Better Arabic Parsing: Baselines, Evaluations, and Analysis

Spence Green and Christopher D. Manning
Computer Science Department, Stanford University
{spenceg,manning}@stanford.edu

Abstract

In this paper, we offer broad insight into the underperformance of Arabic constituency parsing by analyzing the interplay of linguistic phenomena, annotation choices, and model design. First, we identify sources of syntactic ambiguity understudied in the existing parsing literature. Second, we show that although the Penn Arabic Treebank is similar to other treebanks in gross statistical terms, annotation consistency remains problematic. Third, we develop a human interpretable grammar that is competitive with a latent variable PCFG. Fourth, we show how to build better models for three different parsers. Finally, we show that in application settings, the absence of gold segmentation lowers parsing performance by 2–5% F1.

1 Introduction

It is well-known that constituency parsing models designed for English often do not generalize easily to other languages and treebanks.[1] Explanations for this phenomenon have included the relative informativeness of lexicalization (Dubey and Keller, 2003; Arun and Keller, 2005), insensitivity to morphology (Cowan and Collins, 2005; Tsarfaty and Sima'an, 2008), and the effect of variable word order (Collins et al., 1999). Certainly these linguistic factors increase the difficulty of syntactic disambiguation. Less frequently studied is the interplay among language, annotation choices, and parsing model design (Levy and Manning, 2003; Kübler, 2005).

[1]The apparent difficulty of adapting constituency models to non-configurational languages has been one motivation for dependency representations (Hajič and Zemánek, 2004; Habash and Roth, 2009).

To investigate the influence of these factors, we analyze Modern Standard Arabic (henceforth MSA, or simply "Arabic") because of the unusual opportunity it presents for comparison to English parsing results. The Penn Arabic Treebank (ATB) syntactic guidelines (Maamouri et al., 2004) were purposefully borrowed without major modification from English (Marcus et al., 1993). Further, Maamouri and Bies (2004) argued that the English guidelines generalize well to other languages. But Arabic contains a variety of linguistic phenomena unseen in English. Crucially, the conventional orthographic form of MSA text is *unvocalized*, a property that results in a deficient graphical representation. For humans, this characteristic can impede the acquisition of literacy. How do additional ambiguities caused by devocalization affect statistical learning? How should the absence of vowels and syntactic markers influence annotation choices and grammar development? Motivated by these questions, we significantly raise baselines for three existing parsing models through better grammar engineering.

Our analysis begins with a description of syntactic ambiguity in unvocalized MSA text (§2). Next we show that the ATB is similar to other treebanks in gross statistical terms, but that annotation consistency remains low relative to English (§3). We then use linguistic and annotation insights to develop a manually annotated grammar for Arabic (§4). To facilitate comparison with previous work, we exhaustively evaluate this grammar and two other parsing models when gold segmentation is assumed (§5). Finally, we provide a realistic evaluation in which segmentation is performed both in a pipeline and jointly with parsing (§6). We quantify error categories in both evaluation settings. To our knowledge, ours is the first analysis of this kind for Arabic parsing.

2 Syntactic Ambiguity in Arabic

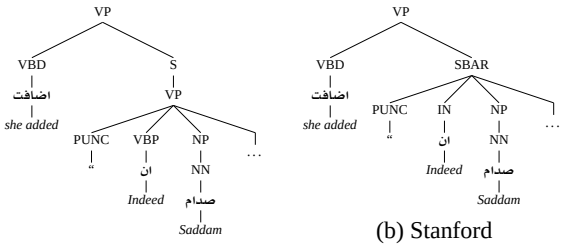
Arabic is a morphologically rich language with a root-and-pattern system similar to other Semitic languages. The basic word order is VSO, but SVO, VOS, and VO configurations are also possible.[2] Nouns and verbs are created by selecting a consonantal root (usually triliteral or quadriliteral), which bears the semantic core, and adding affixes and diacritics. Particles are uninflected. Diacritics can also be used to specify grammatical relations such as case and gender. But diacritics are not present in unvocalized text, which is the standard form of, e.g., news media documents.[3]

Let us consider an example of ambiguity caused by devocalization. Table 1 shows four words whose unvocalized surface forms **ان** *an* are indistinguishable. Whereas Arabic linguistic theory assigns (1) and (2) to the class of pseudo verbs **إن وأخواتها** *inna and her sisters* since they can be inflected, the ATB conventions treat (2) as a complementizer, which means that it must be the head of SBAR. Because these two words have identical complements, syntax rules are typically unhelpful for distinguishing between them. This is especially true in the case of quotations—which are common in the ATB—where (1) will follow a verb like (2) (Figure 1).

Even with vocalization, there are linguistic categories that are difficult to identify without semantic clues. Two common cases are the attributive adjective and the process nominal **المصدر** *maSdar*, which can have a verbal reading.[4] Attributive adjectives are hard because they are orthographically identical to nominals; they are inflected for gender, number, case, and definiteness. Moreover, they are used as substantives much more frequently than is done in English.

