Improving Arabic Dependency Parsing with Lexical and Inflectional Morphological Features

Yuval Marton, Nizar Habash and **Owen Rambow**
Center for Computational Learning Systems (CCLS)
Columbia University
{ymarton,habash,rambow}@ccls.columbia.edu

Abstract

We explore the contribution of different lexical and inflectional morphological features to dependency parsing of Arabic, a morphologically rich language. We experiment with all leading POS tagsets for Arabic, and introduce a few new sets. We show that training the parser using a simple regular expressive extension of an impoverished POS tagset with high prediction accuracy does better than using a highly informative POS tagset with only medium prediction accuracy, although the latter performs best on gold input. Using controlled experiments, we find that definiteness (or determiner presence), the so-called phi-features (person, number, gender), and undiacritzed lemma are most helpful for Arabic parsing on predicted input, while case and state are most helpful on gold.

1 Introduction

Parsers need to learn the syntax of the modeled language, in order to project structure on newly seen sentences. Parsing model design aims to come up with features that best help parsers to learn the syntax and choose among different parses. One aspect of syntax, which is often not explicitly modeled in parsing, involves morphological constraints on syntactic structure, such as agreement. In this paper, we explore the role of morphological features in parsing Modern Standard Arabic (MSA). For MSA, the space of possible morphological features is fairly large. We determine which morphological features help and why, and we determine the upper bound for their contribution to parsing quality.

We first present the corpus we use (§2), then relevant Arabic linguistic facts (§3); we survey related work (§4), describe our experiments (§5), and conclude with analysis of parsing error types (§6).

2 Corpus

We use the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009). Specifically, we use the portion converted from part 3 of the Penn Arabic Treebank (PATB) (Maamouri et al., 2004) to the CATiB format, which enriches the CATiB dependency trees with full PATB morphological information. CATiB's dependency representation is based on traditional Arabic grammar and emphasizes syntactic case relations. It has a reduced POS tagset (with six tags only), but a standard set of eight dependency relations: **SBJ** and **OBJ** for subject and (direct or indirect) object, respectively, (whether they appear pre- or post-verbally); **IDF** for the idafa (possessive) relation; **MOD** for most other modifications; and other less common relations that we will not discuss here. For more information, see (Habash and Roth, 2009). The CATiB treebank uses the word segmentation of the PATB.[1] It splits off several categories of orthographic clitics, but not the definite article ال *Al*. In all of the experiments reported in this paper, we use the gold segmentation. An example CATiB dependency tree is shown in Figure 1.[2]

3 Relevant Linguistic Concepts

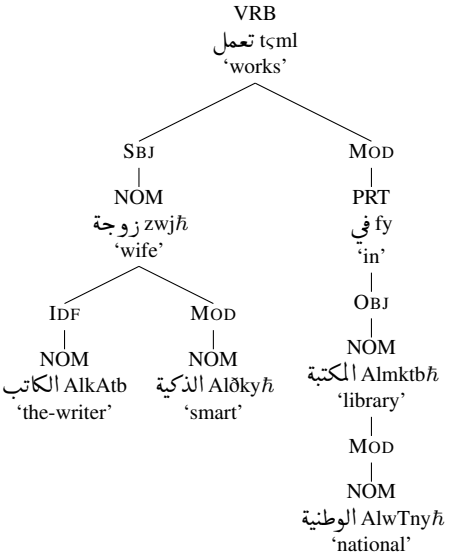
Morphemes: At a shallow level, Arabic words can be described in terms of their morphemes. In addition to concatenative prefixes and suffixes, Ara-

[1] Tokenization involves further decisions on the segmented token forms, such as spelling normalization.

[2] All Arabic transliterations are presented in the HSB transliteration scheme (Habash et al., 2007).

Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, pages 13–21,
Los Angeles, California, June 2010.

Figure 1: CATiB. تعمل زوجة الكاتب الذكية في المكتبة الوطنية *tʕml zwjħ AlkAtb Alðkyħ fy Almktbħ AlwTnyħ* 'The writer's smart wife works at the national library.' (Annotation example)



bic has templatic morphemes called *root* and *pattern*. For example, the word يكاتبون *yu+kAtib+uwn* 'they correspond' has one prefix and one suffix, in addition to a stem composed of the root ك ت ب *k-t-b* 'writing related' and the pattern *1A2i3*. [3]

Lexeme and Features: At a deeper level, Arabic words can be described in terms of sets of inflectional and lexical morphological features. We first discuss lexical features. The set of word forms that only vary inflectionally among each other is called the *lexeme*. A *lemma* is a particular word form used to represent, or cite, the lexeme word set. For example, verb lemmas are third person masculine singular perfective. We explore using both diacritized lemma, and undiacritized lemma (*lmm*). Just as the lemma abstracts over inflectional morphology, the root abstracts over both inflectional and derivational morphology and thus provides a deeper level of lexical abstraction than the lemma. The pattern feature is the pattern of the lemma of the lexeme, not of the word form.

