشغل *Al$gl* in LA but العمل *AlEml* in MSA. In contrast, the word for 'men' is the same in both LA and MSA (though LA also has another word), الرجال *AlrjAl*. There are typically also differences in function words, in our example ش *$* (LA) and لا *lA* (MSA) for 'not'. Morphologically, we see that LA بيحبو *byHbw* has the same stem as MA يحب *yHb*, but with two additional morphemes: the present aspect marker *b-* which does not exist in MSA, and the agreement marker -w, which is used in MSA only in subject-initial sentences, while in LA it is always used.

Syntactically, we observe three differences. First, the subject precedes the verb in LA (SVO order), but follows in MSA (VSO order). This is in fact not a strict requirement, but a strong preference: both varieties allow both orders. Second, we see that the demonstrative determiner follows the noun (with cliticized definite article) in LA, but precedes it in MSA. Finally, we see that the negation marker ش *$* follows the verb in LA, while it precedes the verb in MSA.[2] The two phrase structure trees are shown in Figure 1.1 in the LDC convention. Unlike the phrase structure trees, the (unordered) dependency trees are isomorphic: they differ only in the node labels (Figure 1.2).

[2]Levantine also has other negation markers that precede the verb, as well as the circumfix *m- -$*.

# Chapter 2

# Linguistic Resources

## 2.1 Corpora

We use the MSA treebanks 1, 2 and 3 (ATB) from the LDC (Maamouri et al., 2004), which consist of 625,000 words (750,000 tokens after tokenization) of newspaper and newswire text (1900 news articles from 3 different sources). ATB is manually morphologically disambiguated and syntactically annotated. We split the corpus into 10% development data, 80% training data and 10% test data, respectively, all respecting document/article boundaries. The development and training data were randomized on the document level. The training data (ATB-Train) comprises 17,617 sentences and 588,244 tokens.

The Levantine treebank LATB comprises 33,000 words of treebanked conversational telephone transcripts collected as part of the LDC CALL HOME project. The treebanked section is primarily Jordanian dialect The data is annotated by the LDC for speech effects such as disfluencies and repairs. We removed the speech effects, rendering the data more text-like. The orthography and syntactic analysis chosen by the LDC for LA closely follow previous choices for MSA, see Figure 1.1 for two examples. The LATB is used exclusively for development and testing, not for training. We split the data in half respecting document boundaries. The resulting development data comprises 1928 sentences and 11151 tokens (DEV). The test data comprises 2051 sentences and 10,644 tokens (TEST).

Both the LATB and ATB are transliterated into ASCII characters using the Buckwalter transliteration scheme.[1] For all the experiments, we use the non-vocalized (undiacritized) version of both treebanks, as well as the collapsed POS tag set provided by the LDC for MSA and LA.

[1] http://www.ldc.upenn.edu/myl/morph/buckwalter.html

## 2.2 Lexicons

Two lexicons were created to bridge the gap between MSA and LA: a small lexicon comprising 321 LA/MSA word form pairs covering LA closed-class words and a few frequent open-class words; and a big lexicon which contains the small lexicon and an additional 1,560 LA/MSA word form pairs. We associate with the word pairs in the two lexicons both uniform probabilities and biased probabilities using Expectation Maximization (EM). Thereby, this process yields four different lexicons: Small lexicon with uniform probabilities (S-LEX-UN); Small Lexicon with EM-based probabilities (S-LEX-EM); Big Lexicon with uniform probabilities (B-LEX-UN); and Big Lexicon with EM-based probabilities (B-LEX-EM).

# Chapter 3

# Lexicon Induction from Corpora

## 3.1 Background

Given a pair of corpora, possibly in two different languages, is it possible to build a lexical mapping (translation mapping) between words in the one corpus and words in the other? A successful lexical mapping would relate a word in the one corpus only to words in the other corpus that mirror the meaning of that word (possibly in its context). In principle, in a lexical mapping some of the ambiguity of a word, with regard to sense and translation, could be resolved since the mapping might map some occurences of the word (i.e., tokens) to some specific translations but not other possible ones. For example, the first occurence of the word *bank* in *A finanical bank is different from a river bank.* would translate to Dutch as *bank* whereas the second would translate into *oever*. Given the word *bank*, both senses/translations are possible. When the word is embedded in context, one sense only might be suitable.

How many word tokens can be mapped and how much ambiguity can be resolved depends, in large part, on the ways in which the two corpora are found to be similar. Parallel corpora that constitute a translation of one another, for example, could be very similar along various dimensions (including topic, genre, style, mode and so on), whereas unrelated corpora could differ along any subset of these and other dimensions. When the two corpora are strongly related, it might be possible to map a *word token* in context to various word tokens in their context, thereby mapping specific tokens to tokens. When the two corpora are highly unrelated, it might be more suitable to map a word type (rather than token) to various word types (leading to a so called translation lexicon akin to a dictionary), thereby preserving part of the ambiguity that a word type represents. Hence, given a pair of corpora, one could actively pursue the question: what lexical (translation) mapping would fit this pair of corpora best?

Because a certain sense of reduced ambiguity is the driving force behind finding such a lexical mapping for a given pair of corpora, it stands in sheer contrast to a (translation) dictionary which is meant to present all possible translations (and senses)

of a word type (rather than a word token). Therefore, when such a mapping can be built between pairs of corpora, it promises to yield a powerfull lexical resource for various applications, and possibly for porting different tools (e.g., POS taggers and parsers) from one language to another.

One specific, extensively studied kind of lexical mapping is the alignment of the word tokens in two parallel corpora for the purposes of statistical machine translation (Brown et al., 1990; Al-Onaizan et al., 1999; Melamed, 2000). In a pair of parallel corpora, every sentence (or text chunk) from the one corpus is aligned with a sentence (or text chunk) in the other corpus. Sentence alignment implies that the alignment between word tokens can be constrained to a pair of aligned sentences. The work on aligning parallel corpora has yielded very accurate lexical mappings that are being employed in various forms within machine translation. However, parallel corpora are often not available for various reasons. For example, because Modern Standard Arabic (MSA) is a written language and the (strongly related) Arabic dialects (e.g., Levantine, Egyptian, Moroccan) are spoken, it is highly unlikely that a pair of parallel corpora MSA-dialect exist. For these situations, one has to do with whatever pairs of corpora exist.

According to Rapp (Rapp, 1999), the alignment of parallel corpora may exploit clues concerning the correspondence of sentence and word order, correlation between word frequencies and the availability of cognates in parallel corpora. For non-parallel corpora, the first clue, which strongly constrains the lexical mapping to tokens within a pair of sentences, is not available. The second clue weakens as the corpora become more and more unrelated, whereas the third one captures only a limited set of lexical correspondences. Hence, work on inducing lexical mappings, such as translation lexicons, from comparable and unrelated corpora (Rapp, 1999; Fung, 1995; Fung and McKeown, 1997; Diab and Finch, 2000) is based on the correlation of the co-occurrence patterns of words in the one corpus with the co-occurence patterns of words in the other corpus. Work along this general approach employs various distributional similarity measure based on statistics collected in co-occurence matrices of words and their neighboring words. Usually, a "seed translation lexicon" is needed to initiate the process of mapping because the matrix entry for a word is defined in terms of its co-occurence with words in the seed translation lexicon. The induction algorithm adds word pairs to the translation lexicon.

In this chapter, we study the problem of inducing a lexical mapping between different pairs of comparable and unrelated corpora. Our goal is to explore the kind of factors that play a role in the accuracy of the induced lexical mapping, and to suggest ways to improve accuracy in light of these factors. The kind of factors that we consider are of three types:

Deep (semantic/discourse/pragmatic) factors such as topic/genre/mode that are expressed in the statistics for both the choice of words and the sentence word order.

Surface statistical factors such as corpora sizes, the number of common words, or in general the sparseness of the statistics.

Quality and size of the seed translation lexicon.

Intuitively speaking, the first kind of factors puts an upperbound on the quality of the translation lexicon that can be obtained (quality in terms of number of pairs and the level of ambiguity). This factor can be determinental for the kind of mapping that suits the two corpora at hand. Hence, a major hypothesis of the present work states that the more unrelated the corpora become, the more ambiguous the mapping must be to achieve reasonable accuracy and coverage.

The second kind puts an upperbound on the success of the statistical methods. The third kind is specific to the method taken in most work on this topic (Rapp, 1999; Fung, 1995; Fung and McKeown, 1997; Diab and Finch, 2000) which are based on the "find one, get more" principle. The seed lexicon constitutes the point of initialization for the algorithm which could be very important for the quality of the "get more" part of the algorithm.

In this chapter we aim at exploring the effect of each of the factors listed above on the induction of a translation mapping depending on the level of (un)relatedness of a pair of corpora. For the induction of a translation lexicon (mapping) we base our expolrations on Rapp's method (Rapp, 1999). Rapp's method is based on the same general ideas as other existing work on this topic and seems as good as any for that matter. We propose a simple adaption of Rapp's method and present various empirical explorations on differing pairs of corpora. The current results seem inconclusive, especially when applied to the corpora of interest here: a written MSA corpus and a "cleaned up" version of a Levantine dialect corpus (see Section 2.1).