	Word	Head Of	Complement	POS
1	إنّ *inna* “Indeed, truly”	VP	Noun	VBP
2	أنّ *anna* “That”	SBAR	Noun	IN
3	إن *in* “If”	SBAR	Verb	IN
4	أن *an* “to”	SBAR	Verb	IN

Table 1: Diacritized particles and pseudo-verbs that, after orthographic normalization, have the equivalent surface form ان *an*. The distinctions in the ATB are linguistically justified, but complicate parsing. Table 8a shows that the best model recovers SBAR at only 71.0% F1.



(a) Reference (b) Stanford

Figure 1: The Stanford parser (Klein and Manning, 2002) is unable to recover the verbal reading of the unvocalized surface form ان *an* (Table 1).

Process nominals name the action of the transitive or ditransitive verb from which they derive. The verbal reading arises when the *maSdar* has an NP argument which, in vocalized text, is marked in the accusative case. When the *maSdar* lacks a determiner, the constituent as a whole resembles the ubiquitous annexation construct **الإضافة** *iDafa*. Gabbard and Kulick (2008) show that there is significant attachment ambiguity associated with *iDafa*, which occurs in 84.3% of the trees in our development set. Figure 4 shows a constituent headed by a process nominal with an embedded adjective phrase. All three models evaluated in this paper incorrectly analyze the constituent as *iDafa*; none of the models attach the attributive adjectives properly.

For parsing, the most challenging form of ambiguity occurs at the discourse level. A defining characteristic of MSA is the prevalence of *discourse markers* to connect and subordinate words and phrases (Ryding, 2005). Instead of offsetting new topics with punctuation, writers of MSA insert connectives such as **و** *wa* and **ف** *fa* to link new elements to both preceding clauses and the text as a whole. As a result, Arabic sentences are usually long relative to English, especially after

[2]Unlike machine translation, constituency parsing is not significantly affected by variable word order. However, when grammatical relations like subject and object are evaluated, parsing performance drops considerably (Green et al., 2009). In particular, the decision to represent arguments in verb-initial clauses as VP internal makes VSO and VOS configurations difficult to distinguish. Topicalization of NP subjects in SVO configurations causes confusion with VO (pro-drop).

[3]Techniques for automatic vocalization have been studied (Zitouni et al., 2006; Habash and Rambow, 2007). However, the data sparsity induced by vocalization makes it difficult to train statistical models on corpora of the size of the ATB, so vocalizing and then parsing may well not help performance.

[4]Traditional Arabic linguistic theory treats both of these types as subcategories of noun **الاسم**.

Length	English (WSJ)	Arabic (ATB)
≤ 20	41.9%	33.7%
≤ 40	**92.4%**	73.2%
≤ 63	99.7%	**92.6%**
≤ 70	99.9%	94.9%

Table 2: Frequency distribution for sentence lengths in the WSJ (sections 2–23) and the ATB (p1–3). English parsing evaluations usually report results on sentences up to length 40. Arabic sentences of up to length 63 would need to be evaluated to account for the same fraction of the data. We propose a limit of 70 words for Arabic parsing evaluations.

	Part of Speech	Tag	Freq.
و *wa* "and"	conjunction	CC	4256
	preposition	IN	6
	abbreviation	NN	6
ف *fa* "so, then"	conjunction	CC	160
	connective particle	RP	67
	abbreviation	NN	22
	response conditioning particle	RP	11
	subordinating conjunction	IN	3

Table 3: Dev set frequencies for the two most significant discourse markers in Arabic are skewed toward analysis as a conjunction.

segmentation (Table 2). The ATB gives several different analyses to these words to indicate different types of coordination. But it conflates the coordinating and discourse separator functions of *wa* (واو العطف) into one analysis: conjunction (Table 3). A better approach would be to distinguish between these cases, possibly by drawing on the vast linguistic work on Arabic connectives (Al-Batal, 1990). We show that noun-noun vs. discourse-level coordination ambiguity in Arabic is a significant source of parsing errors (Table 8c).

3 Treebank Comparison

3.1 Gross Statistics

Linguistic intuitions like those in the previous section inform language-specific annotation choices. The resulting structural differences between treebanks can account for relative differences in parsing performance. We compared the ATB[5] to treebanks for Chinese (CTB6), German (Negra), and English (WSJ) (Table 4). The ATB is disadvantaged by having fewer trees with longer average

	ATB	CTB6	Negra	WSJ
Trees	23449	28278	20602	**43948**
Word Types	40972	45245	51272	**46348**
Tokens	738654	782541	355096	**1046829**
Tags	32	34	**499**	45
Phrasal Categories	22	26	**325**	27
Test OOV	16.8%	22.2%	**30.5%**	13.2%
Per Sentence				
Depth (μ / σ^2)	3.87 / 0.74	**5.01 / 1.44**	3.58 / 0.89	4.18 / 0.74
Breadth (μ / σ^2)	**14.6 / 7.31**	10.2 / 4.44	7.50 / 4.56	12.1 / 4.65
Length (μ / σ^2)	**31.5 / 22.0**	27.7 / 18.9	17.2 / 10.9	23.8 / 11.2
Constituents (μ)	**32.8**	32.5	8.29	19.6
μ Const. / μ Length	1.04	**1.18**	0.482	0.820

Table 4: Gross statistics for several different treebanks. Test set OOV rate is computed using the following splits: ATB (Chiang et al., 2006); CTB6 (Huang and Harper, 2009); Negra (Dubey and Keller, 2003); English, sections 2-21 (train) and section 23 (test).

yields.[6] But to its great advantage, it has a high ratio of non-terminals/terminals (μ Constituents / μ Length). Evalb, the standard parsing metric, is biased *toward* such corpora (Sampson and Babarczy, 2003). Also surprising is the low test set OOV rate given the possibility of morphological variation in Arabic. In general, several gross corpus statistics favor the ATB, so other factors must contribute to parsing underperformance.