The inflectional morphological features[4] define the dimensions of Arabic inflectional morphology, or the space of variations of a particular word. PATB-tokenized words vary along nine dimensions: GENDER and NUMBER (for nominals and verbs); PERSON, ASPECT, VOICE and MOOD (for verbs); and CASE, STATE, and the attached definite article proclitic DET (for nominals). The inflectional features abstract away from the specifics of morpheme forms, since they can affect more than one morpheme in Arabic. For example, changing the value of the aspect feature in the example above from imperfective to perfective yields the word form كاتبوا *kAtab+uwA* 'they corresponded', which differs in terms of prefix, suffix and pattern.

Inflectional features interact with syntax in two ways. First, there are agreement features: two words in a sentence which are in a specific syntactic configuration have the same value for a specific set of features. In MSA, we have subject-verb agreement on PERSON, GENDER, and NUMBER (but NUMBER only if the subject precedes the verb), and we have noun-adjective agreement in PERSON, NUMBER, GENDER, and DET.[5] Second, morphology can show a specific syntactic configuration on a single word. In MSA, we have CASE and STATE marking. Different types of dependents have different CASE; for example, verbal subjects are always marked NOMINATIVE. CASE and STATE are rarely explicitly manifested in undiacritized MSA.

Lexical features do not participate in syntactic constraints on structure as inflectional features do. Instead, bilexical dependencies are used in parsing to model semantic relations which often are the only way to disambiguate among different possible syntactic structures; lexical features provide a way of reducing data sparseness through lexical abstraction. We compare the effect on parsing of different subsets of lexical and inflectional features. Our hypothesis is that the inflectional features involved in agreement and the lexical features help parsing.

The core POS tagsets: Words also have associated part-of-speech (POS) tags, e.g., "verb", which further abstract over morphologically and syntactically similar lexemes. Traditional Arabic grammars often describe a very general three-way distinction into verbs, nominals and particles. In comparison, the tagset of the Buckwalter Morphological Analyzer (Buckwalter, 2004) used in the PATB has a core POS set of 44 tags (before morphologi-

[3]The digits in the pattern correspond to the positions root radicals are inserted.

[4]The inflectional features we use in this paper are form-based (illusory) as opposed to functional features (Smrž, 2007). We plan to work with functional features in the future.

[5]We do not explicitly address here agreement phenomena that require more complex morpho-syntactic modeling. These include adjectival modifiers of irrational (non-human) plural nominals, and pre-nominal number modifiers.

cal extension). Henceforth, we refer to this tagset as CORE44. Cross-linguistically, a core set containing around 12 tags is often assumed, including: noun, proper noun, verb, adjective, adverb, preposition, particles, connectives, and punctuation. Henceforth, we reduce CORE44 to such a tagset, and dub it CORE12. The CATIB6 tagset can be viewed as a further reduction, with the exception that CATIB6 contains a passive voice tag; however, it constitutes only 0.5% of the tags in the training.

Extended POS tagsets: The notion of "POS tagset" in natural language processing usually does *not* refer to a core set. Instead, the Penn English Treebank (PTB) uses a set of 46 tags, including not only the core POS, but also the complete set of morphological features (this tagset is still fairly small since English is morphologically impoverished). In modern standard Arabic (MSA), the corresponding type of tagset (core POS extended with a complete description of morphology) would contain upwards of 2,000 tags, many of which are extremely rare (in our training corpus of about 300,000 words, we encounter only 430 of such POS tags with complete morphology). Therefore, researchers have proposed tagsets for MSA whose size is similar to that of the English PTB tagset, as this has proven to be a useful size computationally. These tagsets are hybrids in the sense that they are neither simply the core POS, nor the complete morphological tagset, but instead they choose certain morphological features to include along with the core POS tag.