Beside the empirical study for inducing a translation lexicon, we also present a mathematical framework for inducing probabilities for a given translation lexicon from a pair of non-parallel, non-aligned corpora. At the moment of writing the empirical study of this method was rather limited. We list this method here for the sake of completeness, especially that a version of this method was used for parsing experiments during the JHU summer workshop (within the Treebank Transduction approach) and yielded improved results over a non-probabilistic translation lexicon.

## 3.2 Related Work

Similar experiments to build lexicons or obtain other information from comparable corpora have been performed, but none apply directly to our situation with the languages and resources we are trying to use, and none try to analyze what features of the corpora are important for these methods to work. Some try to create parallel corpora from comparable corpora; in (Fung and Cheung, 2004), parallel sentences are extracted from text that is usually unrelated on the document level in order to collect more data to the end of improving a bilingual lexicon. However, this is a bootstrapping method and therefore needs a seed lexicon to begin the algorithm for the purpose of computing lexical similarity scores, which involves problems related to the choice and availablility of a seed dictionary which will be discussed in more detail in the context of the methods ultimately used in the experiments. This method also requires large corpora that contain parallel sentences somewhere; if the corpora have no parallel sentences, we will not be able to find any. A method similar to this is used monolingually to find paraphrases (Barzilay, 2003). This could work bilingually with a seed lexicon, since

this method also computes lexical similarities between sentences, which is trivial when both corpora are in the same language. Another attempt at extracting parallel sentences (Munteanu et al., 2004) with the goal of improving a complete machine translation system uses unrelated government texts and news corpora, but the Maximum Entropy classifier this algorithm relies on requires parallel corpora from both domains of 5000 sentences each. This may seem small compared to some of the parallel corpora available, but for the language pair and domains we are interested in, having 5000 parallel sentences could possibly eliminate our problem and might provide enough training data to more accurately modify a MSA parser or POS tagger for a dialect.

Use of information on the internet has also been shown to be promising (Resnik and Smith, 2003) but again is not applicable for our language pair. The dialects are mainly spoken, and while there may be some web logs or informal websites written in the dialects, the methods that search the web for parallel texts typically search for pages that link to their own translation by looking for certain structures that indicate as such. Other methods use a lexicon to search for webpages that link to translated pages with a high lexical similarity. A method that might be interesting to try along these lines would be to search for comparable corpora, but it is difficult to qualify the degree of comparability, and gathering the comparable corpora automatically will likely introduce noise.

Other methods try to use a bridge language in order to build a dictionary instead of trying to extract one from parallel sentences or comparable corpora (Mann and Yarowsky, 2001; Schafer and Yarowsky, 2002). These methods typically try to build a lexicon between English and a target language that is closely related to a language which already has an established dictionary with English and contains words that have a small string edit distance from words in the target language– for example, if we have a lexicon from English to Spanish, we could induce a lexicon from English to Portugese by using the relationship between words in Spanish and words in Portugese. This is different from our problem in that we are not trying to build an English-dialect lexicon, and we would like as complete a lexicon as possible and not just the close cognate pairs.

The most relevant previous work to this particular problem is work done by Rapp (Rapp, 1999). This method entails counting co-occurrences of words for which we have no translation with words in a one-to-one seed dictionary. It relies upon the assumption that words in the two corpora that have similar co-occurrence distributions with the seed dictionary words are likely translations. The co-occurence frequency vectors are transformed into association vectors by computing the log likelihood of each co-occurrence count compared to the frequency of the current word and the current seed dictionary word and normalized so that these vectors sum to one and are therefore the same length. Vectors from words in one language are compared to vectors in the other language by matching the vector components according to the relation of the seed dictionary words and computing the city block distance between the two vectors. A list of candidate translations in order of smallest city distance is produced for each unknown word in both languages. Rapp used unrelated English and German news corpora with a seed dictionary of 16,000 entries. For evaluation, he held out 100 frequent words and was able to find an acceptable translation for 72%. A similar method (Diab and Finch, 2000) does not use a seed dictionary and computes similarity monolingually

first between the top (at most) 1000 frequent words, then compares all of these vectors to all of the possible vectors for the other language. This method uses the assumption that punctuation is similar between the two languages and uses a few punctuation marks as a kind of seed, but our Levantine corpus uses hardly any punctuation so a small seed dictionary would have to be used instead, much like Rapp's method. Another issue is that this method works well for large comparable corpora for the frequent words, and takes a long time due to all the comparisons, so we chose to work with Rapp's algorithm.

## 3.3 Our Approach

We use Rapp's algorithm, but we add a novel modification: after the candidate word lists for each language were calculated, we chose the word pair we were most confident about, added this pair to the seed dictionary, and repeated the process. In this way we expanded the seed dictionary and hoped to improve the results of the words not in the dictionary. We defined the word pair we were most confident about to be the word which had the largest difference between the city block distance of the first and second words in its candidate list. We did not address the issue of when to stop iterating, but set a threshold on how small the maximum difference could get based on optimal results from some preliminary trials. Other modifications from Rapp's setup include varying the window size and varying the minimum number of times a word must occur in order for the algorithm to analyze it. Some preliminary experiments indicated that the best window size for our purposes was four words on either side, and the minimum frequency depends on corpus size. We did not take word order into account as Rapp does, with the justification that there are known word order differences between Levantine and MSA that may affect the results; this aspect needs further investigation. Another difference is that the seed dictionary Rapp used was very large (about 16,000 entries) and his evaluation was done on 100 held out frequent words. We are interested in starting with a small seed dictionary (about 100 entries) in order to build a lexicon of as many words as possible. Translations that occur in both corpora are what we can attempt to find using this method. For example, if one corpus contains the word "cat" but the other corpus never mentions cats at all, since we are only using the information indicated by the co-occurrences in the particular corpora we have, in this case there is no way to find the translation for "cat". The results reported will be the number of words that the algorithm finds correctly out of the total number possible. The correct words will include entries in the original seed dictionary that are correct, the number of words added correctly to the dictionary, and the number of words whose first candidate in their list of possible translations is the correct one.

## 3.4 Experiments

The variations between corpora include content, genre (speech vs. text), and size. These affect the inherent similarity between the distributions of the words in the corpora, and the controlled experiments varying each of these independently will show

| Corpus 1 | Corpus 2 | Word Overlap | % found Corpus 1 | % found Corpus 2 |
|---|---|---|---|---|
| meetings* | briefings | 936 | 34.2 | 34.3 |
| meetings | briefings* | 936 | 34.6 | 34.5 |
| gigaword* | briefings | 1434 | 21.8 | 18.3 |
| gigaword | briefings* | 1434 | 20.2 | 19.2 |
| meetings* | gigaword | 758 | 17.8 | 16.1 |
| meetings | gigaword* | 758 | 13.9 | 13.6 |

Table 3.1: Full corpora results. An * indicates the corpus from which the seed dictionary was constructed.

how each influences the words that are possible to extract using this method. A parameter that affects the accuracy of the extraction of the possible words is the choice of seed dictionary.

Most of these experiments will be English-to-English, and then a few experiments involving Levantine and MSA will be discussed.

### 3.4.1 English Corpora

In order to facilitate direct evaluation and to allow for more control in the types of corpora used, we performed some English to English experiments with the extension of Rapp's method. The resulting extracted lexicons can be checked automatically that their entries match exactly. The corpora were chosen in order to provide variation across genre (speech vs. text) and content. The first corpus is a collection of meeting transcriptions by the ISCI. This is speech that includes partial sentences, disfluencies, and other speech effects, and the subject matter discussed is usually about natural language processing research. The second corpus is a collection of White House press briefing transcripts from 2002 downloaded from `http://www.whitehouse.gov`. This includes some statements that are read verbatim, but it is mostly spontaneous speech, albeit transcribed cleaner without many of the speech effects present in the meetings corpus. The topics are issues pertaining to United States politics. The third corpus is a collection of news articles from the AFP in the English Gigaword corpus. This is news text about a variety of subjects and is from December 2002 and has some articles about United States politics around the same time. A trivial extraction choosing sentences that contained either the words "president", "United States", "US", "UN", or "Iraq" was performed to try and capture the sentences discussing the same subjects. More sophisticated extraction methods will be explored later, but this created a corpus of similar size and content to the briefings corpus. Thus, with these choices of corpora, there is a variation in subject with genre close to constant (meetings and briefings), a variation in genre with subject close to constant (briefings and gigaword) and a variation in both subject and genre (meetings and gigaword). Each of the three corpora are about 4 MB in size.

### 3.4.2 Effect of Subject and Genre Similarity

Results between all pairs of corpora appear in Table 3.1. The table shows the word overlap, which is the number of words over the frequency threshold that appeared in both corpora and therefore are theoretically possible to extract. The frequency threshold for this set of experiments was 25 occurrences. The percentages reported for each corpus are the amount of words out of the word overlap that we have correct translations for. This number includes the number of words in the seed dictionary that appear in both corpora (since a word being in the seed dictionary does not necessarily have to have this property), the number of words added to the seed dictionary correctly, and the number of words in that corpora's list of words and their ten best possible translations whose correct translation is in the first candidate. While the meetings and briefings corpora did not have as many words in common as the briefings and the gigaword corpora, the extracted words from those was more accurate. This is most likely due to the genre commonality between the meetings and briefings corpora that would cause more of the frequent words to be similar and used in identifiable ways, such as talking in the first person and describing events currently happening. The content commonality between the briefings and gigaword corpora may have more words that are over the frequency threshold but are not extremely prevalent and are domain specific nouns and verbs that are used similarly and are hard to distinguish from each other. The meetings and gigaword corpora extraction has the least word overlap and performed the worst, understandably, since these corpora have differences in both genre and content.