3.2 Inter-annotator Agreement

Annotation consistency is important in any supervised learning task. In the initial release of the ATB, inter-annotator agreement was inferior to other LDC treebanks (Maamouri et al., 2008). To improve agreement during the revision process, a dual-blind evaluation was performed in which 10% of the data was annotated by independent teams. Maamouri et al. (2008) reported agreement between the teams (measured with Evalb) at 93.8% F1, the level of the CTB. But Rehbein and van Genabith (2007) showed that Evalb should not be used as an indication of real difference—or similarity—between treebanks.

Instead, we extend the *variation n-gram* method of Dickinson (2005) to compare annotation error rates in the WSJ and ATB. For a corpus C, let M be the set of tuples $\langle n, l \rangle$, where n is an n-gram with bracketing label l. If any n appears

[5] LDC A-E catalog numbers: LDC2008E61 (ATBp1v4), LDC2008E62 (ATBp2v3), and LDC2008E22 (ATBp3v3.1). We map the ATB morphological analyses to the shortened "Bies" tags for all experiments.

[6] Generative parsing performance is known to deteriorate with sentence length. As a result, Habash et al. (2006) developed a technique for splitting and chunking long sentences. In application settings, this may be a profitable strategy.

	Corpus		Sample	Error %	
	Trees	Nuclei	n-grams	Type	n-gram
WSJ 2–23	43948	25041	746	12.0%	**2.10%**
ATB	23449	20292	2100	**37.0%**	1.76%

Table 5: Evaluation of 100 randomly sampled variation nuclei types. The samples from each corpus were independently evaluated. The ATB has a much higher fraction of nuclei per tree, and a higher type-level error rate.

in a corpus position without a bracketing label, then we also add $\langle n, \texttt{NIL} \rangle$ to M. We call the set of unique n-grams with multiple labels in M the *variation nuclei* of C.

Bracketing variation can result from either annotation errors or linguistic ambiguity. Human evaluation is one way to distinguish between the two cases. Following Dickinson (2005), we randomly sampled 100 variation nuclei from each corpus and evaluated each sample for the presence of an annotation error. The human evaluators were a non-native, fluent Arabic speaker (the first author) for the ATB and a native English speaker for the WSJ.[7]

Table 5 shows type- and token-level error rates for each corpus. The 95% confidence intervals for type-level errors are (5580, 9440) for the ATB and (1400, 4610) for the WSJ. The results clearly indicate increased variation in the ATB relative to the WSJ, but care should be taken in assessing the magnitude of the difference. On the one hand, the type-level error rate is not calibrated for the number of n-grams in the sample. At the same time, the n-gram error rate is sensitive to samples with extreme n-gram counts. For example, one of the ATB samples was the determiner ذلك *dhalik* "that." The sample occurred in 1507 corpus positions, and we found that the annotations were consistent. If we remove this sample from the evaluation, then the ATB type-level error rises to only 37.4% while the n-gram error rate increases to 6.24%. The number of ATB n-grams also falls below the WSJ sample size as the largest WSJ sample appeared in only 162 corpus positions.

[7] Unlike Dickinson (2005), we strip traces and only consider POS tags when pre-terminals are the only intervening nodes between the nucleus and its bracketing (e.g., unaries, base NPs). Since our objective is to compare distributions of bracketing discrepancies, we do not use heuristics to prune the set of nuclei.

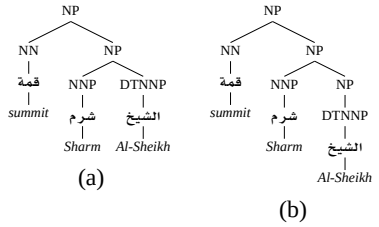


Figure 2: An ATB sample from the human evaluation. The ATB annotation guidelines specify that proper nouns should be specified with a flat NP (a). But the city name *Sharm Al-Sheikh* is also *iDafa*, hence the possibility for the incorrect annotation in (b).

4 Grammar Development

We can use the preceding linguistic and annotation insights to build a manually annotated Arabic grammar in the manner of Klein and Manning (2003). Manual annotation results in human interpretable grammars that can inform future treebank annotation decisions. A simple lexicalized PCFG with second order Markovization gives relatively poor performance: 75.95% F1 on the test set.[8] But this figure is surprisingly competitive with a recent state-of-the-art baseline (Table 7).