The following are the various tagsets we compare in this paper: **(a)** the core POS tagsets CORE44 and the newly introduced CORE12; **(b)** CATiB treebank tagset (CATIB6) (Habash and Roth, 2009); and its newly introduced extension, CATIBEX, by greedy regular expressions indicating particular morphemes such as the prefix ال *Al+* or the suffix ون *+wn*.[6] **(c)** the PATB full tagset (BW), size ≈2000+ (Buckwalter, 2004); and two extensions of the PATB reduced tagset (PENN POS, a.k.a. RTS, size 24), both outperforming it: **(d)** Kulick et al. (2006)'s tagset (KULICK), size ≈43, one of whose most important extensions is the marking of the definite article clitic, and **(e)** Diab and BenAjiba (2010)'s EXTENDED RTS tagset (ERTS), which marks gender, number and definiteness, size ≈134; Besides using morphological information to extend POS tagsets, we explore using it in separate features in parsing models. Following this exploration, we also extend CORE12, producing **(f)** CORE12EX (see Section 5 for details).

[6] Inspired by a similar extension in Habash and Roth (2009).

4 Related Work

Much work has been done on the use of morphological features for parsing of morphologically rich languages. Collins et al. (1999) report that an optimal tagset for parsing Czech consists of a basic POS tag plus a CASE feature (when applicable). This tagset (size 58) outperforms the basic Czech POS tagset (size 13) and the complete tagset (size ≈3000+). They also report that the use of gender, number and person features did not yield any improvements. We get similar results for CASE in the gold experimental setting but not when using predicted POS tags (POS tagger output). This may be a result of CASE tagging having a lower error rate in Czech (5.0%) (Hajič and Vidová-Hladká, 1998) compared to Arabic (≈14.0%, see Table 3). Similarly, Cowan and Collins (2005) report that the use of a subset of Spanish morphological features (number for adjectives, determiners, nouns, pronouns, and verbs; and mode for verbs) outperforms other combinations. Our approach is comparable to their work in terms of its systematic exploration of the space of morphological features. We also find that the number feature helps for Arabic. Looking at Hebrew, a Semitic language related to Arabic, Tsarfaty and Sima'an (2007) report that extending POS and phrase structure tags with definiteness information helps unlexicalized PCFG parsing.

As for work on Arabic, results have been reported on PATB (Kulick et al., 2006; Diab, 2007), the Prague Dependency Treebank (PADT) (Buchholz and Marsi, 2006; Nivre, 2008) and the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009).

Besides the work we describe in §3, Nivre (2008) reports experiments on Arabic parsing using his MaltParser (Nivre et al., 2007), trained on the PADT. His results are not directly comparable to ours because of the different treebanks representations and tokenization used, even though all our experiments reported here were performed using the MaltParser. Our results agree with previous published work on Arabic and Hebrew in that marking the definite article is helpful for parsing. However, we go beyond previous work in that we also extend this morphologically enhanced feature set to include additional

lexical and inflectional morphological features. Previous work with MaltParser in Russian, Turkish and Hindi showed gains with case but not with agreement features (Nivre et al., 2008; Eryigit et al., 2008; Nivre, 2009). Our work is the first to show gains using agreement in MaltParser and in Arabic dependency parsing.

5 Experiments

5.1 Experimental Space

We examined a large space of settings including the following: **(a)** the contribution of POS tagsets to the parsing quality, as a function of the amount of information encoded in the tagset; **(b)** parsing performance on gold vs. predicted POS and morphological feature values for all models; **(c)** prediction accuracy of each POS tagset and morphological feature; **(d)** the contribution of numerous morphological features in a controlled fashion; and **(e)** the contribution of certain feature and POS tagset combinations. All results are reported mainly in terms of labeled attachment accuracy (parent word and the dependency relation to it). Unlabeled attachment accuracy and label accuracy are also given, space permitting.

5.2 Parser

For all experiments reported here we used the syntactic dependency parser MaltParser v1.3 (Nivre, 2003; Nivre, 2008; Kübler et al., 2009) – a transition-based parser with an input buffer and a stack, using SVM classifiers to predict the next state in the parse derivation. All experiments were done using the Nivre "eager" algorithm.[7] We trained the parser on the training portion of PATB part 3 (Maamouri et al., 2004). We used the same split as in Zitouni et al. (2006) for dev/test, and kept the test unseen during training.

