Parsing Arabic Dialects

David Chiang[*], **Mona Diab**[†], **Nizar Habash**[†], **Owen Rambow**[†], **Safiullah Shareef**[‡]
∗ ISI, University of Southern California
† CCLS, Columbia University
‡ The Johns Hopkins University
chiang@isi.edu, {mdiab,habash,rambow}@cs.columbia.edu, safi@jhu.edu

Abstract

The Arabic language is a collection of spoken dialects with important phonological, morphological, lexical, and syntactic differences, along with a standard written language, Modern Standard Arabic (MSA). Since the spoken dialects are not officially written, it is very costly to obtain adequate corpora to use for training dialect NLP tools such as parsers. In this paper, we address the problem of parsing transcribed spoken Levantine Arabic (LA). We do not assume the existence of any annotated LA corpus (except for development and testing), nor of a parallel corpus LA-MSA. Instead, we use explicit knowledge about the relation between LA and MSA.

1 Introduction: Arabic Dialects

The Arabic language is a collection of spoken dialects and a standard written language.[1] 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 not a native language of any Arabic speaking people, i.e., children do not learn it from their parents but in school. Most native speakers of Arabic are unable to produce sustained spontaneous MSA. Dialects vary not only along a geographical continuum but also with other sociolinguistic variables such as the urban/rural/Bedouin dimension.

The multidialectal situation has important negative consequences for Arabic natural language processing (NLP): since the spoken dialects are not officially written and do not have standard orthography, it is very costly to obtain adequate corpora, even unannotated corpora, to use for training NLP tools such as parsers. 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.[2] 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 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 be argued that it is easier to create corpora than to encode all this knowledge explicitly. In response, we claim 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

[1] This paper is based on work done at the 2005 Johns Hopkins Summer Workshop, which was partially supported by the National Science Foundation under Grant No. 0121285. Diab, Habash, and Rambow were supported for additional work by DARPA contract HR0011-06-C-0023 under the GALE program. We wish to thank audiences at JHU for their useful feedback. The authors are listed in alphabetical order.

[2] We exclude from this study part-of-speech (POS) tagging and LA/MSA lexicon induction. See (Rambow et al., 2005) for these issues, as well as for more details on parsing.

resources is feasible; we do not provide a demonstration of cost-effectiveness.

The paper is organized as follows. After discussing related work and available corpora, we present linguistic issues in LA and MSA (Section 4). 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); treebank transduction, in which the MSA treebank is turned into an LA treebank (Section 6); and grammar transduction, in which an MSA grammar is turned into an LA grammar which is then used for parsing LA (Section 7). We summarize and discuss the results in Section 8.

2 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.

3 Linguistic Resources

We use the MSA treebanks 1, 2 and 3 (ATB) from the LDC (Maamouri et al., 2004). We split the corpus into 10% development data, 80% training data and 10% test data all respecting document boundaries. The training data (ATB-Train) comprises 17,617 sentences and 588,244 tokens.

The Levantine treebank LATB (Maamouri et al., 2006) comprises 33,000 words of treebanked conversational telephone transcripts collected as part of the LDC CALL HOME project. The treebanked section is primarily in the Jordanian subdialect of LA. 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 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). 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.

Two lexicons were created: 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 assign to the mappings in the two lexicons both uniform probabilities and biased probabilities using Expectation Maximization (EM; see (Rambow et al., 2005) for details of the use of EM). We thus have four different lexicons: Small lexicon with uniform probabilities (SLXUN); Small Lexicon with EM-based probabilities (SLXEM); Big Lexicon with uniform probabilities (BLXUN); and Big Lexicon with EM-based probabilities (BLXEM).

4 Linguistic Facts

We illustrate the differences between LA and MSA using an example[3]:

(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$gl* in LA but العمل *AlEml* in MSA. In contrast, the word for ‘men’ is the same in both LA and MSA: الرجال *AlrjAl*. There are typically also differences in function words, in our example ش *$* (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

[3] Arabic Examples are transliterated using the Buckwalter scheme: http://www.ldc.upenn.edu/myl/morph/buckwalter.html