In our grammar, features are realized as annotations to basic category labels. We start with noun features since written Arabic contains a very high proportion of NPs. **genitiveMark** indicates recursive NPs with a indefinite nominal left daughter and an NP right daughter. This is the form of recursive levels in *iDafa* constructs. We also add an annotation for one-level *iDafa* (**oneLevelIdafa**) constructs since they make up more than 75% of the *iDafa* NPs in the ATB (Gabbard and Kulick, 2008). For all other recursive NPs, we add a common annotation to the POS tag of the head (**recursiveNPHead**).

Base NPs are the other significant category of nominal phrases. **markBaseNP** indicates these non-recursive nominal phrases. This feature includes named entities, which the ATB marks with a flat NP node dominating an arbitrary number of NNP pre-terminal daughters (Figure 2).

For verbs we add two features. First we mark any node that dominates (at any level) a verb

[8] We use head-finding rules specified by a native speaker of Arabic. This PCFG is incorporated into the Stanford Parser, a factored model that chooses a 1-best parse from the product of constituency and dependency parses.

Feature	States	Tags	F1	Indiv. ΔF1
—	3208	33	76.86	—
recursiveNPHead	3287	38	77.46	+0.60
genitiveMark	3471	38	77.88	+0.42
splitPUNC	4221	47	77.98	+0.10
markContainsVerb	5766	47	79.16	+1.18
markBaseNP	6586	47	79.5	+0.34
markOneLevelIdafa	7202	47	79.83	+0.33
splitIN	7595	94	80.48	+0.65
containsSVO	9188	94	80.66	+0.18
splitCC	9492	124	80.87	+0.21
markFem	10049	141	80.95	+0.08

Table 6: Incremental dev set results for the manually annotated grammar (sentences of length ≤ 70).

phrase (**markContainsVerb**). This feature has a linguistic justification. Historically, Arabic grammar has identified two sentences types: those that begin with a nominal (الجملة الاسمية), and those that begin with a verb (الجملة الفعلية). But foreign learners are often surprised by the verbless predications that are frequently used in Arabic. Although these are technically nominal, they have become known as "equational" sentences. **markContainsVerb** is especially effective for distinguishing root S nodes of equational sentences. We also mark all nodes that dominate an SVO configuration (**containsSVO**). In MSA, SVO usually appears in non-matrix clauses.

Lexicalizing several POS tags improves performance. **splitIN** captures the verb/preposition idioms that are widespread in Arabic. Although this feature helps, we encounter one consequence of variable word order. Unlike the WSJ corpus which has a high frequency of rules like VP $\rightarrow$ VB PP, Arabic verb phrases usually have lexicalized intervening nodes (e.g., NP subjects and direct objects). For example, we might have VP $\rightarrow$ VB NP PP, where the NP is the subject. This annotation choice weakens **splitIN**.

The ATB gives all punctuation a single tag. For parsing, this is a mistake, especially in the case of interrogatives. **splitPUNC** restores the convention of the WSJ. We also mark all tags that dominate a word with the feminine ending ة *taa marbuuTa* (**markFeminine**).

To differentiate between the coordinating and discourse separator functions of conjunctions (Table 3), we mark each CC with the label of its right sister (**splitCC**). The intuition here is that the role of a discourse marker can usually be determined by the category of the word that follows it. Because conjunctions are elevated in the parse trees when they separate recursive constituents, we choose the right sister instead of the category of the next word. We create equivalence classes for verb, noun, and adjective POS categories.

5 Standard Parsing Experiments

We compare the manually annotated grammar, which we incorporate into the Stanford parser, to both the Berkeley (Petrov et al., 2006) and Bikel (Bikel, 2004) parsers. All experiments use ATB parts 1–3 divided according to the canonical split suggested by Chiang et al. (2006). Preprocessing the raw trees improves parsing performance considerably.[9] We first discard all trees dominated by X, which indicates errors and non-linguistic text. At the phrasal level, we remove all function tags and traces. We also collapse unary chains with identical basic categories like NP $\rightarrow$ NP. The preterminal morphological analyses are mapped to the shortened "Bies" tags provided with the treebank. Finally, we add "DT" to the tags for definite nouns and adjectives (Kulick et al., 2006).

The orthographic normalization strategy we use is simple.[10] In addition to removing all diacritics, we strip instances of *taTweel* تطويل, collapse variants of *alif* ا to bare *alif*,[11] and map Arabic punctuation characters to their Latin equivalents. We retain segmentation markers—which are consistent only in the vocalized section of the treebank—to differentiate between e.g. هم "they" and هم+ "their." Because we use the vocalized section, we must remove null pronoun markers.

In Table 7 we give results for several evaluation metrics. Evalb is a Java re-implementation of the standard labeled precision/recall metric.[12]

[9] Both the corpus split and pre-processing code are available at http://nlp.stanford.edu/projects/arabic.shtml.

[10] Other orthographic normalization schemes have been suggested for Arabic (Habash and Sadat, 2006), but we observe negligible parsing performance differences between these and the simple scheme used in this evaluation.

[11] *taTweel* (ـ) is an elongation character used in Arabic script to justify text. It has no syntactic function. Variants of *alif* are inconsistently used in Arabic texts. For *alif* with *hamza*, normalization can be seen as another level of devocalization.