There are five default *attributes*, in the MaltParser terminology, for each token in the text: word ID (ordinal position in the sentence), word form, POS tag, head (parent word ID), and deprel (the dependency relation between the current word and its parent). There are default *MaltParser features* (in the machine learning sense),[8] which are the values of functions over these attributes, serving as input to the MaltParser internal classifiers. The most commonly used feature functions are the top of the input buffer (next word to process, denoted buf[0]), or top of the stack (denoted stk[0]); following items on buffer or stack are also accessible (buf[1], buf[2], stk[1], etc.). Hence MaltParser features are defined as POS tag at top of the stack, word form at top of the buffer, etc. Kübler et al. (2009) describe a "typical" MaltParser model configuration of attributes and features.[9] Starting with it, in a series of initial controlled experiments, we settled on using buf[0], buf[1], stk[0], stk[1] for the wordform, and buf[0], buf[1], buf[2], buf[3], stk[0], stk[1], stk[2] for the POS tag. For features of all new MaltParser-attributes (discussed later), we used buf[0] and stk[0]. We did not change the features for the deprel. This new MaltParser configuration resulted in gains of 0.3-1.1% in labeled attachment accuracy (depending on the POS tagset) over the default MaltParser configuration. We also experimented with using normalized word forms (*Alif Maqsura* conversion to *Ya*, and hamza removal from each *Alif*) as is common in parsing and statistical machine translation literature. This resulted in a small decrease in performance (0.1-0.2% in labeled attachment accuracy). We settled on using the non-normalized word form. All experiments reported below were conducted using this new configuration.

5.3 Parsing quality as a function of POS tag richness

We turn first to the contribution of POS information to parsing quality, as a function of the amount of information encoded in the POS tagset. A first rough estimation for the amount of information is the actual tagset size, as it appears in the training data. For this purpose we compared POS tagsets based on, or closely inspired by, previously published work. These sets are typically morphologically-enriched (marking the existence of a determiner in the word, person, gender, number, etc.). The num-

[7] Nivre (2008) reports that non-projective and pseudo-projective algorithms outperform the "eager" projective algorithm in MaltParser; however, our training data did not contain any non-projective dependencies, so there was no point in using these algorithms. The Nivre "standard" algorithm is also reported to do better on Arabic, but in a preliminary experimentation, it did slightly worse than the "eager" one. This could be due to high percentage of right branching (left headed structures) in our Arabic training set, an observation already noted in Nivre (2008).

[8] The terms "feature" and "attribute" are over loaded in the literature. We use them in the linguistic sense, unless specifically noted otherwise, e.g., "MaltParser feature(s)".

[9] It is slightly different from the default configuration.

ber of tag types occurring in the training data follow each tagset in parentheses: BW (430 tags), ERTS (134 tags), KULICK (32 tags), and the smallest POS tagset published: CATIB6 (6 tags). In optimal conditions (using gold POS tags), the richest tagset (BW) is indeed the best performer (84.02%), and the poorest (CATIB6) is the worst (81.04%). Mid-size tagsets are in the high 82%, with the notable exception of KULICK, which does better than ERTS, in spite of having 1/4 the tagset size; moreover, it is the best performer in unlabeled attachment accuracy (85.98%), in spite of being less than tenth the size of BW. Our extended mid-size tagset, CATIBEX, was a mid-level performer as expected.

In order to control the level of morphological and lexical information in the POS tagset, we used the above-mentioned additional tagsets: CORE44 (40 tags), and CORE12 (12 tags). Both were also mid-size mid-level performers (in spite of containing no morphological extension), with CORE12 doing slightly better. See Table 1 columns 2-4.

5.4 Predicted POS tags

So far we discussed optimal (gold) conditions. But in practice, POS tags are annotated by automatic taggers, so parsers get *predicted* POS tags as input, as opposed to gold (human-annotated) tags. The more informative the tagset, the less accurate the tag prediction might be, so the effect on overall parsing quality is unclear. Therefore, we repeated the experiments above with POS tags predicted by the Morphological Analysis and Disambiguation for Arabic (MADA) toolkit (Habash and Rambow, 2005). See Table 1, columns 5-7. It turned out that BW, the best gold performer, with lowest POS prediction accuracy (81.8%), suffered the biggest drop (11.38%) and was the worst performer with predicted tags. The simplest tagset, CATIB6, and its extension, CATIBEX, benefited from the highest POS prediction accuracy (97.7%), and their performance suffered the least. CATIBEX was the best performer with predicted POS tags. Performance drop and POS prediction accuracy are given in columns 8 and 9, respectively. Next, we augmented the parsing models with inflectional and lexical morphological features.

5.5 Inflectional features

Experimenting with inflectional morphological features is especially important in Arabic parsing, since Arabic is morphologically rich. In order to further explore the contribution of inflectional and lexical morphological information in a controlled manner, we focused on the best performing core POS tagset, CORE12 as baseline; using three different setups, we added nine morphological features, extracted from MADA: DET, PERSON, ASPECT, VOICE, MOOD, GENDER, NUMBER, STATE, and CASE. In setup ***All***, we augmented the baseline model with all nine MADA features (as nine additional MaltParser attributes); in setup ***Sep***, we augmented the baseline model with each of the MADA features, one at a time, separately; and in setup ***Greedy***, we combined them in a greedy heuristic (since the entire feature space is too vast to exhaust): starting with the most gainful feature from *Sep*, adding the next most gainful feature, keeping it as additional MaltParser attribute if it helped, or discarding it otherwise, and repeating this heuristics through the least gainful feature. We also augmented the same baseline CORE12 model with a manually constructed list of surface affixes (e.g., *Al+, +wn, ħ*) as additional MaltParser attributes (LINGNGRAMS). This list was also in the base of the CATIBEX extension; it is linguistically informed, yet represents a simple (albeit shallow) alternative to morphological analysis. Results are given in Table 2.

