Improving Arabic Dependency Parsing with Form-based and Functional Morphological Features

Yuval Marton
T.J. Watson Research Center
IBM
yymarton@us.ibm.com

Nizar Habash and **Owen Rambow**
Center for Computational Learning Systems
Columbia University
{habash,rambow}@ccls.columbia.edu

Abstract

We explore the contribution of morphological features – both lexical and inflectional – to dependency parsing of Arabic, a morphologically rich language. Using controlled experiments, we find that definiteness, person, number, gender, and the undiacritzed lemma are most helpful for parsing on automatically tagged input. We further contrast the contribution of form-based and functional features, and show that functional gender and number (e.g., "broken plurals") and the related rationality feature improve over form-based features. It is the first time functional morphological features are used for Arabic NLP.

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, which often plays an important role in morphologically rich languages. 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. We also explore going beyond the easily detectable, regular form-based ("surface") features, by representing *functional* values for some morphological features. We expect that representing lexical abstractions and inflectional features participating in agreement relations would help parsing quality, but other inflectional features would not help. We further expect functional features to be superior to surface-only features.

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

2 Corpus

We use the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009). Specifically, we use the portion converted automatically 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 – henceforth CATIB6), 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 et al. (2009). The CATiB treebank uses the word segmentation of the PATB: 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

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1586–1596,
Portland, Oregon, June 19-24, 2011.

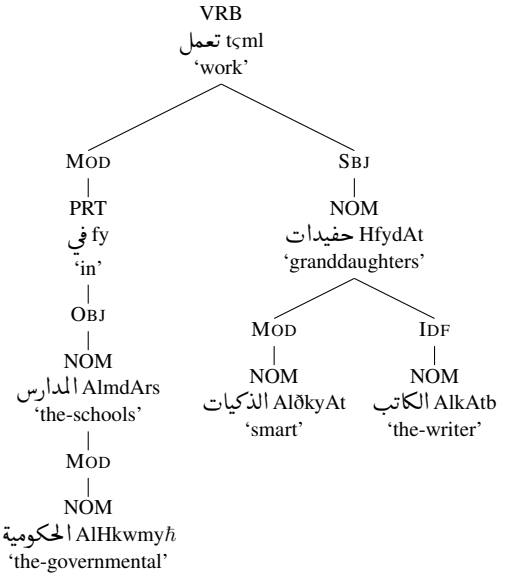

Figure 1: CATiB Annotation example (tree display from right to left). تعمل حفيدات الكاتب الذكيات في المدارس الحكومية *tçml HfydAt AlkAtb AlðkyAt fy AlmdArs AlHkwmyħ* 'The writer's smart granddaughters work for public schools.'

segmentation. An example CATiB dependency tree is shown in Figure 1.

3 Relevant Linguistic Concepts

In this section, we present the linguistic concepts relevant to our discussion of Arabic parsing.

Orthography The Arabic script uses optional diacritics to represent short vowels, consonantal doubling and the indefininteness morpheme (*nunation*). For example, the word كَتَبَ *kataba* 'he wrote' is often written as كتب *ktb*, which can be ambiguous with other words such as كُتُبٌ *kutubũ* 'books'. In news text, only around 1.6% of all words have any diacritic (Habash, 2010). As expected, the lack of diacritics contributes heavily to Arabic's morphological ambiguity. In this work, we only use undiacritized text; however, some of our parsing features which are derived through morphological disambiguation include diacritics (specifically, *lemmas*, see below).

Morphemes Words can be described in terms of their morphemes; in Arabic, in addition to concatenative prefixes and suffixes, there are 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 كتب *k-t-b* 'writing related' and the pattern *1A2i3*.[1]

Lexeme and features Alternatively, Arabic words can be described in terms of lexemes and inflectional features. The set of word forms that only vary inflectionally among each other is called the *lexeme*. A *lemma* is a specific word form chosen to represent the lexeme word set; for example, Arabic verb lemmas are third person masculine singular perfective. We explore using both the diacritized lemma and the undiacritized lemma (hereafter 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, indicating the "core" meaning of the word. The *pattern* is a generally complementary abstraction sometimes indicating sematic notions such causation and reflexiveness. We use the pattern of the lemma, not of the word form. We group the ROOT, PATTERN, LEMMA and LMM in our discussion as *lexical features*. Nominal lexemes can also be classified into two groups: rational (i.e., human) or irrational (i.e.,, non-human).[2] The rationality feature interacts with syntactic agreement and other inflectional features (discussed next); as such, we group it with those features in this paper's experiments.

The *inflectional features* define the the space of variations of the word forms associated with a lexeme. 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). Inflectional features abstract away from the specifics of morpheme forms. Some inflectional features affect more than one morpheme in the same word. 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.