[12] For English, our Evalb implementation is identical to the most recent reference (EVALB20080701). For Arabic we

Model	System	Length	Leaf Ancestor			Evalb			Tag
			Corpus	Sent	Exact	LP	LR	F1	%
Stanford (v1.6.3)	Baseline	70	0.791	0.825	**358**	80.37	79.36	79.86	95.58
		all	0.773	0.818	358	78.92	77.72	78.32	95.49
	GoldPOS	70	0.802	0.836	452	81.07	80.27	80.67	99.95
Bikel (v1.2)	Baseline (Self-tag)	70	0.770	0.801	278	77.92	76.00	76.95	94.64
		all	0.752	0.794	278	76.96	75.01	75.97	94.63
	Baseline (Pre-tag)	70	0.771	0.804	295	78.35	76.72	77.52	**95.68**
		all	0.752	0.796	295	77.31	75.64	76.47	95.68
	GoldPOS	70	0.775	0.808	309	78.83	77.18	77.99	96.60
Berkeley (Sep. 09)	(Petrov, 2009)	all	—	—	—	76.40	75.30	75.85	—
	Baseline	70	**0.809**	**0.839**	335	**82.32**	**81.63**	**81.97**	95.07
		all	0.796	0.834	336	81.43	80.73	81.08	95.02
	GoldPOS	70	0.831	0.859	496	84.37	84.21	84.29	99.87

Table 7: Test set results. Maamouri et al. (2009b) evaluated the Bikel parser using the same ATB split, but only reported dev set results with gold POS tags for sentences of length ≤ 40. The Bikel GoldPOS configuration only supplies the gold POS tags; it does not force the parser to use them. We are unaware of prior results for the Stanford parser.

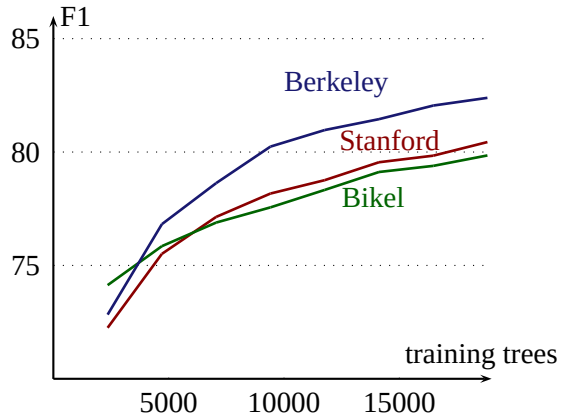


Figure 3: Dev set learning curves for sentence lengths ≤ 70. All three curves remain steep at the maximum training set size of 18818 trees.

The Leaf Ancestor metric measures the cost of transforming guess trees to the reference (Sampson and Babarczy, 2003). It was developed in response to the non-terminal/terminal bias of Evalb, but Clegg and Shepherd (2005) showed that it is also a valuable diagnostic tool for trees with complex deep structures such as those found in the ATB. For each terminal, the Leaf Ancestor metric extracts the shortest path to the root. It then computes a normalized Levenshtein edit distance between the extracted chain and the reference. The range of the score is between 0 and 1 (higher is better). We report micro-averaged (whole corpus) and macro-averaged (per sentence) scores along with the number of exactly matching guess trees.

add a constraint on the removal of punctuation, which has a single tag (PUNC) in the ATB. Tokens tagged as PUNC are not discarded unless they consist entirely of punctuation.

5.1 Parsing Models

The Stanford parser includes both the manually annotated grammar (§4) and an Arabic unknown word model with the following lexical features:

1. Presence of the determiner ال *Al*
2. Contains digits
3. Ends with the feminine affix ة *p*
4. Various verbal (e.g., وا, ت) and adjectival suffixes (e.g., ية)

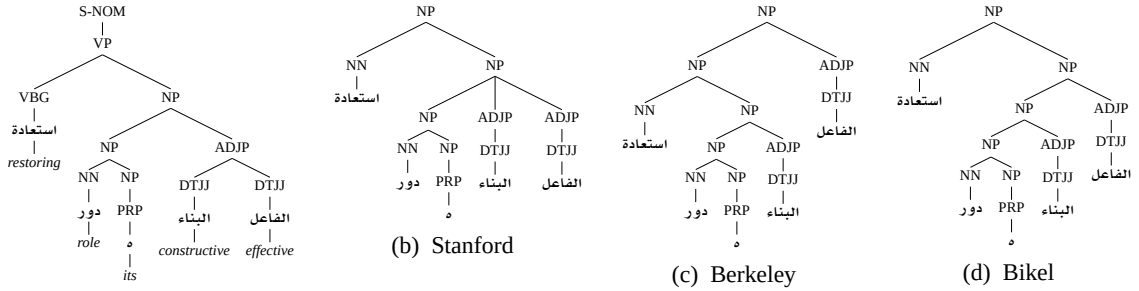
Other notable parameters are second order vertical Markovization and marking of unary rules.