### 3.4.3 Choice of Seed Dictionary

The choice of seed dictionary impacts the quality of the words added to the dictionary and the words not in the dictionary we are trying to translate. In the English-English case, any dictionary could be created since the translations are the same word, but in a case of another language pair we may be limited by what we have available and the number of additional translations we are able to obtain from a lexicographer. What we had available from previous experiments in the Levantine and MSA case is a list of closed class words and the top 100 most frequent words in the Levantine corpus. The top 100 most frequent words seem to hold the most information as far as co-occurrences with other words in the corpus, so this was the choice for the dictionary in the English-English case. The next decision is which corpus' top 100 words are chosen. Experiments varying only the choice of dictionary showed that this impacted the results most when the distribution of frequent words is very different between the two corpora. The numbers reported in Table 3.2 for each pair of corpora are the average differences in results from the first corpus and the second corpus when the seed dictionary is changed. The smallest difference occurs between the meetings and the briefings corpora, while the largest difference is between meetings and gigaword. The large difference between the meetings and gigaword corpora is partly due to the number of words in each seed dictionary that appear in the other corpus above the frequency threshold– 82 out of 100 words from the meetings seed dictionary appear in the gigaword corpus, while only 61 words from the gigaword dictionary appear in the meetings corpus. This shows that the high frequency words from the gigaword corpus are not similar to those in the meetings

| Corpus 1 | Corpus 2 | Avg difference |
|---|---|---|
| meetings | briefings | .3 |
| briefings | gigaword | 1.25 |
| gigaword | meetings | 3.2 |

Table 3.2: Differences in accuracy affected by dictionary choice, averaged between Corpus 1 and Corpus 2 results

corpus. This further reinforces that the meetings and briefings corpora have the most similarity in their distributions of words that are frequent enough for analysis, and that having similarity in genre makes more of an impact than similarity in content.

### 3.4.4 Effect of Corpus Size

Since the assumption is that we have available a small corpus in one language and a large corpus in the other, and factors such as difference in genre, and content impact the results, some sort of normalization of the corpora before applying this method of lexicon extraction would presumably improve accuracy. The gigaword corpus used throughout the paper so far was a trivial extraction attempting to extract sentences on the same subject of the meetings corpus, but this method may have missed similar sentences and kept irrelevant ones. Starting with the small briefings corpus on the order of the Levantine corpus's size and knowing that having a similar distribution in the most frequent words seems to be helpful, an extraction can be performed from the whole corpus of AFP documents from December 2002 to try to match the distribution of the small briefings corpus. Since this isn't possible to obtain exactly, a greedy search based on the frequencies of the top 100 words in the briefings corpus can be used. Since we assumed that the top 100 most frequent words are also our seed dictionary, this makes it possible to replicate these results with a language pair other than English-English by using the translated seed dictionary words to extract from the other language's corpus. This method is very greedy and will keep a sentence from the gigaword corpus if it contains at least one of these 100 words and the frequency of these 100 words in the new extracted gigaword corpus does not exceed the frequency of the words in the briefings corpus. This created a corpus about three times the size of the small briefings corpus, but this is much closer to normalizing for size than using the gigaword corpus from the previous experiments would be. The control for this experiment is a corpus similarly sized as the extracted gigaword corpus by taking the first section of the gigaword corpus blindly. These corpora had a similar word overlap amount with the small briefings corpus (355 words with the extracted corpus and 328 words with the control corpus, the frequency threshold is 20). An interesting side effect of this extraction method was that the percentage of sentences in the extracted corpus that contain quotation marks was 43%, while the percentage in the control corpus was 29%. This shows that the extraction process is making a corpus that is more similar to the briefings corpus that is mostly speech. The percentage of correct words found from the small briefings corpus and the extracted corpus were 44.5% and 47.0% respectively, while the results from the

| | Levantine | | MSA | |
|---|---|---|---|---|
| | top 1 | top 10 | top 1 | top 10 |
| No extraction | 3.1 | 28.1 | 3.7 | 33.3 |
| Extraction | 5.1 | 24.4 | 13.7 | 39.7 |

Table 3.3: Levantine and MSA results, threshold = 15

small briefings corpus and the control corpus were 41.8% and 42.4%. The correct seed dictionary additions portion of this number especially illustrated a difference; there were 20 correct words added using the extracted corpus and 9 correct words added using the control corpus. This shows that extracting sentences from a large corpus with the aim of creating a distribution similar to that of a small corpus normalizes some of the differences between the corpora in order to find the similarities at the word level easier since there is less noise.

### 3.4.5 Results on MSA and Levantine

With Levantine and MSA we had to make a few more modifications in order to test these results. The small lexicon we had available of the closed class words and the top 100 most frequent words from the Levantine corpus contained some many to many relations. These cause inconsistencies with the co-occurrence vectors since we cannot separate which counts should go to which instance of the same word that is related to more than one word. The seed dictionary used in these experiments was then just the one to one entries from the original small lexicon. The evaluation can no longer be direct since the correct translations are not known, but we can use the seed dictionary itself plus the many to many dictionaries as test cases and consider an entry correct if a word has one of its possible translations as the extracted one. In preliminary experiments, the iterations of adding words to the seed dictionary seemed to be adding all unrelated words. This suggests a problem with the metric by which a word pair is chosen for the dictionary. Because of this, the results for Levantine and MSA will be reported after one iteration, no words will be added to the dictionary, and the percentages reported are out of the word pairs in the entire lexicon we have available that occur over the frequency threshold in both corpora and have the correct translation appearing in the first candidate or in the top ten candidates. The first experiment uses the small Levantine corpus available to us and a similarly sized first portion of the MSA corpus, while the second experiment uses a similarly sized corpus extracted from the entire MSA corpus similarly to the method described above using the dictionary words for the extraction process. The results in Table 3.3 show that this extraction process also helps in this case, but the results are generally worse than English-English.

## 3.5 Discussion

One inherent problem with this method is the lack of word sense disambiguation. The word “work” appeared in both the meetings and the briefings corpus fairly frequently

yet was not correctly identified. The candidate translations for "work" from either side were unrelated words such as "hold" and "Congress". Further examination showed that "work" was used as both a noun and a verb almost equally in the meetings corpus while it was used almost exclusively as a verb in the briefings corpus. One attempt at tagging the instances of "work" in the meetings corpus as "work_n" and "work_v" improved the candidate list somewhat; the list for "work_v" contained more verbs such as "leave" and "hold" while the list for "work_n" now contained "spending" and "business", but neither of these got "work" from the briefings corpus and the briefings corpus "work" did not choose either of these. However, the part-of-speech in this case does not seem to be quite enough to separate the sense differences. The meetings corpus tends to use the verb "work" in the sense of something being possible or feasible– "this will work". The briefings corpus uses the verb "work" mostly in the sense of exertion towards an end– "The United States can work with other governments". It would be ideal to only use comparable corpora that would use the same sense of most words, but those may be difficult to find. Word sense differences may have to be handled by some other method.

None of our results equalled Rapp's 72% accuracy, but it is obvious that we are using much smaller corpora and seed dictionaries and evaluating on words that are less frequent. This method may simply not be suited exactly to this particular set of constraints on resources. Since there is less information available from the statistics from smaller corpora, different methods may need to be used to identify the most important components and indicators of word similarity.

## 3.6 Estimation of Probabilities for a Translation Lexicon

In the preceding discussion we aimed at extracting a translation lexicon from a given pair of corpora. Given such a translation lexicon, a probability may be assigned to every translation pair taking the statistics of the two corpora into account. The general problem of estimating the probabilities for a set of pairs from a pair of corpora is more general than the more known problem of aligning the sentences and words in a parallel corpus. In this work we present an Expectation-Maximization (EM) algorithm which assumes that the pair of corpora is a sample from a mixture (weighted interpolation) of two conditional distributions in which the words of the one corpus are generated given the words of the other corpus under the given translation lexicon.

Let $V$ and $V^{'}$ be two vocabularies. Let $C$ be a corpus of sentences over $V$, and $C^{'}$ another corpus of sentences over $V^{'}$. The corpora are assumed unrelated, but we assume we have a lexicon $L \subseteq (V \times V^{'})$. The problem we are faced with is to provide estimates of probabilities $P(v, v^{'})$ for all pairs $\langle v, v^{'} \rangle \in L$ using the corpora $C$ and $C^{'}$.

Generally speaking, and because the corpora are not related and cannot be expected to contain sentences that can be fully aligned, we cannot assume a sentence-level alignment (although one might expect some of the "phrases" from the one corpus to be translated into the other corpus, though possibly in different contexts). Hence, we will assume that there is no alignment possible between the sentences or phrases of the

two corpora (later on we might want to consider ways to relax this strong assumption somewhat).