Somewhat surprisingly, setup *All* hurts performance on the predicted input. This can be explained if one examines the prediction accuracy of each feature (Table 3). Features which are not predicted with very high accuracy, such as CASE (86.3%), can dominate the negative contribution, even though they are principle top contributors in optimal (gold) conditions (see discussion below). The determiner feature (DET), followed by the STATE (construct state, *idafa*) feature, were top individual contributors in setup *Sep*. Adding DET and all the so-called phi-features (PERSON, NUMBER, GENDER) in the *Greedy* setup, yielded 1.43% gain over the CORE12 baseline. Adding LINGNGRAMS yielded a 1.19% gain over the CORE12 baseline.

We repeated the same setups (*All, Sep*, and *Greedy*) with gold POS tags, to examine the contribution of the morphological features in optimal conditions. Here CASE, followed by STATE and DET, were the top contributors. Performance of CASE is the notable difference from the predicted conditions above. Surprisingly, only CASE and STATE helped in the *Greedy* setup, although one might expect that the phi features help too. (See lower half of Table 2).

Table 1: Parsing performance with each POS tagset, on gold and predicted input. labeled = labeled attachment accuracy (dependency + relation). unlabeled = unlabeled attachment accuracy (dependency only). label acc = relation label prediction accuracy. labeled diff = difference between labeled attachment accuracy on gold and predicted input. POS acc = POS tag prediction accuracy.

tagset	gold			predicted			gold-pred.	POS	tagset
	labeled	unlabled	label acc.	labeled	unlabled	label acc.	labeled diff.	acc.	size
CATIB6	81.04	83.66	92.59	78.31	82.03	90.55	**-2.73**	**97.7**	6
CATIBEX	82.52	84.97	93.40	**79.74**	**83.30**	**91.44**	-2.78	**97.7**	44
CORE12	82.92	85.40	93.52	78.68	82.48	90.63	-4.24	96.3	12
CORE44	82.71	85.17	93.28	78.39	82.16	90.36	-4.32	96.1	40
ERTS	82.97	85.23	93.76	78.93	82.56	90.96	-4.04	95.5	134
KULICK	83.60	**85.98**	94.01	79.39	83.15	91.14	-4.21	95.7	32
BW	**84.02**	85.77	**94.83**	72.64	77.91	86.46	-11.38	81.8	430

Table 2: CORE12 POS tagset with morphological features. Left half: Using predicted POS tags. In it: Top part: Adding all nine features to CORE12. Second part: Adding each feature separately, comparing difference from CORE12+madafeats, predicted (second part). Third part: Greedily adding best features from third part, predicted; difference from previous successful greedy step. Bottom part: Surface affixes (leading and trailing character n-grams). Right half: Left half repeated with gold tags.

set -up	predicted POS and features: CORE12+...	labeled	diff.	unlabeled	gold POS and features: CORE12+...	labeled	diff.	unlabeled
All	(baseline repeated)	78.68	–	82.48	(baseline repeated)	82.92	–	85.40
	+madafeats	77.91	-0.77	82.14	+madafeats	85.15	2.23	86.61
Sep	+DET	**79.82**	**1.14**	83.18	+CASE	**84.61**	**1.69**	86.30
	+STATE	79.34	0.66	82.85	+STATE	84.15	1.23	86.38
	+GENDER	78.75	0.07	82.35	+DET	83.96	1.04	86.21
	+PERSON	78.74	0.06	82.45	+NUMBER	83.08	0.16	85.50
	+NUMBER	78.66	-0.02	82.39	+PERSON	83.07	0.15	85.41
	+VOICE	78.64	-0.04	82.41	+VOICE	83.05	0.13	85.42
	+ASPECT	78.60	-0.08	82.39	+MOOD	83.05	0.13	85.47
	+MOOD	78.54	-0.14	82.35	+ASPECT	83.01	0.09	85.43
	+CASE	75.81	-2.87	80.24	+GENDER	82.96	0.04	85.24
Greedy	+DET+STATE	79.42	-0.40	82.84	+CASE+STATE	**85.37**	**0.76**	86.88
	+DET+GENDER	79.90	0.08	83.20	+CASE+STATE+DET	85.18	-0.19	86.66
	+DET+GENDER+PERSON	79.94	0.04	83.21	+CASE+STATE+NUMBER	85.36	-0.01	86.87
	+DET+PHI	**80.11**	**0.17**	83.29	+CASE+STATE+PERSON	85.27	-0.10	86.76
	+DET+PHI+VOICE	79.96	-0.15	83.18	+CASE+STATE+VOICE	85.25	-0.12	86.76
	+DET+PHI+ASPECT	80.01	-0.10	83.20	+CASE+STATE+MOOD	85.23	-0.14	86.72
	+DET+PHI+MOOD	80.03	-0.08	83.21	+CASE+STATE+ASPECT	85.23	-0.14	86.78
	—				+CASE+STATE+GENDER	85.26	-0.11	86.75
	+NGRAMSLING	**79.87**	**1.19**	83.21	+NGRAMSLING	**84.02**	**1.10**	86.16