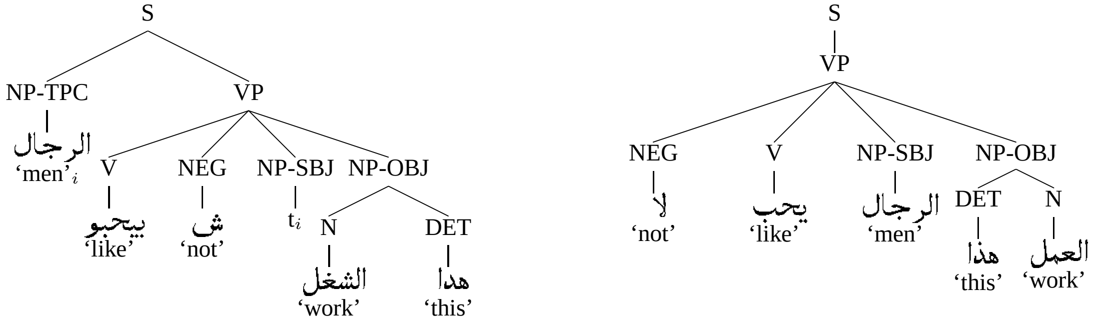

Figure 1: LDC-style left-to-right phrase structure trees for LA (left) and MSA (right) for sentence (1)

-*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 in LA, but precedes it in MSA. Finally, we see that the negation marker follows the verb in LA, while it precedes the verb in MSA.[4] The two phrase structure trees are shown in Figure 1 in the LDC convention. Unlike the phrase structure trees, the (unordered) dependency trees for the MSA and LA sentences (not shown here for space considerations) are isomorphic. They differ only in the node labels.

5 Sentence Transduction

In this approach, we 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 translated into an orthographically ambiguous form in MSA من *mn* 'from' or 'who'.

5.1 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., 2004). 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 Experimental Results

Table 1 describes the results of the sentence transduction path on the development corpus (DEV) in different settings: using no POS tags in the input versus using gold POS tags in the input, and using SLXUN versus BLXUN. The baseline results are obtained by parsing the LA sentence directly using the MSA parser (with and without gold POS tags). The results are reported in terms of PARSEVAL's

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

	No Tags	Gold Tags
Baseline	59.4/51.9/55.4	64.0/58.3/61.0
SLXUN	63.8/58.3/61.0	67.5/63.4/65.3
BLXUN	65.3/61.1/63.1	66.8/63.2/65.0

Table 1: Sentence transduction results on DEV (labeled precision/recall/F-measure)

	No Tags	Gold Tags
Baseline	53.5	60.2
SLXUN	57.7	64.0

Table 2: Sentence transduction results on TEST (labeled F-measure)

Precision/Recall/F-Measure.

Using SLXUN improves the F1 score for no tags and for gold tags. A further improvement is gained when using the BLXUN lexicon with no POS tags in the input, but this improvement disappears when we use BLXUN with gold POS tags. We suspect that the added translation ambiguity from BLXUN is responsible for the drop. We also experimented with the SLXEM and BLXEM lexicons. There was no consistent improvement.

In Table 2, we report the F-Measure score on the test set (TEST) for the baseline and for SLXUN (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.

5.3 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 systems 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 newly introduced errors.

6 Treebank Transduction

In this approach, the idea is to convert the MSA treebank (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.

6.1 MSA Transformations

We now list the transformations we applied to ATB-Train:

6.1.1 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*.

*Sentence Splitting (***TOPS***):* A fair number of sentences in the ATB has 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.

6.1.2 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 illustrated in Figure 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 $. 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 both 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.

6.1.3 Lexical Substitution

We use the four lexicons described in Section 3. These resources are created with a coverage bias from LA to MSA. As an approximation, we reversed the directionality to yield MSA to LA lexicons, retaining the assigned probability scores. Manipulations involving lexical substitution are applied only to the lexical items without altering the POS tag or syntactic structure.

6.1.4 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 a possessive pronoun. Accordingly, we translated all MSA verbs meaning *want*/*need* into the noun *bd* and changed their POS tag to the nominal tag 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.

*lys Construction (***LYS***):* In the MSA data, *lys* is interchangeably marked as a verb and as a particle. However, in the LA data, *lys* occurs only as a particle. Therefore, we convert all occurrences of *lys* into RP.

6.2 Experimental Results

We transform ATB-Train into an LA-like treebank using different strategies, and then train the Bikel parser on the resulting LA-like treebank. We parse the LA test set with the Bikel parser trained in this manner. As before, we report results on DEV and TEST sets, without POS tags and with gold POS tags, using the Parseval metrics of labeled precision, labeled recall and f-measure. Table 3 summarizes the results on the LA development set.