Usually, for estimation of the joint probability for the lexicon pairs using Maximum-Likelihood we usually need a "complete" corpus, i.e. a corpus of pairs which will allow relative frequency estimates. In our situation, where the data is incomplete (we have only single-language corpora) we must use estimation methods that can deal with incomplete data, e.g. the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

For the EM algorithm to be applicable we have to commit to some model of the data, i.e. a set of parameters. The corpora will be assumed to constitute samples from this model. Given such a set of parameters, the EM algorithms starts with an intialization of the values of these parameters and proceeds to reestimate these values iteratively by two steps: (1) Expectation-step: create a complete corpus of pairs using the model expectations for the given corpora, and (2) Maximization-step: use the relative frequency estimate from the complete corpus to update the model parameters. In principle, the iterations stop when the parameter values (or log-likelihhood of corpus given model) converge.

One way to proceed with the EM algorithm is to use each corpus on its own to reestimate some model probabilities. The actual estimate is then taken to be the expectation value of the two final reestimates. This is essentially the approach taken in (Vogel et al., 2000) for a different problem of alignment between parallel corpora.

We take a different approach here. We assume a mixture model $P(.,.)$ which is equal to some weighted average of two models $P1(.,.)$ and $P2(.,.)$ specified next:

- Model 1: $P1(v,v^{'}) = P(v)\vec{P}_{(o)}(v^{'}|v)$
- Model 2: $P2(v,v^{'}) = P^{'}(v^{'})\vec{P}_{(o)}{}^{'}(v|v^{'})$

where $P(.)$ and $P^{'}(.)$ are relative frequency estimates of the prior probabilities, and $\tilde{P}_j(.\mid.)$ and $\tilde{P}_j{}^{'}(.\mid.)$, in this case ($j=0$), are two latent distributions that must also be estimated, and the subscript $(j)$ (for $j \geq 0$) in $\tilde{P}_{(j)}(.|.)$ and $\tilde{P}_{(j)}{}^{'}(.|.)$ is the iteration counter within the EM algorithm.

Our joint probability estimate will be the average of the two models whereby the the two models are weighted by the relative sizes of the two corpora to the total size of both corpora together:

$$P(v,v^{'}) = \frac{freq(v)}{N+N^{'}}\vec{P}_{(o)}(v^{'}|v) + \frac{freq(v^{'})}{N+N^{'}}\vec{P}_{(o)}{}^{'}(v|v^{'})$$

where $N$ and $N^{'}$ are the sizes of both corpora respectively (total count of all unigram events), and $freq(x)$ is the frequency of event $x$ in the two corpora. Note that $P(v)\frac{N}{N+N^{'}} = \frac{freq(v)}{N+N^{'}}$ and $P^{'}(v^{'})\frac{N^{'}}{N+N^{'}} = \frac{freq(v^{'})}{N+N^{'}}$.

Intuitively speaking, our algorithm starts with the Expectation-step under the assumptions that (1) the lexicon $L$ specifies all the non-zero probability of pairs of words,

and that (2) $\tilde{P}_{(0)}(. \mid .)$ and $\tilde{P}_{(0)}{}'(. \mid .)$ are initialized, for example, a uniform distribution over the relevant lexicon $L$ entries. This way, two complete corpora are created, one from $C$ using Model 1 and lexicon $L$ (in the one direction), and one from $C'$ using Model 2 and lexicon $L$ (in the other direction). Both complete corpora consist of pairs of a word and its translation with a conditional probability assigned by the models. Crucially, the two complete corpora are concatenated together and used for reestimation iteratively as specified next.

At every iteration $j$ the translation probabilities are re-estimated in two steps, whereby updating takes places:

1. The reestimation of the joint probabilities $P_{(j)}(.,.)$ is equal to the Maximum-Likelihood estimate (relative frequency) over the union of the two resulting corpora (by appending them), i.e.

$$P_{(j)}(v, v') = \frac{freq(v)}{N+N'}\tilde{P}_{(j-1)}(v'|v) + \frac{freq(v')}{N+N'}\vec{P}_{(j-1)}{}'(v|v')$$

2. Then the conditional probabilities $\tilde{P}_{(j)}(v'|v)$ and $\tilde{P}_{(j)}{}'(v|v')$ are updated as follows:

$$\tilde{P}_{(j)}(v'|v) = \frac{P_{(j)}(v, v')}{P(v)}$$

$$\tilde{P}_{(j)}{}'(v|v') = \frac{P_{(j)}(v, v')}{P'(v')}$$

The algorithm is terminated for the smallest $j > 0$ for which holds that the log-likelihood of the complete corpus under the model parameters converges.

Preliminary empirical results with this method on MSA-Levantine have been obtained by estimating the probabilities for a manually constructed translation lexicon from the Levantine corpus and the MSA corpus. The resulting probabilistic lexicon was employed within the Treebank Translation approach to parsing Levantine. Parsing using the induced probabilities gives better parsing results over the results of the same parser using the original non-probabilistic lexicon (see results reported in Section 5.2.3 and Section 5.3.2. A through empirical exploration of this method has not been completed.

## 3.7 Conclusions and future work

Extracting a bilingual lexicon automatically from comparable and unrelated corpora is a difficult task. The accuracy of the results depends upon the distribution of frequent words which is influenced most by the genre similarity of the corpora. Other factors such as the content and the sizes of the corpora have an impact on the results, and extraction from a large corpus to match the distribution of a small corpus can partially

normalize for these factors and improve accuracy. Results for the language pair of Levantine and MSA show promise but extracting a bilingual lexicon from corpora with these size and resource constraints may require different methods.

There are various modifications to the algorithm and setup of the experiments that could improve results. If one language has available resources such as part-of-speech taggers and parsers, using this information may give better results for lexicon extraction. This type of information could be used to determine which seed dictionary words are more important in the co-occurrence vectors for distinguishing between possible candidates. The less helpful seed dictionary words could be dropped from the vectors in the other language for comparison with this word, but this may eliminate negative correlation information. Another area of possible improvement is the use of iterations to add words to the seed dictionary. The metric by which a word pair is chosen to be added may not be the best. It may be possible to choose an optimal point at which the iterations should stop instead of fixing a threshold by doing some sort of evaluation after each addition to see whether it improves or worsens the results. More experimentation is needed to ascertain the effect these changes would have.

Another issue that should be explored is that of assigning probabilities in the case of many to many relations in the lexicon. One attempt could be the application of the expectation maximization algorithm described in section 3.6.

Application driven evaluation and application improvement through lexicon improvement is the long term goal for the results of this work. Part-of-speech tagging and parsing for Levantine using information about MSA are the primary applications and language pair, although exploring these techniques with languages that are less closely related is another area of interest for the future.

# Chapter 4

# Part-of-Speech Tagging

## 4.1 Introduction

Before discussing the more complex problem of adapting an MSA syntactic parser to process another Arabic dialect, it is useful to first consider the somewhat simpler task of porting a part-of-speech tagger from MSA to a dialect. Many of the major challenges encountered here are resonated later in the full parsing case.

In this chapter, we explore strategies in adapting a POS tagger that was trained to tag MSA sentences for tagging Levantine sentences. We have conducted experiments to address the following questions:

- **What is the tagging accuracy of an MSA POS tagger on Levantine data?** Since MSA and Levantine share many common words, it may seem plausible that an MSA tagger may perform adequately on Levantine sentences. Our results, however, suggest that this is not the case. We find that the tagger's accuracy on Levantine data is significantly worse than on MSA data. We present the relevant background information about the tagger in Section 4.2 and the details of this baseline experiment in Section 4.2.2.

- **What is the tagging accuracy of a tagger that is developed solely from (raw) Levantine data?** This is not an easy question to answer because there are different approaches to develop this kind of taggers and because it is difficult to make a fair comparison between this kind of taggers and the type of supervised taggers discussed above. We experimented with a tagger developed using the unsupervised learning method proposed by Clark (2001). We find that the tagging quality of this unsupervised tagger is not as good as supervised taggers.

- **What are important factors for adapting an MSA tagger for Levantine data?** We tackled the adaptation problem with different sources of information, including: basic linguistic knowledge about MSA and Levantine, varying sizes and qualities of MSA-Levantine lexicon, and a (small) tagged Levantine corpus. Our findings suggest that having a small but high-quality lexicon for the most com-

mon Levantine words results in the biggest "bang for the buck." The details of these studies are reported in Section 4.3.

Based on these experimental results, we believe that a better tagging accuracy can be achieved by allowing the model to make better use of our linguistic knowledge about the language. We propose a novel tagging model that supports an explicit representation of the root-template patterns of Arabic. Experimenting with this model is on-going work.

## 4.2 Preliminaries

Central to our discussion of adaptation strategies is the choice of representation for the POS tagging model. For the experiments reported in this chapter, we use the Hidden Markov Model (HMM) as the underlying representation for the POS tagger.HMM has been a popular choice of representation for POS tagging because because it affords reasonably good performance and because it is not too computationally complex. While slightly higher performance can be achieved by a POS tagger with a more sophisticated learning models such as Support Vector Machines (Diab et al., 2004a), we decided to work with HMMs so that we may gain a clearer understanding of the factors that influence the adaptation process.

To find the most likely tag sequence $\hat{T} = \{t_1, t_2, \ldots t_n\}$ for the input word sentence $\hat{W} = \{w_1, w_2, \ldots w_n\}$ using an HMM tagger, we need to compute:

$$\hat{T} = \arg\ \max_T P(T|W) = \arg\ \max_T P(W|T)P(T).$$

The transition probability $P(T)$ is typically approximated by an $N$-gram model:

$$P(T) \approx p(t_1)p(t_2|t_1) \prod_{i=3}^{n} p(t_i|t_{i-1}, \ldots, t_{i-N+1}).$$

That is, the distribution of tags for the ith word in the sentence only depends on the previous $N-1$ tags. For the experiments reported in this section, we used the bigram model (i.e., $N$=2) to reduce the complexity of the model. The observation probability $P(W|T)$ is computed by

$$P(W|T) \approx \prod_{i=1}^{n} p(w_i|t_i),$$

which assumes that the words are conditionally independent given its tag.