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

[2]Note that rationality ('human-ness' 'عاقل/غير عاقل') is narrower than animacy; its expression is wide-spead in Arabic, but less so English, where it mainly shows in pronouns (*he/she* vs. *it*) and relativizers (*the student who...* vs. *the desk/bird which...*).

Surface vs. functional features Additionally, some inflectional features, specifically gender and number, are expressed using different morphemes in different words (even within the same part-of-speech). There are four *sound* gender-number suffixes in Arabic:[3] $+\phi$ *(null morpheme)* for masculine singular, ة+ $+\hbar$ for feminine singular, ون+ *+wn* for masculine plural and ات+ *+At* for feminine plural. Plurality can be expressed using *sound plural* suffixes or using a pattern change together with *singular* suffixes. A sound plural example is the word pair حفيدة/حفيدات *Hafiyd+aħ/Hafiyd+At* 'granddaughter/granddaughters'. On the other hand, the plural of the inflectionally and morphemically feminine singular word مدرسة *madras+aħ* 'school' is the word مدارس *madAris+*ϕ 'schools', which is feminine and plural inflectionally, but has a masculine singular suffix. This irregular inflection, known as *broken plural*, is similar to the English *mouse/mice*, but is much more common in Arabic (over 50% of plurals in our training data). A similar inconsistency appears in feminine nouns that are not inflected using *sound* gender suffixes, e.g., the feminine form of the masculine singular adjective أزرق *Âzraq+*ϕ 'blue' is زرقاء *zarqA'+*ϕ not أزرقة* **Âzraq+aħ*. To address this inconsistency in the correspondence between inflectional features and morphemes, and inspired by (Smrž, 2007), we distinguish between two types of inflectional features: surface (or form-based)[4] features and functional features.

Most available Arabic NLP tools and resources model morphology using surface inflectional features and do not mark rationality; this includes the PATB (Maamouri et al., 2004), the Buckwalter morphological analyzer (BAMA) (Buckwalter, 2004) and tools using them such as the Morphological Analysis and Disambiguation for Arabic (MADA) system (Habash and Rambow, 2005). The Elixir-FM analyzer (Smrž, 2007) readily provides the functional inflectional number feature, but not full functional gender (only for adjectives and verbs but not for nouns), nor rationality. Most recently, Alkuhlani and Habash (2011) present a version of the PATB (part 3) that is annotated for functional gender, number and rationality features for Arabic. We use this resource in modeling these features in Section 5.5.

Morpho-syntactic interactions Inflectional features and rationality interact with syntax in two ways. In **agreement relations**, two words in a specific syntactic configuration have coordinated values for specific sets of features. MSA has standard (i.e., matching value) agreement for subject-verb pairs on PERSON, GENDER, and NUMBER, and for noun-adjective pairs on NUMBER, GENDER, CASE, and DET. There are three very common cases of exceptional agreement: verbs preceding subjects are always singular, adjectives of irrational plural nouns are always feminine singular, and verbs whose subjects are irrational plural are also always feminine singular. See the example in Figure 1: the adjective, الذكيات *Alðkyat* 'smart', of the feminine plural (and rational) حفيدات *HafiydAt* 'granddaughters' is feminine plural; but the adjective, الحكومية *AlHkwmyħ* 'the-governmental', of the feminine plural (and irrational) مدارس *madAris* 'schools' is feminine singular. These agreement rules always refer to functional morphology categories; they are orthogonal to the morpheme-feature inconsistency discussed above.

MSA exhibits **marking relations** in CASE and STATE marking. Different types of dependents have different CASE, e.g., verbal subjects are always marked NOMINATIVE. CASE and STATE are rarely explicitly manifested in undiacritized MSA. The DET feature plays an important role in distinguishing between the N-N *idafa* (possessive) construction, in which only the last noun may bear the definite article, and the N-A modifier construction, in which both elements generally exhibit agreement in definiteness.

Lexical features do not constrain syntactic structure as inflectional features do. Instead, bilexical dependencies are used to model semantic relations which often are the only way to disambiguate among different possible syntactic structures. Lexical abstraction also reduces data sparseness.

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 com-

[3] We ignore duals, which are regular in Arabic, and case/state variations in this discussion for simplicity.

[4] Smrž (2007) uses the term *illusory* for surface features.

parison, the tagset of the Buckwalter Morphological Analyzer (Buckwalter, 2004) used in the PATB has a core POS set of 44 tags (before morphological extension). 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, this tag 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 PATB-tokenized 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 morphologically enriched tagset, but instead they selectively enrich the core POS tagset with only certain morphological features. A full dicussion of how these tagsets affect parsing is presented in Marton et al. (2010); we summarize the main points here.