Modifying the Berkeley parser for Arabic is straightforward. After adding a ROOT node to all trees, we train a grammar using six split-and-merge cycles and no Markovization. We use the default inference parameters.

Because the Bikel parser has been parameterized for Arabic by the LDC, we do not change the default model settings. However, when we pre-tag the input—as is recommended for English—we notice a 0.57% F1 improvement. We use the log-linear tagger of Toutanova et al. (2003), which gives 96.8% accuracy on the test set.

5.2 Discussion

The Berkeley parser gives state-of-the-art performance for all metrics. Our baseline for all sentence lengths is 5.23% F1 higher than the best previous result. The difference is due to more careful



(a) Reference

Figure 4: The constituent *Restoring of its constructive and effective role* parsed by the three different models (gold segmentation). The ATB annotation distinguishes between verbal and nominal readings of *maSdar* process nominals. Like verbs, *maSdar* takes arguments and assigns case to its objects, whereas it also demonstrates nominal characteristics by, e.g., taking determiners and heading *iDafa* (Fassi Fehri, 1993). In the ATB, استعادة *asta'adah* is tagged 48 times as a noun and 9 times as verbal noun. Consequently, all three parsers prefer the nominal reading. Table 8b shows that verbal nouns are the hardest pre-terminal categories to identify. None of the models attach the attributive adjectives correctly.

pre-processing. However, the learning curves in Figure 3 show that the Berkeley parser does not exceed our manual grammar by as wide a margin as has been shown for other languages (Petrov, 2009). Moreover, the Stanford parser achieves the most exact Leaf Ancestor matches and tagging accuracy that is only 0.1% below the Bikel model, which uses pre-tagged input.

In Figure 4 we show an example of variation between the parsing models. We include a list of per-category results for selected phrasal labels, POS tags, and dependencies in Table 8. The errors shown are from the Berkeley parser output, but they are representative of the other two parsing models.

6 Joint Segmentation and Parsing

Although the segmentation requirements for Arabic are not as extreme as those for Chinese, Arabic is written with certain cliticized prepositions, pronouns, and connectives connected to adjacent words. Since these are distinct syntactic units, they are typically segmented. The ATB segmentation scheme is one of many alternatives. Until now, all evaluations of Arabic parsing—including the experiments in the previous section—have assumed gold segmentation. But gold segmentation is not available in application settings, so a segmenter and parser are arranged in a pipeline. Segmentation errors cascade into the parsing phase, placing an artificial limit on parsing performance.

Lattice parsing (Chappelier et al., 1999) is an alternative to a pipeline that prevents cascading errors by placing all segmentation options into the parse chart. Recently, lattices have been used successfully in the parsing of Hebrew (Tsarfaty, 2006; Cohen and Smith, 2007), a Semitic language with similar properties to Arabic. We extend the Stanford parser to accept pre-generated lattices, where each word is represented as a finite state automaton. To combat the proliferation of parsing edges, we prune the lattices according to a hand-constructed lexicon of 31 clitics listed in the ATB annotation guidelines (Maamouri et al., 2009a). Formally, for a lexicon L and segments $I \in L$, $O \notin L$, each word automaton accepts the language $I^*(O+I)I^*$. Aside from adding a simple rule to correct *alif* deletion caused by the preposition ل, no other language-specific processing is performed.

Our evaluation includes both weighted and unweighted lattices. We weight edges using a unigram language model estimated with Good-Turing smoothing. Despite their simplicity, unigram weights have been shown as an effective feature in segmentation models (Dyer, 2009).[13] The joint parser/segmenter is compared to a pipeline that uses MADA (v3.0), a state-of-the-art Arabic segmenter, configured to replicate ATB segmentation (Habash and Rambow, 2005). MADA uses an ensemble of SVMs to first re-rank the output of a deterministic morphological analyzer. For each

[13] Of course, this weighting makes the PCFG an improper distribution. However, in practice, unknown word models also make the distribution improper.

Label	# gold	F1
ADJP	**1216**	**59.45**
SBAR	**2918**	**69.81**
FRAG	254	72.87
VP	5507	78.83
S	6579	78.91
PP	7516	80.93
NP	34025	84.95
ADVP	1093	90.64
WHNP	787	96.00

(a) Major phrasal categories

Tag	# gold	%	Tag	# gold	%
VBG	**182**	**48.84**	**JJR**	**134**	**92.83**
VN	**163**	**60.37**	DTNNS	1069	94.29
VBN	352	72.42	**DTJJ**	**3361**	**95.07**
DTNNP	932	83.48	NNP	4152	95.09
JJ	**1516**	**86.09**	NN	10336	95.23
ADJ_NUM	**277**	**88.93**	DTNN	6736	95.78
VBP	2139	89.94	NOUN_QUANT	352	98.16
RP	818	91.23	PRP	1366	98.24
NNS	907	91.75	CC	4076	98.92
DTJJR	**78**	**92.41**	IN	8676	99.07
VBD	2580	92.42	DT	525	99.81