The collection of transition probabilities $p(t_i|t_j)$ and observation probabilities $p(w|t)$ forms the parameter set of the HMM. In order to develop a high quality POS tagger, we seek to estimate the parameters so that they accurately reflect the usage of the language. Under a supervised setting, the parameters are estimated based on training examples in which the sentences have been given their correct POS tags. Under an unsupervised setting, the parameters have to be estimated without knowing the correct POS tags for the sentences.

In considering a direct application of the MSA tagger to Levantine data (Section 4.2.2), we wish to ascertain to what extent are the parameter values estimated from

MSA data a good surrogate for the Levantine data. In taking an unsupervised learning approach, we try to determine how well the parameter can be estimated without knowledge of the correct POS tags. To adapt an MSA trained tagger for Levantine (Section 4.3), we want to find good strategies to modify the parameter values so that the model is reflective of the Levantine data.

### 4.2.1 Data

We use the same data set as the parsing experiments. The details of the data processing has already been described earlier in the report. Here we summarize those data statistics and characteristics that are the most relevant to POS tagging:

- We assume tagging takes place after tokenization; thus, the task is to assign a tag to every word delimited by white spaces.
- In all gold standards, we assume a reduced tag-set, commonly referred to as the Bies tag-set (Maamouri et al., 2003), that focuses on major parts of speech, exclusive of morphological information, number, and gender.
- The average sentence length of the MSA corpus is 33 words, whereas the average sentence length of the Levantine corpus is 6 words. The length difference is symptomatic of the more significant differences between the two corpora, as we have discussed earlier. Due to these differences, parameter values estimated from one corpus are probably not good estimations for a different corpus.
- A significant portion of the words in the Levantine corpus appeared in the MSA corpus (80% by tokens, 60% by types). However, the same orthographic representation does not always guarantee the same POS tag. Analysis of the gold standard shows that at least 6% of the overlapped words (by types) have no common tag assignments. Moreover, the frequency distribution of the tag assignments are usually very different.

### 4.2.2 Baseline: Direct Application of the MSA Tagger

As a first baseline, we consider the performance of a bigram tagger whose parameter values are estimated from annotated MSA data. When tested on new (previously unseen) MSA sentences, the tagger performs reasonably well, with a 93% accuracy. While this is not a state-of-the-art figure (which is in the high 90's), it shows that a bigram model forms an adequate representation for POS tagging and that there was enough annotated MSA data to train the parameters of the model.

On the Levantine development data set, however, the MSA tagger performs significantly worse, with an accuracy of 69% (the accuracy on the Levantine test data set is 64%). This shows that the parameter values estimated from the MSA are not a good match for the Levantine data, even though more than half of the words in the two corpora are common.

It is perhaps unsurprising that the accuracy suffered a decrease. The MSA corpus is much larger and more varied than the Levantine corpus. Many parameters of the

MSA tagger (especially the observation probabilities $p(w|t)$) are not useful in predicting tags of Levantine sentences; moreover, the model does not discriminate between different words that only appeared in the Levantine corpus. To reduce the effect of the MSA only words on the observation probabilities, we renormalized each observation distribution $P(W|t)$ so that the probability mass for those words that only appear in the MSA corpus is redistributed to the words that only appear in the Levantine corpus (proportional to each word's unigram frequency). We did not modify the parameters for the transition probabilities. The change brought about a slight improvement. The tagging accuracy of the development set went up to 70% (and up to 66% for the test set). The low accuracy rate suggests that more parameter re-estimations will be necessary to adapt the tagger for Levantine. We explore adaptation strategies in Section 4.3.

## 4.3 Adaptation

From the results of the baseline experiments, we hypothesize that it may be easier to re-estimate the parameter values of an MSA tagger for Levantine data by incorporating our available knowledge about MSA and Levantine and the relationship between them than to develop a Levantine tagger from scratch. In this study, we consider three possible information sources. One is to use general linguistic knowledge about the language to handle out-of-vocabulary words. For example, we know that when a word begins with the prefix *al*, it is more likely to be a noun. Another is to make use of a MSA-Levantine lexicon. Finally, although creating a large annotated training corpus for every dialect of interest is out of the question, it may be possible to have human experts to annotate a *very limited* number of sentences in the dialect of interest. To gain a better understanding of how different information sources affect the tagging model, we conducted a set of experiments to study the changes in the estimated parameter values as we incorporate these different types of information.

### 4.3.1 Basic Linguistic Knowledge

As our baseline studies indicate, the most unreliable part of the MSA tagger are its observation probabilities $p(w|t)$. We need to remove parameters representing words only used in MSA, but we also wish to estimate the observation probabilities for words that only appeared in Levantine. How much probability mass should be kept for the old estimates (for words common to both MSA and Levantine)? How much probability mass should be assigned to the newly introduced Levantine words?

One possibility is to reclaim all the probability mass from the MSA-only words and redistribute it to the Levantine words according to some heuristic. This could be problematic if a POS category lost most of its observation probability mass (due to generating many MSA-only words), since only a small portion of its distribution was estimated from observed (MSA) data.

It seems intuitive that the weighting should be category dependent. For closed-class categories (e.g., prepositions, conjunctions, participles), the estimates for the existing

words should be reliable and only a relatively small probability mass should be allotted to adding in new Levantine words; whereas for open-class categories (e.g., nouns, verbs, adverbs), we should allow for more probability mass to go to new words. We determine the weighting factor for each category by computing the portion of MSA-only words that category has. For instance, suppose the noun category contains 20% MSA-only words (by number of words, not by probability mass); we would renormalize the observation probability distribution so that 80% of the mass goes to words common to MSA-Levantine and 20% to estimate new Levantine words.

Next, we focus on heuristics for estimating new Levantine words. As mentioned in Section 4.2.2, estimating the parameters based on unigram probability of the words themselves is not very helpful. After all, we would not expect different part-of-speech tags to have the same kind of distribution of words. A somewhat better strategy is to allot probability mass to words according to the unigram probability distribution of the part-of-speech.[1] For example, most nouns have a relatively low unigram probability (even common nouns are not as frequent as closed-class words); therefore, if a word appears frequently in the Levantine corpus, its portion in the observation distribution $p(w|noun)$ ought to be small.

Comparing the unigram probability of an unknown word against the unigram probability distribution of each POS tag helps to differentiate between closed-class words and open-class words. Another similar kind of statistics is the distribution over the lengths of words for each POS tag. Closed-class categories such as prepositions and determiners typically have short words whereas nouns tend to have long words. Finally, a word's first and last few characters may also provide useful information. For example, many noun words begin with *al*. For each POS category, we build a probability distribution over its first two letters and last two letters.

Although these heuristics are rather crude, they help us modify the original MSA tagger to be more representative of Levantine data. The modified tagger has an accuracy rate of 73% on the development set, 70% on the test set, which is a 3% absolute increase in accuracy over the simple modification used for the baseline.

### 4.3.2 Knowledge about Lexical Mappings

A problem with the reclaim-and-redistribute strategy described in the previous section is that some MSA words are represented differently under Levantine. For example, a participal in MSA is represented as *lA* but as *mA* in Levantine. Without knowing the *translation* of the MSA word in Levantine, we would not be able to take advantage of the observation probability parameters related to that word. Moreover, by reclaiming these probability masses, we introduce unnecessary uncertainties in the distributions.

As we have mentioned earlier, however, lexicon development is a challenging task in and of itself; therefore, it may be unrealistic to expect a very complete lexicon. In this section, our experiments used two lexicons, both rely on manual processing. One is a small developed dictionary that contains MSA translations for closed-class words as well as 100 frequent Levantine words ( 300 words combined). Another is a larger

[1] This distribution has to be built from MSA data since we are not assuming that any tagged Levantine data is available.

lexicon that contains MSA translations for most of the words in the development set ( 1800 words).

Given a lexicon, we can directly transfer the observation probabilities for MSA words that have Levantine translations to the new Levantine words. Then, the probability mass of MSA words that have no Levantine mappings are reclaimed and redistributed to Levantine words that have no MSA mappings as before. Because many errors arise due to the mis-tagging of closed-class words, the information from the small lexicon was extremely helpful (increasing the accuracy of the development set to 80%, and that of the test set to 77%). In contrast, having the larger lexicon did not bring forth significant further improvements. The accuracy of the test set increased to 78%. Because the larger lexicon was constructed based on the development data, it does not necessarily have a good coverage of the words used in the test data.

### 4.3.3 Knowledge about Levantine POS Tagging

A third source of information is manually tagged data in the dialect of interest, which can be used as training examples. If a sufficient quantity of data can be tagged, we could apply a straight-forward supervised learning technique to train a dialect tagger directly (same method as training the MSA tagger). However, as we have repeatedly emphasized, it is impractical to expect the availability of this kind of data. Therefore, in this section we focus on whether having a *limited* number of tagged data would be useful.