The following are the various tagsets we use in this paper: **(a)** the core POS tagset CORE12; **(b)** the CATiB treebank tagset CATIBEX, a newly introduced extension of CATIB6 (Habash and Roth, 2009) by simple regular expressions of the word form, indicating particular morphemes such as the prefix ال *Al+* or the suffix ون *+wn*; this tagset is the best-performing tagset for Arabic on predicted values. **(c)** the PATB full tagset (BW), size ≈2000+ (Buckwalter, 2004); We only discuss here the best performing tagsets (on predicted values), and BW for comparison.

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 got similar results for CASE in the gold experimental setting (Marton et al., 2010) 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 2). 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; Green and Manning, 2010), the Prague Dependency Treebank (PADT) (Buchholz and Marsi, 2006; Nivre, 2008) and the Columbia Arabic Treebank (CATiB) (Habash and Roth, 2009). Recently, Green and Manning (2010) analyzed the PATB for annotation consistency, and introduced an enhanced split-state constituency grammar, including labels for short *Idafa* constructions and verbal or equational clauses.

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, even though all the experiments reported here were performed using MaltParser. Our results agree with previous 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 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 using MaltParser to show gains using agreement-oriented features (Marton et al., 2010), and the first to use functional features for this task (this paper).

5 Experiments

Throughout this section, we only report results using *predicted* input feature values (e.g., generated automatically by a POS tagger). After presenting the parser we use (Section 5.1), we examine a large space of settings in the following order: the contribution of numerous inflectional features in a controlled fashion (Section 5.2);[5] the contribution of the lexical features in a similar fashion, as well as the combination of lexical and inflectional features (Section 5.3); an extension of the DET feature (Section 5.4); using functional NUMBER and GENDER feature values, as well as the RATIONALITY feature (Section 5.5); finally, putting best feature combinations to test with the best-performing POS tagset, and on an unseen test set (Section 5.6). All results are reported mainly in terms of labeled attachment accuracy score (parent word and the dependency relation to it, a.k.a. LAS). Unlabeled attachment accuracy score (UAS) is also given. We use McNemar's statistical significance test as implemented by Nilsson and Nivre (2008), and denote $p < 0.05$ and $p < 0.01$ with $^{+}$ and $^{++}$, respectively.

5.1 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.[6] For training, development and testing, we follow the splits used by Roth et al. (2008) for PATB part 3 (Maamouri et al., 2004). We 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),[7] 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 stk[0], word form at buf[0], etc. Kübler et al. (2009) describe a "typical" MaltParser model configuration of attributes and features.[8] Starting with it, in a series of initial controlled experiments, we settled on using buf[0-1] + stk[0-1] for wordforms, and buf[0-3] + stk[0-2] for POS tags. For features of new MaltParser-attributes (discussed later), we used buf[0] + stk[0]. We did not change the features for 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.[9] All experiments reported below were conducted using this new configuration.

5.2 Inflectional features

In order to explore the contribution of inflectional and lexical information in a controlled manner, we focused on the best performing core ("morphology-free") POS tagset, CORE12, as baseline; using three

[5] In this paper, we do not examine the contribution of different POS tagsets, see Marton et al. (2010) for details.

[6] Nivre (2008) reports that non-projective and pseudo-projective algorithms outperform the "eager" projective algorithm in MaltParser, but our training data did not contain any non-projective dependencies. The Nivre "standard" algorithm is also reported to do better on Arabic, but in a preliminary experimentation, it did similarly or slightly worse than the "eager" one, perhaps due to high percentage of right branching (left headed structures) in our Arabic training set – an observation already noted in Nivre (2008).

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

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

[9] We also experimented with normalizing word forms (*Alif Maqsura* conversion to *Ya*, and *hamza* removal from *Alif* forms) as is common in parsing and statistical machine translation literature – but it resulted in a similar or slightly decreased performance, so we settled on using non-normalized word forms.

setup		LAS	LAS_{diff}	UAS
All	**CORE12**	78.68	—	82.48
	+ all inflectional features	77.91	-0.77	82.14
Sep	+DET	**79.82**$^{++}$	**1.14**	83.18
	+STATE	79.34^{++}	0.66	82.85
	+GENDER	78.75	0.07	82.35
	+PERSON	78.74	0.06	82.45
	+NUMBER	78.66	-0.02	82.39
	+VOICE	78.64	-0.04	82.41
	+ASPECT	78.60	-0.08	82.39
	+MOOD	78.54	-0.14	82.35
	+CASE	75.81	-2.87	80.24
Greedy	+DET+STATE	79.42^{++}	0.74	82.84
	+DET+GENDER	79.90^{++}	1.22	83.20
	+DET+GENDER+PERSON	79.94^{++}	1.26	83.21
	+DET+PNG	**80.11**$^{++}$	**1.43**	83.29
	+DET+PNG+VOICE	79.96^{++}	1.28	83.18
	+DET+PNG+ASPECT	80.01^{++}	1.33	83.20
	+DET+PNG+MOOD	80.03^{++}	1.35	83.21