(b) Major POS categories

Parent	Head	Modifer	Dir	# gold	F1
NP	NP	TAG	R	946	0.54
S	**S**	**S**	**R**	**708**	**0.57**
NP	**NP**	**ADJP**	**R**	**803**	**0.64**
NP	**NP**	**NP**	**R**	**2907**	**0.66**
NP	NP	SBAR	R	1035	0.67
NP	**NP**	**PP**	**R**	**2713**	**0.67**
VP	TAG	PP	R	3230	0.80
NP	NP	TAG	L	805	0.85
VP	TAG	SBAR	R	772	0.86
S	VP	NP	L	961	0.87

(c) Ten lowest scoring (Collins, 2003)-style dependencies occurring more than 700 times

Table 8: Per category performance of the Berkeley parser on sentence lengths ≤ 70 (dev set, gold segmentation). **(a)** Of the high frequency phrasal categories, ADJP and SBAR are the hardest to parse. We showed in §2 that lexical ambiguity explains the underperformance of these categories. **(b)** POS tagging accuracy is lowest for *maSdar* verbal nouns (VBG,VN) and adjectives (e.g., JJ). Richer tag sets have been suggested for modeling morphologically complex distinctions (Diab, 2007), but we find that linguistically rich tag sets do not help parsing. **(c)** Coordination ambiguity is shown in dependency scores by e.g., ⟨S S S R⟩ and ⟨NP NP NP R⟩. ⟨NP NP PP R⟩ and ⟨NP NP ADJP R⟩ are both *iDafa* attachment.

input token, the segmentation is then performed deterministically given the 1-best analysis.

Since guess and gold trees may now have different yields, the question of evaluation is complex. Cohen and Smith (2007) chose a metric like SParseval (Roark et al., 2006) that first aligns the trees and then penalizes segmentation errors with an edit-distance metric. But we follow the more direct adaptation of Evalb suggested by Tsarfaty (2006), who viewed exact segmentation as the ultimate goal. Therefore, we only score guess/gold pairs with identical *character* yields, a condition that allows us to measure parsing, tagging, and segmentation accuracy by ignoring whitespace.

Table 9 shows that MADA produces a high quality segmentation, and that the effect of cascading segmentation errors on parsing is only 1.92% F1. However, MADA is language-specific and relies on manually constructed dictionaries. Conversely, the lattice parser requires no linguistic resources and produces segmentations of comparable quality. Nonetheless, parse quality is much lower in the joint model because a lattice is effectively a long sentence. A cell in the bottom row of the parse chart is required for each potential whitespace boundary. As we have said, parse quality decreases with sentence length. Finally, we note that simple weighting gives nearly a 2% F1 improvement, whereas Goldberg and Tsarfaty (2008) found that unweighted lattices were more effective for Hebrew.

	LP	LR	F1	Seg F1	Tag F1	Coverage
STANFORD (Gold)	81.64	80.55	81.09	100.0	95.81	100.0%
MADA	—	—	—	97.67	—	96.42%
MADA+STANFORD	**79.44**	**78.90**	**79.17**	**97.67**	**94.27**	**96.42%**
STANFORDJOINT	76.13	72.61	74.33	94.12	90.13	94.73%
STANFORDJOINT+UNI	77.09	74.97	76.01	96.26	92.23	95.87%

Table 9: Dev set results for sentences of length ≤ 70. Coverage indicates the fraction of hypotheses in which the character yield exactly matched the reference. Each model was able to produce hypotheses for all input sentences. In these experiments, the input lacks segmentation markers, hence the slightly different dev set baseline than in Table 6.

7 Conclusion

By establishing significantly higher parsing baselines, we have shown that Arabic parsing performance is not as poor as previously thought, but remains much lower than English. We have described grammar state splits that significantly improve parsing performance, catalogued parsing errors, and quantified the effect of segmentation errors. With a human evaluation we also showed that ATB inter-annotator agreement remains low relative to the WSJ corpus. Our results suggest that current parsing models would benefit from better annotation consistency and enriched annotation in certain syntactic configurations.

Acknowledgments We thank Steven Bethard, Evan Rosen, and Karen Shiells for material contributions to this work. We are also grateful to Markus Dickinson, Ali Farghaly, Nizar Habash, Seth Kulick, David McClosky, Claude Reichard, Ryan Roth, and Reut Tsarfaty for constructive discussions. The first author is supported by a National Defense Science and Engineering Graduate (NDSEG) fellowship. This paper is based on work supported in part by DARPA through IBM. The content does not necessarily reflect the views of the U.S. Government, and no official endorsement should be inferred.

References

Al-Batal, M. 1990. Connectives as cohesive elements in a modern expository Arabic text. In Eid, Mushira and John McCarthy, editors, *Perspectives on Arabic Linguistics II*. John Benjamins.

Arun, A and F Keller. 2005. Lexicalization in crosslinguistic probabilistic parsing: The case of French. In *ACL*.

Bikel, D M. 2004. Intricacies of Collins' parsing model. *Computational Linguistics*, 30:479–511.

Chappelier, J-C, M Rajman, R Arages, and A Rozenknop. 1999. Lattice parsing for speech recognition. In *TALN*.

Chiang, D, M Diab, N Habash, O Rambow, and S Shareef. 2006. Parsing Arabic dialects. In *EACL*.