5.6 Lexical features

Next, we experimented with adding morphological features involving semantic abstraction to some degree: the diacritized LEMMA (abstracting away from inflectional information, and indicating active/passive voice due to diacritization information), the undiacritized lemma (LMM), the ROOT (further abstraction indicating "core" predicate or action), and the PATTERN (a generally complementary abstraction, often indicating causation and reflexiveness). We experimented with the same setups as above: *All*, *Sep*, and *Greedy*. Adding all four features yielded a minor gain in setup *All*. LMM was the best single contributor (1.05%), closely followed by ROOT (1.03%) in *Sep*. CORE12+LMM+ROOT+LEMMA was the best greedy combination (79.05%) in setup *Greedy*. See Table 4.

5.7 Putting it all together

We further explored whether morphological data should be added to an Arabic parsing model as stand-alone machine learning features, or should they be used to enhance and extend a POS tagset. We created a new POS tagset, CORE12EX, size 81(see bottom of Table 3), by extending the CORE12 tagset with the features that most improved the

CORE12 baseline: DET and the phi features. But CORE12EX did worse than its non-extended (but feature-enhanced) counterpart, CORE12+DET+PHI. Another variant, CORE12EX+DET+PHI, which used both the extended tagset and the additional DET and phi features, did not improve over CORE12+DET+PHI either.

Following the results in Table 2, we added the affix features NGRAMSLING (which proved to help the CORE12 baseline) to the best augmented CORE12+DET+PHI model, dubbing the new model CORE12+DET+PHI+NGRAMSLING, but performance dropped here too. We greedily augmented CORE12+DET+PHI with lexical features, and found that the undiacritzed lemma (LMM) improved performance on predicted input (80.23%). In order to test whether these findings hold with other tagsets, we added the winning features (DET+PHI, with and without LMM) to the best POS tagset in predicted conditions, CATIBEX. Both variants yielded gains, with CATIBEX+DET+PHI+LMM achieving 80.45% accuracy, the best result on predicted input.

5.8 Validating Results on Unseen Test Set

Once experiments on the development set (PATB3-DEV) were done, we ran the best performing models on a previously unseen test set – the test split of part 3 of the PATB (PATB3-TEST). Table 6 shows that the same trends held on this set too, with even greater relative gains, up to 1.77% absolute gains.

Table 3: Feature prediction accuracy and set sizes. * = The set includes a "N/A" value.

feature	acc	set size
normalized word form (A,Y)	99.3	29737
non-normalized word form	98.9	29980
NGRAMSLING preffix	100.0	8
NGRAMSLING suffix	100.0	20
DET	99.6	3*
PERSON	99.1	4*
ASPECT	99.1	5*
VOICE	98.9	4*
MOOD	98.6	5*
GENDER	99.3	3*
NUMBER	99.5	4*
STATE	95.6	4*
CASE	86.3	5*
ROOT	98.4	9646
PATTERN	97.0	338
LEMMA (diacritized)	96.7	16837
LMM (undiacritized lemma)	98.3	15305
CORE12EX	96.0	81

Table 4: Lexical morpho-semantic features. Top part: Adding each feature separately; difference from CORE12, predicted. Bottom part: Greedily adding best features from previous part, predicted; difference from previous successful greedy step.