	No Tags	**Gold Tags**
Baseline	59.5/52/55.5	64.2/58.4/61.1
MORPH		63.9/58/60.8
SLXEM		64.2/59.3/61.7
NEG		64.5/58.9/61.6
STRUCT		64.6/59.2/61.8
+NEG		64.6/59.5/62
+NEG **+SLXEM**	62.1/55.9/**58.8**	65.5/61.3/**63.3**

Table 3: Treebank transduction results on DEV(labeled precision/recall/F-measure)

	No Tags	**Gold Tags**
Baseline	53.5	60.2
STRUCT **+NEG+SLXEM**	57	62.1

Table 4: Treebank transduction results on TEST (labeled F-measure)

In Table 3, **STRUCT** refers to the structural transformations combining **TOPS** with **CON**. Of the Syntactic transformations applied, **NEG** is the only transformation that helps 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 (i.e., lexicons), **SLXEM** helps performance the best. **MORPH** refers to a combination of all the morphological transformations. **MORPH** does not help performance, as we see a decrease from the baseline by 0.3% when applied on its own. When combining **MORPH** with other conditions, we see a consistent decrease. For instance, **STRUCT+NEG+SLXEM+MORPH** yields an f-measure of 62.9 compared to 63.3 yielded by **STRUCT+NEG+SLXEM**. The best results obtained are those from combining **STRUCT** with **NEG** and **SLXEM** for both the No Tag and Gold Tag conditions.

Table 4 shows the results obtained on TEST. As for the sentence transduction case, we see an overall reduction in the performance indicating that the test data is very different from the training data.

6.3 Discussion

The best performing condition always includes **CON**, **TOPS** and **NEG**. **SLXEM** helps as well, 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, playing the role of an LA language model (which we do not have). We do not observe any significant improvement from applying **MORPH**.

7 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.

7.1 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 2). 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).

7.2 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 α and its POS tag, and $\bar{\alpha}$ is the equivalence class of α modulo lexical anchors and their POS tags.

$P(w', t' \mid w, t)$ is assigned as described in Section 3; $P(\bar{\alpha}' \mid \bar{\alpha}, w', t', w, t)$ 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

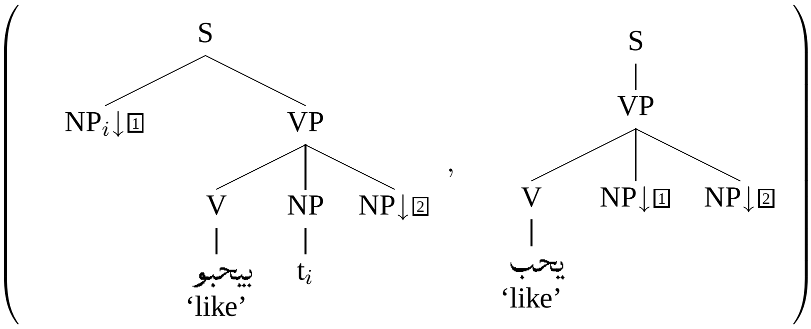

Figure 2: Example elementary tree pair of a synchronous TSG.

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 6.1.2) and **BD** (Section 6.1.4). **NEG** is modified so that we simply insert a *$* negation marker postverbally, as the preverbal markers are handled by MSA trees.

7.3 Experimental Results

We first use DEV to determine which of the transformations are useful. The results are shown in Table 5. The baseline is the same as in the previous two approaches. We see that important improvements are obtained using lexicon SLXUN. 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.) We also experimented with the SLXEM and BLXEM lexicons. There was no consistent improvement.

7.4 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.

	No Tags	**Gold Tags**
Baseline	59.4/51.9/55.4	64.0/58.3/61.0
SLXUN	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
BLXUN	64.9/63.7/64.3	67.9/67.4/67.6

Table 5: Grammar transduction results on development corpus (labeled precision/recall/F-measure)

	No Tags	**Gold Tags**
Baseline	53.5	60.2
SLXUN + **Neg** + **bd**	60.2	67.1

Table 6: Grammar transduction results on TEST (labeled F-measure)

8 Summary of Results and Discussion

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 Table 7, where performance is given as absolute and relative reduction in labeled F-measure error (i.e., $100 - F$). We see that some important improvements in parsing

	No Tags	**Gold Tags**
Sentence Transd.	4.2/9.0%	3.8/9.5%
Treebank Transd.	3.5/7.5%	1.9/4.8%
Grammar Transd.	6.7/14.4%	6.9/17.3%

Table 7: 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

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 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.

References

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).*

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.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. 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.

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. In *6th Applied Natural Language Processing Conference (ANLP'2000)*, 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, 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.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, and Dalila Tabessi. 2006. Developing and using a pilot dialectal Arabic treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC'06*, page to appear, Genoa, Italy.

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

Owen Rambow, David Chiang, Mona Diab, Nizar Habash, Rebecca Hwa, Khalil Sima'an, Vincent Lacey, Roger Levy, Carol Nichols, and Safi ullah Shareef. 2005. Parsing arabic dialects. Final Report, 2005 JHU Summer Workshop.

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

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)*.

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.