First, we establish a qualitative reference point by training a Levantine tagger with all the available data from the development set (about 2000 sentences or 11K words). This fully supervised Levantine tagger has a tagging accuracy of 80% on the data from the test set. Note that the accuracy is lower compared to training and testing on MSA data. This is due to the small size of the training data (the MSA training set is more than ten times as large). We hypothesize that having the MSA tagger as a starting point can compensate for the lack of tagged Levantine training examples. Specifically, we assume that a human expert is willing to label 100 Levantine sentences for us. We set the limit to 100 sentences because it seems like an amount that a person can accomplish reasonably quickly (within a week) and because the number of tagged words will be comparable in size to the smaller lexicon used in the previous subsection ( 300 words).

To address this problem, we consider two factors. First, which 100 sentences should be tagged (so that the tagger's accuracy would improve the most)? In the experiments, we take a greedy approach to find the top 100 sentences that contain the most number of unknown words. Second, how should the tagged data be used? For instance, we could try to extract a single tagging model out of two separately trained Levantine tagger and the MSA tagger, following a method proposed by Xi and Hwa (2005). However, we would also like to include the other information sources (e.g., the lexicon), which makes model merging more difficult. Instead, we simply take the adapted MSA tagging model as an initial parameter estimate and retrain it with the tagged Levantine data.

We find that the addition of 100 tagged Levantine sentences is helpful in adapting an MSA tagger for Levantine data. Retraining the MSA tagger that has already been adapted with the reclaim-and-redistribute method results in an additional 8% increase in accuracy, to 78%, which is close to the performance of the supervised Levantine

| | No lexicon | Small lexicon | Large lexicon |
|---|---|---|---|
| Naive Adaptation | 67% | NA | NA |
| +Minimal Knowledge | 70% | 77% | 78% |
| +Manual Tagging | 78% | 80% | 79% |

Table 4.1: Tagging accuracy of the adapted MSA tagger for the test data. As points of reference, the MSA tagger without adaptation has an accuracy of 64%; A supervised Levantine tagger (trained on 11K words) has an accuracy of 80%.

tagger trained from 2000 sentences. Starting from the adapted MSA tagger that used the small lexicon further improves the performance to 80%. The relatively small rate of improvement suggests that information to be gained from the lexicon and manual tagging duplicate each other. We argue that it may be more worthwhile to develop the lexicon since it can be used in a number of ways, not just for POS tagging.

## 4.4 Summary and Future Work

In summary, our experimental results based on adapting an MSA POS-tagger for Levantine data suggest that leveraging from existing resources is a viable option. We considered three factors that might influence adaptation: whether we have some general knowledge about the languages, whether we have a translation lexicon between MSA and the dialect, and whether we have any manually tagged data in the dialect. The results summarized in Table 4.1 suggest that the most useful information source is a small lexicon of frequent words. Combining the information from the small lexicon and a parameter renormalization strategy based on minimal linguistic knowledge, we see the biggest improvement in the tagger. Since the results are approaching the accuracy of a supervised method, we hypothesize that better tagging accuracy can be achieved by allowing the tagging model to exploit knowledge about the language.

# Chapter 5

# Parsing

## 5.1 Related Work

There has been a fair amount of interest in parsing one language using another language, see for example (Smith and Smith, 2004; Hwa et al., 2004) for recent work. Much of this work uses synchronized formalisms as do we in the grammar transduction approach. However, these approaches rely on parallel corpora. For MSA and its dialects, there are no naturally occurring parallel corpora. It is this fact that has led us to investigate the use of explicit linguistic knowledge to complement machine learning.

We refer to additional relevant work in the appropriate sections.

## 5.2 Sentence Transduction

### 5.2.1 Introduction

The basic idea behind this approach is to parse an MSA translation of the LA sentence and then link the LA sentence to the MSA parse. Machine translation (MT) is not easy, especially when there are no MT resources available such as naturally occurring parallel text or transfer lexicons. However, for this task we have three encouraging insights. First, for really close languages it is possible to obtain better translation quality by means of simpler methods (Hajic et al., 2000). Second, suboptimal MSA output can still be helpful for the parsing task without necessarily being fluent or accurate (since our goal is parsing LA, not translating it to MSA). And finally, translation from LA to MSA is easier than from MSA to LA. This is a result of the availability of abundant resources for MSA as compared to LA: for example, text corpora and tree banks for language modeling and a morphological generation system (Habash, 2004).

One disadvantage of this approach is the lack of structural information on the LA side for translation from LA to MSA, which means that we are limited in the techniques we can use. Another disadvantage is that the translation can add more ambiguity to the parsing problem. Some unambiguous dialect words can become syntactically ambiguous in MSA. For example, the LA words من *mn* 'from' and مين *myn* 'who' both are

| | No Tags | Gold Tags |
|---|---|---|
| **Baseline** | 59.4/51.9/55.4 | 64.0/58.3/61.0 |
| **S-LEX-UN** | 63.8/58.3/61.0 | 67.5/63.4/65.3 |
| **B-LEX-UN** | 65.3/61.1/63.1 | 66.8/63.2/65.0 |

Figure 5.1: Results on DEV (labeled precision/recall/F-measure)

| | No Tags | Gold Tags |
|---|---|---|
| **Baseline** | 53.5 | 60.2 |
| **Small LEX** | 57.7 | 64.0 |

Figure 5.2: Results on TEST (labeled F-measure)

translated into an orthographically ambiguous form in MSA من *mn* 'from' or 'who'.

### 5.2.2 Implementation

Each word in the LA sentence is translated into a bag of MSA words, producing a sausage lattice. The lattice is scored and decoded using the SRILM toolkit with a trigram language model trained on 54 million MSA words from Arabic Gigaword (Graff, 2003). The text used for language modeling was tokenized to match the tokenization of the Arabic used in the ATB and LATB. The tokenization was done using the ASVM Toolkit (Diab et al., 2004b). The 1-best path in the lattice is passed on to the Bikel parser (Bikel, 2002), which was trained on the MSA training ATB. Finally, the terminal nodes in the resulting parse structure are replaced with the original LA words.

### 5.2.3 Experimental Results

Table 5.1 describes the results of the sentence transduction path on the development corpus (DEV) in different settings: using no POS tags versus gold tags, and using S-LEX-UN versus B-LEX-UN. (We will include significance results in the final paper.) Additionally, the baseline results for parsing the LA sentence directly using the MSA parser are included for comparison (with and without gold POS tags). The results are reported in terms of PARSEVAL's Precision/Recall/F-Measure.

Using S-LEX-UN improves the F1 score for no tags and for gold tags. A further improvement is gained when using the B-LEX-UN in the case of the no tags, but this contribution is reverted in the case of gold tags. We suspect that the added translation ambiguity from B-LEX-UN is responsible for the drop.

In Figure 5.2, we report the F-Measure score on the test set (TEST) for the baseline and for S-LEX-UN (with and without gold POS tags). We see a general drop in performance between DEV and TEST for all combinations suggesting that TEST is a harder set to parse than DEV.

### 5.2.4 Discussion

The current implementation does not handle cases where the word order changes between MSA and LA. Since we start from an LA string, identifying constituents to permute is clearly a hard task. We experimented with identifying strings with the postverbal LA negative particle *$* and then permuting them to obtain the MSA preverbal order. The original word positions are "bread-crumbed" through the system's language modeling and parsing steps and then used to construct an unordered dependency parse tree labeled with the input LA words. (A constituency representation is meaningless since word order changes from LA to MSA.) The results were not encouraging since the effect of the positive changes was undermined by new errors introduced. We also experimented with the S-LEX-EM and B-LEX-EM lexicons. There was no consistent improvement gained.

## 5.3 Treebank Transduction

In this approach, the idea is to convert the ATB-Train into an LA-like treebank using linguistic knowledge of the systematic variations on the syntactic, lexical and morphological levels across the two varieties of Arabic. We then train a statistical parser on the newly transduced treebank and test the parsing performance against the gold test set of the LA treebank sentences.

### 5.3.1 MSA Transformations

We now list the transformations we applied to ATB-Train

#### Structural Transformations

*Consistency checks (CON):* These are conversions that make the ATB annotation more consistent. For example, there are many cases where *SBAR* and *S* nodes are used interchangeably in the MSA treebank. Therefore, an *S* clause headed by a complementizer is converted to an *SBAR*.

*Fragmentation (FRAG):* Due to genre differences between the MSA and LA data, the LA treebank sentences frequently have *SBAR, SQ, NP, PP, etc* as root nodes. In an attempt to bridge the genre difference and mimic the fragment distribution in the LA treebank, we fragment the MSA treebank by extracting sentence fragments and rendering them as independent sentences while keeping the source sentences intact.

*Sentence Splitting (TOPS):* A fair number of sentences in the ATB have a root node *S* with several embedded direct descendant *S* nodes, sometimes conjoined using the conjunction *w*. We split such sentences into several shorter sentences.

#### Syntactic Transformations

There are several possible systematic syntactic transformations. We focus on three major ones due to their significant distributional variation in MSA and LA. They are highlighted in Figure 1.1.

*Negation (NEG):* In MSA negation is marked with preverbal negative particles. In LA, a negative construction is expressed in one of three possible ways: *m$*/*mA* preceding the verb; a particle *$* suffixed onto the verb; or a circumfix of a prefix *mA* and suffix it $. See Figure 1.1 for an example of the *$* suffix. We converted all negation instances in the ATB-Train three ways reflecting the LA constructions for negation.