Clegg, A and A Shepherd. 2005. Evaluating and integrating treebank parsers on a biomedical corpus. In *ACL Workshop on Software*.

Cohen, S and N A Smith. 2007. Joint morphological and syntactic disambiguation. In *EMNLP*.

Collins, M, J Hajic, L Ramshaw, and C Tillmann. 1999. A statistical parser for Czech. In *ACL*.

Collins, M. 2003. Head-Driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Cowan, B and M Collins. 2005. Morphology and reranking for the statistical parsing of Spanish. In *NAACL*.

Diab, M. 2007. Towards an optimal POS tag set for Modern Standard Arabic processing. In *RANLP*.

Dickinson, M. 2005. *Error Detection and Correction in Annotated Corpora*. Ph.D. thesis, The Ohio State University.

Dubey, A and F Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *ACL*.

Dyer, C. 2009. Using a maximum entropy model to build segmentation lattices for MT. In *NAACL*.

Fassi Fehri, A. 1993. *Issues in the structure of Arabic clauses and words*. Kluwer Academic Publishers.

Gabbard, R and S Kulick. 2008. Construct state modification in the Arabic treebank. In *ACL*.

Goldberg, Y and R Tsarfaty. 2008. A single generative model for joint morphological segmentation and syntactic parsing. In *ACL*.

Green, S, C Sathi, and C D Manning. 2009. NP subject detection in verb-initial Arabic clauses. In *Proc. of the Third Workshop on Computational Approaches to Arabic Script-based Languages (CAASL3)*.

Habash, N and O Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *ACL*.

Habash, N and O Rambow. 2007. Arabic diacritization through full morphological tagging. In *NAACL*.

Habash, N and R Roth. 2009. CATiB: The Columbia Arabic Treebank. In *ACL, Short Papers*.

Habash, N and F Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *NAACL*.

Habash, N, B Dorr, and C Monz. 2006. Challenges in building an Arabic-English GHMT system with SMT components. In *EAMT*.

Hajič, J and P Zemánek. 2004. Prague Arabic dependency treebank: Development in data and tools. In *NEMLAR*.

Huang, Z and M Harper. 2009. Self-training PCFG grammars with latent annotations across languages. In *EMNLP*.

Klein, D and C D Manning. 2002. Fast exact inference with a factored model for natural language parsing. In *NIPS*.

Klein, D and C D Manning. 2003. Accurate unlexicalized parsing. In *ACL*.

Kübler, S. 2005. How do treebank annotation schemes influence parsing results? Or how not to compare apples and oranges. In *RANLP*.

Kulick, S, R Gabbard, and M Marcus. 2006. Parsing the Arabic Treebank: Analysis and improvements. In *TLT*.

Levy, R and C D Manning. 2003. Is it harder to parse Chinese, or the Chinese treebank? In *ACL*.

Maamouri, M and A Bies. 2004. Developing an Arabic Treebank: Methods, guidelines, procedures, and tools. In *Proc. of the Workshop on Computational Approaches to Arabic Script-based Languages (CAASL1)*.

Maamouri, M, A Bies, T Buckwalter, and W Mekki. 2004. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR*.

Maamouri, M, A Bies, and S Kulick. 2008. Enhancing the Arabic Treebank: A collaborative effort toward new annotation guidelines. In *LREC*.

Maamouri, M, A Bies, S Krouna, F Gaddeche, and B Bouziri. 2009a. Penn Arabic Treebank guidelines v4.92. Technical report, Linguistic Data Consortium, University of Pennsylvania, August 5.

Maamouri, M, A Bies, and S Kulick. 2009b. Creating a methodology for large-scale correction of treebank annotation: The case of the Arabic Treebank. In *MEDAR*.

Marcus, M, M A Marcinkiewicz, and B Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

Petrov, S, L Barrett, R Thibaux, and D Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *ACL*.

Petrov, S. 2009. *Coarse-to-Fine Natural Language Processing*. Ph.D. thesis, University of California-Berkeley.

Rehbein, I and J van Genabith. 2007. Treebank annotation schemes and parser evaluation for German. In *EMNLP-CoNLL*.

Roark, B, M Harper, E Charniak, B Dorr, M Johnson, J G Kahne, Y Liuf, Mari Ostendorf, J Hale, A Krasnyanskaya, M Lease, I Shafran, M Snover, R Stewart, and L Yung. 2006. SParseval: Evaluation metrics for parsing speech. In *LREC*.

Ryding, K. 2005. *A Reference Grammar of Modern Standard Arabic*. Cambridge University Press.

Sampson, G and A Babarczy. 2003. A test of the leaf-ancestor metric for parse accuracy. *Natural Language Engineering*, 9:365–380.

Toutanova, K, D Klein, C D Manning, and Y Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *NAACL*.

Tsarfaty, R and K Sima'an. 2008. Relational-realizational parsing. In *COLING*.

Tsarfaty, R. 2006. Integrated morphological and syntactic disambiguation for Modern Hebrew. In *ACL*.

Zitouni, I, J S Sorensen, and R Sarikaya. 2006. Maximum entropy based restoration of Arabic diacritics. In *ACL*.