	POS tagset	labeled	diff.	unlab.	label
All	CORE12 (repeated)	78.68	–	82.48	90.63
	CORE12+LMM+ROOT +LEMMA+PATTERN	78.85	0.17	82.46	90.82
Sep	CORE12+lmm	78.96	1.05	82.54	90.80
	CORE12+ROOT	78.94	1.03	82.64	90.72
	CORE12+LEMMA	78.80	0.89	82.42	90.71
	CORE12+PATTERN	78.59	0.68	82.39	90.60
Greedy	CORE12+LMM+ROOT	79.04	0.08	82.63	90.86
	CORE12+LMM+ROOT +LEMMA	**79.05**	0.01	82.63	90.87
	CORE12+LMM+ROOT +PATTERN	78.93	-0.11	82.58	90.82

Table 6: Results on PATB3-TEST for models which performed best on PATB3-DEV – predicted input.

POS tagset	labeled	diff.	unlab.	label
CORE12	77.29	–	81.04	90.05
CORE12+DET+PHI	78.57	1.28	81.66	91.09
CORE12+DET+PHI+LMM	**79.06**	**1.77**	82.07	91.37

6 Error Analysis

For selected feature sets, we look at the overall error reduction with respect to the CORE12 baseline, and see what dependency relations particularly profit from that feature combination: What dependencies achieve error reductions greater than the average error reduction for that feature set over the whole corpus. We investigate dependencies by labels, and for **MOD** we also investigate by the POS label of the dependent node (so **MOD-P** means a preposition node attached to a governing node using a **MOD** arc).

DET: As expected, it particularly helps **IDF** and **MOD-N**. The error reduction for **IDF** is 19.3%!

STATE: Contrary to naïve expectations, STATE does not help **IDF**, but instead *increases* error by 9.4%. This is presumably because the feature does not actually predict construct state except when construct state is marked explicitly, but this is rare.

DET+PHI: The phi features are the only subject-verb agreement features, and they are additional agreement features (in addition to definiteness) for noun-noun modification. Indeed, relative to just adding DET, we see the strongest increases in these two dependencies, with an additional average in-

Table 5: Putting it all together

POS tagset	inp.qual.	labeled	diff.	unlabeled	label Acc.
CORE12+DET+PHI (repeated)	predicted	**80.11**	0.17	83.29	91.82
CORE12+DET+PHI	gold	84.20	-0.95	86.23	94.49
CORE12EX	predicted	78.89	-1.22	82.38	91.17
CORE12EX	gold	83.06	0.14	85.26	93.80
CORE12EX+DET+PHI	predicted	79.19	-0.92	82.52	91.39
CORE12+DET+PHI+NGRAMSLING	predicted	79.77	-0.34	83.03	91.66
CORE12+DET+PHI+LMM	predicted	**80.23**	0.12	83.34	91.94
CORE12+DET+PHI+LMM+ROOT	predicted	80.10	-0.13	83.25	91.84
CORE12+DET+PHI+LMM+PATTERN	predicted	80.03	-0.20	83.15	91.77
CATIBEX+DET+PHI	predicted	80.00	0.26	83.29	91.81
CATIBEX+DET+PHI+LMM	predicted	**80.45**	0.71	83.65	92.03

crease for **IDF** (presumably because certain N-N modifications are rejected in favor of **IDF**s). All other dependencies remain at the same level as with only DET.

LMM, ROOT, LEMMA: These features abstract over the word form and thus allow generalizations in bilexical dependecies, which in parsing stand in for semantic modeling. The strongest boost from these features comes from **MOD-N** and **MOD-P**, which is as expected since these dependencies are highly ambiguous, and **MOD-P** is never helped by the morphological features.

DET+PHI+LMM: This feature combination yields gains on all main dependency types (**SBJ, OBJ, IDF, MOD-N, MOD-P, MOD-V**). But the contribution from the inflectional and lexical features are unfortunately not additive. We also compare the improvement contributed just by LMM as compared to DET and PHI. This improvement is quite small, but we see that **MOD-N** does not improve (in fact, it gets worse – presumably because there are too many features), while **MOD-P** (which is not helped by the morphological features) does improve. Oddly, **OBJ** also improves, for which we have no explanation.

When we turn to our best-performing configuration, CATIBEX with the added DET, phi features (PERSON, NUMBER, GENDER), and LMM, we see that this configuration improves over CORE12 with the same features for two dependency types only: **SBJ** and **MOD-N** These are exactly the two types for which agreement features are useful, and both the features DET+PHI and the CATIBEX POS tagset represent information for agreement. The question arises why this information is not redundant. We speculate that the fact that we are learning different classifiers for different POS tags helps MaltParser learn attachment decisions which are specific to types of dependent node morphology.