*VSO-SVO Ordering (SVO):* Both Verb Subject Object (VSO) and Subject Verb Object (SVO) constructions occur in MSA and LA treebanks. But pure VSO constructions – where there is no pro-drop – occur in the LA corpus only 10% of the data, while VSO is the most frequent ordering in MSA. Hence, the goal is to skew the distributions of the SVO constructions in the MSA data. Therefore, VSO constructions are replicated and converted to SVO constructions.

*Demonstrative Switching (DEM):* In LA, demonstrative pronouns precede or, more commonly, follow the nouns they modify, while in MSA demonstrative pronoun only precede the noun they modify. Accordingly, we replicate the LA constructions in ATB-Train and moved the demonstrative pronouns to follow their modified nouns while retaining the source MSA ordering simultaneously.

#### Lexical Substitution

We use the four lexicons described in Section 2. These resources are created with a coverage bias from LA to MSA. As an approximation, we reversed the directionality to yield MSA to LA lexical retaining the assigned probability scores. Manipulations involving lexical substitution are applied only to the lexical items without altering the POS tag or syntactic structure.

#### Morphological Transformations

We applied some morphological rules to handle specific constructions in the LA. The POS tier as well as the lexical items were affected by these manipulations.

*bd Construction (BD)*: *bd* is an LA noun that means *want*. It acts like a verb in verbal constructions yielding VP constructions headed by NN. It is typically followed by an enclitic possessive pronoun. Accordingly, we translated all the verbs meaning *want*/*need* into the noun *bd* and changed their respective POS tag to NN. In cases where the subject of the MSA verb is pro-dropped, we add a clitic possessive pronoun in the first or second person singular. This was intended to bridge the genre and domain disparity between the MSA and LA data.

*Aspectual Marker b (ASP):* In dialectal Arabic, present tense verbs are marked with an initial *b*. Therefore we add a *b* prefix to all verbs of POS tag type VBP. The aspectual marker is present on the verb *byHbw* in the LA example in Figure 1.1.

### 5.3.2 Evaluation

**Tools**: We use the Bikel parser for syntactic parsing.
**Data**: For training we use the transformed ATB-Train. We report results on gold POS tagged DEV and TEST using the Parseval metrics of labeled precision, labeled recall and f-measure.

| Condition | Gold Tags |
|---|---|
| *Baseline* | 63.0/57.5/60.1 |
| **STRUCT** | 64.6/59.2/61.8 |
| **NEG** | 64.5/58.9/61.6 |
| **STRUCT+NEG** | 64.6/59.5/62 |
| **S-LEX-EM** | 64/58.6/61.2 |
| **MORPH** | 63.9/58/60.8 |
| **S-LEX-EM+MORPH** | 63.9/58.3/61 |
| **STRUCT+NEG+MORPH** | 64.6/59.5/62 |
| **STRUCT+NEG+S-LEX-EM** | 65.4/60.9/**63.1** |
| **STRC+NEG+S-LEX-EM+MORPH** | 65.1/60.3/62.6 |

Figure 5.3: Results on DEV(labeled precision/recall/F-measure)

| | Gold Tags |
|---|---|
| Baseline | 60.2 |
| STRC+NEG+S-LEX-EM | 61.5 |

Figure 5.4: Results on TEST (labeled F-measure)

**Results**: Table 5.3 illustrates the results on the LA development set.

In Table 5.3, **STRUCT** is a combination of *CON* and *TOPS*. *FRAG* does not yield performance improvement. Of the Syntactic transformations applied, **NEG** is the only transformation that helped performance. Both *SVO* and *DEM* decrease the performance from the baseline with F-measures of 59.4 and 59.5, respectively. Of the lexical substitutions, S-LEX-EM helped performance the best. **MORPH** refers to a combination of the *BD* and it ASP transformations. As illustrated in the table, the best results obtained are those from combining **STRUCT** with **NEG** and **S-LEX-EM** yielding a 8.1% error reduction on DEV. Table 5.4 illustrate the results obtained on TEST. We see an overall reduction in the performance indicating that the test data is very different from the training data. However overall, we see similar trends to those observed with DEV where the best conditions on DEV are the best conditions on TEST.

The best condition **STRC+NEG+S-LEX-EM** shows an error reduction of 3.4%.

**Discussion**: The best performing condition always include *CON*, *TOPS* and *NEG*. *S-LEX-EM* helped a little however, due to the inherent directionality of the resource, its impact is limited. We experimented with the other lexicons but none of them helped improve performance. We believe that the EM probabilities helped in biasing the lexical choices in lieu of an LA language model. We do not observe any significant improvement from applying **MORPH**.

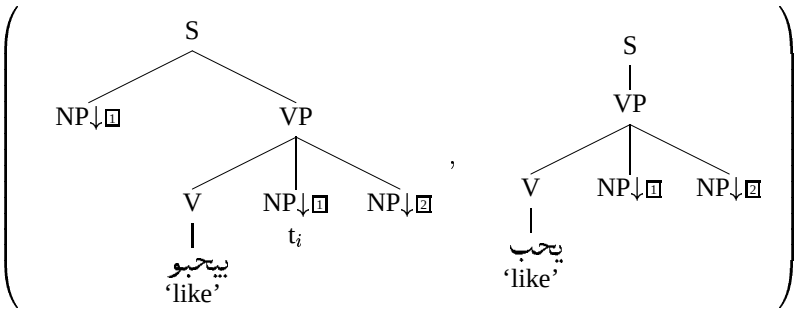


Figure 5.5: Example elementary tree pair of a synchronous TSG.

## 5.4 Grammar Transduction

The grammar-transduction approach uses the machinery of synchronous grammars to relate MSA and LA. A synchronous grammar composes paired *elementary trees*, or fragments of phrase-structure trees, to generate pairs of phrase-structure trees. In the present application, we start with MSA elementary trees (plus probabilities) induced from the ATB and transform them using handwritten rules into dialect elementary trees to yield an MSA-dialect synchronous grammar. This synchronous grammar can be used to parse new dialect sentences using statistics gathered from the MSA data.

Thus this approach can be thought of as a variant of the treebank-transduction approach in which the syntactic transformations are localized to elementary trees. Moreover, because a parsed MSA translation is produced as a byproduct, we can also think of this approach as being related to the sentence-transduction approach.

### 5.4.1 Preliminaries

The parsing model used is essentially that of Chiang (Chiang, 2000), which is based on a highly restricted version of tree-adjoining grammar. In its present form, the formalism is tree-substitution grammar (Schabes, 1990) with an additional operation called *sister-adjunction* (Rambow et al., 2001). Because of space constraints, we omit discussion of the sister-adjunction operation in this paper.

A tree-substitution grammar is a set of elementary trees. A frontier node labeled with a nonterminal label is called a *substitution site*. If an elementary tree has exactly one terminal symbol, that symbol is called its *lexical anchor*.

A derivation starts with an elementary tree and proceeds by a series of composition operations. In the substitution operation, a substitution site is rewritten with an elementary tree with a matching root label. The final product is a tree with no more substitution sites.

A *synchronous* TSG is a set of pairs of elementary trees. In each pair, there is a one-to-one correspondence between the substitution sites of the two trees, which we represent using boxed indices (Figure 5.5). The substitution operation then rewrites a pair of coindexed substitution sites with an elementary tree pair. A *stochastic* synchronous

TSG adds probabilities to the substitution operation: the probability of substituting an elementary tree pair $\langle \alpha, \alpha' \rangle$ at a substitution site pair $\langle \eta, \eta' \rangle$ is $P(\alpha, \alpha' \mid \eta, \eta')$.

When we parse a monolingual sentence $S$ using one side of a stochastic synchronous TSG, using a straightforward generalization of the CKY and Viterbi algorithms, we obtain the highest-probability paired derivation which includes a parse for $S$ on one side, and a parsed translation of $S$ on the other side. It is also straightforward to calculate inside and outside probabilities for re-estimation by Expectation-Maximization (EM).

### 5.4.2 An MSA-dialect synchronous grammar

We now describe how we build our MSA-dialect synchronous grammar. As mentioned above, the MSA side of the grammar is extracted from the ATB in a process described by Chiang and others (Chiang, 2000; Xia et al., 2000; Chen, 2001). This process also gives us MSA-only substitution probabilities $P(\alpha \mid \eta)$.

We then apply various transformation rules (described below) to the MSA elementary trees to produce a dialect grammar, at the same time assigning probabilities $P(\alpha' \mid \alpha)$. The synchronous-substitution probabilities can then be estimated as:

$$\begin{aligned} P(\alpha, \alpha' \mid \eta, \eta') &\approx P(\alpha \mid \eta) P(\alpha' \mid \alpha) \\ &\approx P(\alpha \mid \eta) P(w', t' \mid w, t) \\ & \quad P(\bar{\alpha}' \mid \bar{\alpha}, w', t', w, t) \end{aligned}$$

where $w$ and $t$ are the lexical anchor of $\alpha$ and its POS tag, and $\bar{\alpha}$ is the equivalence class of $\alpha$ modulo lexical anchors and their POS tags.

$P(w', t' \mid w, t)$ is assigned as described in Section 2; $P(\bar{\alpha}' \mid \bar{\alpha}, w', t', w, t)$ is initially assigned by hand. Because the full probability table for the latter would be quite large, we smooth it using a backoff model so that the number of parameters to be chosen is manageable. Finally, we reestimate these parameters using EM.

Because of the underlying syntactic similarity between the two varieties of Arabic, we assume that every tree in the MSA grammar extracted from the MSA treebank is also an LA tree. In addition, we perform certain tree transformations on all elementary trees which match the pattern: NEG and SVO (Section 5.3.1) and BD (Section 5.3.1). NEG is modified so that we simply insert a *$* negation marker postverbally, as the preverbal markers are handled by MSA trees.

### 5.4.3 Experimental Results

We first use DEV to determine which of the transformations are useful. The results are shown in Figure 5.6. We see that important improvements are obtained using lexicon S-LEX-UN. Adding the SVO transformation does not improve the results, but the NEG and BD transformations help slightly, and their effect is (partly) cumulative. (We did not perform these tuning experiments on input with no POS tags.)

| | No Tags | Gold Tags |
|---|---|---|
| **Baseline** | 57.6/53.5/55.5 | 63.9/62.5/63.2 |
| **S-LEX-UN** | 63.0/60.8/61.9 | 66.9/67.0/66.9 |
| + **SVO** | | 66.9/66.7/66.8 |
| + **NEG** | | 67.0/67.0/67.0 |
| + **BD** | | 67.4/67.0/67.2 |
| + **NEG** + **BD** | | 67.4/67.1/67.3 |
| **B-LEX-UN** | 64.9/63.7/64.3 | 67.9/67.4/67.6 |

Figure 5.6: Results on development corpus (labeled precision/recall/F-measure)

| | No Tags | Gold Tags |
|---|---|---|
| **Baseline** | 53.0 | 63.3 |
| **Small LEX** + **Neg** + **bd** | 60.2 | 67.1 |

Figure 5.7: Results on TEST (labeled F-measure)

### 5.4.4 Discussion

We observe that the lexicon can be used effectively in our synchronous grammar framework. In addition, some syntactic transformations are useful. The SVO transformation, we assume, turned out not to be useful because the SVO word order is also possible in MSA, so that the new trees were not needed and needlessly introduced new derivations. The BD transformation shows the importance not of general syntactic transformations, but rather of lexically specific syntactic transformations: varieties within one language family may differ more in terms of the lexico-syntactic constructions used for a specific (semantic or pragmatic) purpose than in their basic syntactic inventory. Note that our tree-based synchronous formalism is ideally suited for expressing such transformations since it is lexicalized, and has an extended domain of locality.

# Chapter 6

# Summary of Results and Discussion

## 6.1 Results on Parsing

We have built three frameworks for leveraging MSA corpora and explicit knowledge about the lexical, morphological, and syntactic differences between MSA and LA for parsing LA. The results on TEST are summarized in Figure 6.1, where performance is given as absolute and relative reduction in labeled F-measure error (i.e., $100 - F$).[1] We see that some important improvements in parsing quality can be achieved. We also remind the reader that on the ATB, state-of-the-art performance is currently about 75% F-measure.

There are several important ways in which we can expand our work. For the sentence-transduction approach, we plan to explore the use of a larger set of permutations; to use improved language models on MSA (such as language models built on genres closer to speech); to use lattice parsing (Sima'an, 2000) directly on the translation lattice and to integrate this approach with the treebank transduction approach. For the treebank and grammar transduction approaches, we would like to explore more

[1]The baselines for the three approaches have been slightly different, due to the use of different parsers and different tokenizations. It is for this reason that we choose to compare the results using error reduction.

| | **No Tags** | **Gold Tags** |
|---|---|---|
| **Sentence Transd.** | 4.2/9.0% | 3.8/9.5% |
| **Treebank Transd.** | | 1.3/3.2% |
| **Grammar Transd.** | 7.2/15.3% | 3.8/10.4% |

Figure 6.1: Results on test corpus: absolute/percent error reduction in F-measure over baseline (using MSA parser on LA test corpus); all numbers are for best obtained results using that method

systematic syntactic, morphological, and lexico-syntactic transformations. We would also like to explore the feasibility of inducing the syntactic and morphological transformations automatically. Specifically for the treebank transduction approach, it would be interesting to apply an LA language model for the lexical substitution phase as a means of pruning out implausible word sequences.

For all three approaches, one major impediment to obtaining better results is the disparity in genre and domain which affects the overall performance. This may be bridged by finding MSA data that is more in the domain of the LA test corpus than the MSA treebank.

# Bibliography


Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, I. Dan Melamed, Franz-Josef Och, David Purdy, Noah A. Smith, and David Yarowsky. 1999. Statistical Machine Translation. Technical report, JHU. http://citeseer.nj.nec.com/al-onaizan99statistical.html.

Regina Barzilay. 2003. *Information Fusion for Multidocument Summarization: Paraphrasing and Generation*. Ph.D. thesis, Columbia University.

Daniel M. Bikel. 2002. Design of a multi-lingual, parallel-processing statistical parsing engine. In *Proceedings of International Conference on Human Language Technology Research (HLT)*.

Peter F. Brown, John Cocke, Stephen A. DellaPietra, Vincent J. DellaPietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June.

John Chen. 2001. *Towards Efficient Statistical Parsing Using Lexicalized Grammatical Information*. Ph.D. thesis, University of Delaware.

David Chiang. 2000. Statistical parsing with an automatically-extracted tree adjoining grammar. In *38th Meeting of the Association for Computational Linguistics (ACL'00)*, pages 456–463, Hong Kong, China.

Alexander Clark. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proc. of the 5th Computational Natural Language Learning Conference*.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(B):1–38.

Mona Diab and Steven Finch. 2000. A statistical word level translation model for comparable corpora. In *Proceedings of Conference on Content Based Multimeda Information Access RIAO '00*, Paris, France.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004a. Automatic tagging of arabic text: From raw text to base phrase chunks. In *5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, Boston, MA.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004b. Automatic tagging of arabic text: From raw text to base phrase chunks. In *5th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL04)*, Boston, MA.

Pascale Fung and Percy Cheung. 2004. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain, July.

Pascale Fung and Kathleen McKeown. 1997. Finding terminology translations from non-parallel corpora. In *Proceedings of the Fifth Workshop on Very Large Corpora*, pages 192–202, Aug.

Pascale Fung. 1995. A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 236–243.

David Graff. 2003. Arabic Gigaword, LDC Catalog No.: LDC2003T12. Linguistic Data Consortium, University of Pennsylvania.

Nizar Habash. 2004. Large scale lexeme based arabic morphological generation. In *Proceedings of Traitement Automatique du Langage Naturel (TALN-04)*. Fez, Morocco.

Jan Hajic, Jan Hric, and Vladislav Kubon. 2000. Machine Translation of very close languages. pages 7–12, Seattle.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2004. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*.

Mohamed Maamouri, Ann Bies, Hubert Jin, and Tim Buckwalter. 2003. Arabic treebank: Part 1 v 2.0. Distributed by the Linguistic Data Consortium. LDC Catalog No.: LDC2003T06.

Mohamed Maamouri, Ann Bies, and Tim Buckwalter. 2004. The penn arabic treebank : Building a large-scale annotated arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Gideon S. Mann and David Yarowsky. 2001. Multipath translation lexicon induction via bridge languages. In *Proceedings of the Second Meeting of the North American Association for Computational Linguistics*, Pittsburgh, PA, June.

I. Dan Melamed. 2000. Models of translational equivalence among words. *Computational Linguistics*, 26(2):221–249, June.

Dragos Stefan Munteanu, Alexander Fraser, and Daniel Marcu. 2004. Improved machine translation performance via parallel sentence extraction from comparable corpora. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*.

Owen Rambow, K. Vijay-Shanker, and David Weir. 2001. D-Tree Substitution Grammars. *Computational Linguistics*, 27(1).

Reinhard Rapp. 1999. Automatic identification of word translations from unrelated english and german corpora. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 519–526, College Park, MD.

Philip Resnik and Noah A. Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3).

Yves Schabes. 1990. *Mathematical and Computational Aspects of Lexicalized Grammars*. Ph.D. thesis, Department of Computer and Information Science, University of Pennsylvania.

Charles Schafer and David Yarowsky. 2002. Inducing translatioin lexicons via diverse similarity measures and bridge languages. In *Proceedings of the Conference on Computational Natural Language Learning*, Taipei, Taiwan.

Khalil Sima'an. 2000. Tree-gram parsing: Lexical dependencies and structural relations. In *Proceedings of 38th Annual Meeting of the Association for Computational Linguistics (ACL'00)*, Hong Kong, China.

David A. Smith and Noah A. Smith. 2004. Bilingual parsing with factored estimation: Using english to parse korean. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP04)*.

S. Vogel, F. Och, C. Tillmann, S. Nieen, H. Sawaf, and H. Ney. 2000. Statistical methods for machine translation. In Wolfgang Wahlster, editor, *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer Verlag, Berlin.

Chenhai Xi and Rebecca Hwa. 2005. A backoff model for bootstrapping resources for non-english languages. In *Proc. of HLT/EMNLP-05*, Oct.

Fei Xia, Martha Palmer, and Aravind Joshi. 2000. A uniform method of grammar extraction and its applications. In *Proc. of the EMNLP 2000*, Hong Kong.