In summary, our best performing configuration yields an error reduction of 8.3% over the core POS tag (CORE12). **SBJ** errors are reduced by 13.3%, **IDF** errors by 17.7%, and **MOD-N** errors by 14.9%. Error reduction for **OBJ**, **MOD-P**, and **MOD-V** are all less than 4%. We note that the remaining **MOD-P** errors make up 6.2% of all dependency relations, roughly one third of remaining errors.

7 Conclusions and Future Work

We explored the contribution of different inflectional and lexical features to dependency parsing of Arabic, under gold and predicted POS conditions. While more informative features (e.g., richer POS tags) yield better parsing quality in gold conditions, they are hard to predict, and as such they might not contribute to – and even hurt – the parsing quality under predicted conditions. We find that definiteness (DET), phi-features (PERSON, NUMBER, GENDER), and undiacritzed lemma (LMM) are most helpful for Arabic parsing on predicted input, while CASE and STATE are most helpful on gold.

In the future we plan to improve CASE prediction accuracy; produce high accuracy supertag features, modeling active and passive valency; and use other parsers (e.g., McDonald and Pereira, 2006).

Acknowledgments

This work was supported by the DARPA GALE program, contract HR0011-08-C-0110. We thank Joakim Nivre for his useful remarks, and Ryan Roth for his help with CATiB conversion and MADA.

References

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of Computational Natural Language Learning (CoNLL)*, pages 149–164.

Timothy A. Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. Linguistic Data Consortium, University of Pennsylvania, 2002. LDC Cat alog No.: LDC2004L02, ISBN 1-58563-324-0.

Michael Collins, Jan Hajic, Lance Ramshaw, and Christoph Tillmann. 1999. A statistical parser for czech. In *Proceedings of the 37th Annual Meeting of the the Association for Computational Linguistics (ACL)*, College Park, Maryland, USA, June.

Brooke Cowan and Michael Collins. 2005. Morphology and reranking for the statistical parsing of spanish. In *Proceedings of Human Language Technology (HLT) and the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 795–802.

Mona Diab and Yassine BenAjiba. 2010. From raw text to base phrase chunks: The new generation of AMIRA Tools for the processing of Modern Standard Arabic. In *(to appear)*. Spring LNCS, Special Jubilee edition.

Mona Diab. 2007. Towards an optimal pos tag set for modern standard arabic processing. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria.

Gülsen Eryigit, Joakim Nivre, and Kemal Oflazer. 2008. Dependency parsing of turkish. *Computational Linguistics*, 34(3):357–389.

Nizar Habash and Owen Rambow. 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proceedings of the 43rd Annual Meeting of the the Association for Computational Linguistics (ACL)*, Ann Arbor, Michigan, June.

Nizar Habash and Ryan Roth. 2009. Catib: The columbia arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec, Singapore, August.

Nizar Habash, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

Jan Hajič and Barbora Vidová-Hladká. 1998. Tagging Inflective Languages: Prediction of Morphological Categories for a Rich, Structured Tagset. In *Proceedings of the International Conference on Computational Linguistics (COLING)- the Association for Computational Linguistics (ACL)*, pages 483–490.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool Publishers.

Seth Kulick, Ryan Gabbard, and Mitch Marcus. 2006. Parsing the Arabic Treebank: Analysis and improvements. In *Proceedings of the Treebanks and Linguistic Theories Conference*, pages 31–42, Prague, Czech Republic.

Mohamed Maamouri, Ann Bies, Timothy A. Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *Proceedings of the NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the the European Chapter of the Association for Computational Linguistics (EACL)*.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gulsen Eryigit, Sandra Kubler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Joakim Nivre, Igor M. Boguslavsky, and Leonid K. Iomdin. 2008. Parsing the SynTagRus Treebank of Russian. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)*, pages 641–648.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Conference on Parsing Technologies (IWPT)*, pages 149–160, Nancy, France.

Joakim Nivre. 2008. Algorithms for Deterministic Incremental Dependency Parsing. *Computational Linguistics*, 34(4).

Joakim Nivre. 2009. Parsing Indian languages with MaltParser. In *Proceedings of the ICON09 NLP Tools Contest: Indian Language Dependency Parsing*, pages 12–18.

Otakar Smrž. 2007. *Functional Arabic Morphology. Formal System and Implementation*. Ph.D. thesis, Charles University, Prague.

Reut Tsarfaty and Khalil Sima'an. 2007. Three-dimensional parametrization for parsing morphologically rich languages. In *Proceedings of the 10th International Conference on Parsing Technologies (IWPT)*, pages 156–167, Morristown, NJ, USA.

Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya. 2006. Maximum Entropy Based Restoration of Arabic Diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING) and the 44th Annual Meeting of the the Association for Computational Linguistics (ACL)*, pages 577–584, Sydney, Australia.