Table 1: CORE12 with inflectional features, predicted input. Top: Adding all nine features to CORE12. Second part: Adding each feature separately, comparing difference from CORE12. Third part: Greedily adding best features from second part.

different setups, we added nine morphological features with values predicted by MADA: DET (presence of the definite determiner), PERSON, ASPECT, VOICE, MOOD, GENDER, NUMBER, STATE (morphological marking as head of an *idafa* construction), 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 the MADA features, one at a time; 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 if it helped, or discarding it otherwise, and continuing through the least gainful feature. See Table 1.

Somewhat surprisingly, setup *All* hurts performance. This can be explained if one examines the prediction accuracy of each feature (top of Table 2). Features which are not predicted with very high accuracy, such as CASE (86.3%), can dominate the negative contribution, even though they are top contributors when provided as gold input (Marton et al., 2010); when all features are provided as gold input, *All* actually does better than individual features, which puts to rest a concern that its decrease here

feature	acc	set size
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
normalized word form (A,Y)	99.3	29737
non-normalized word form	98.9	29980

Table 2: Feature prediction accuracy and set sizes. * = The set includes a "N/A" value.

setup		LAS	LAS_{diff}	UAS
All	**CORE12 (repeated)**	78.68	—	82.48
	+ all lexical features	78.85	0.17	82.46
Sep	+LMM	**78.96**$^{+}$	**0.28**	82.54
	+ROOT	78.94^{+}	0.26	82.64
	+LEMMA	78.80	0.12	82.42
	+PATTERN	78.59	-0.09	82.39
Greedy	+LMM+ROOT	79.04^{++}	0.36	82.63
	+LMM+ROOT+LEMMA	**79.05**$^{++}$	**0.37**	82.63
	+LMM+ROOT+PATTERN	78.93	0.25	82.58

Table 3: Lexical features. Top part: Adding each feature separately; difference from CORE12 (predicted). Bottom part: Greedily adding best features from previous part.

is due to data sparseness. Here, when features are predicted, the DET feature (determiner), followed by the STATE (construct state, *idafa*) feature, are top individual contributors in setup *Sep*. Adding DET and the so-called ϕ-features (PERSON, NUMBER, GENDER, also shorthanded PNG) in the *Greedy* setup, yields 1.43% gain over the CORE12 baseline.

5.3 Lexical features

Next, we experimented with adding the lexical features, which involve semantic abstraction to some degree: LEMMA, LMM (the undiacritized lemma), and ROOT. 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, closely followed by ROOT in *Sep*. CORE12+LMM+ROOT (with or with-

CORE12 + ...	LAS	LAS_{diff}	UAS
+DET+PNG (repeated)	**80.11**$^{++}$	1.43	83.29
+DET+PNG+LMM	**80.23**$^{++}$	**1.55**	83.34
+DET+PNG+LMM +ROOT	80.10^{++}	1.42	83.25
+DET+PNG+LMM +PATTERN	80.03^{++}	1.35	83.15

Table 4: Inflectional+lexical features together.

CORE12 + ...	LAS	LAS_{diff}	UAS
+DET (repeated)	79.82^{++}	—	83.18
+DET2	**80.13**$^{++}$	0.31	83.49
+DET+PNG+LMM (repeated)	**80.23**$^{++}$	—	83.34
+DET2+PNG+LMM	80.21^{++}	-0.02	83.39

Table 5: Extended inflectional features.

out LEMMA) was the best greedy combination in setup *Greedy*. See Table 3. All lexical features are predicted with high accuracy (bottom of Table 2).

Following the same greedy heuristic, we augmented the best inflection-based model CORE12+DET+PNG with lexical features, and found that only the undiacritized lemma (LMM) alone improved performance (80.23%). See Table 4.

5.4 Inflectional feature engineering

So far we experimented with morphological feature values as predicted by MADA. However, it is likely that from a machine-learning perspective, representing similar categories with the same tag may be useful for learning. Therefore, we next experimented with modifying inflectional features that proved most useful.

As DET may help distinguish the N-N *idafa* construction from the N-A modifier construction, we attempted modeling also the DET values of previous and next elements (as MaltParser's stk[1] + buf[1], in addition to stk[0] + buf[0]). This variant, denoted DET2, indeed helps: when added to the CORE12, DET2 improves non-gold parsing quality by more than 0.3%, compared to DET (Table 5). This improvement unfortunately does not carry over to our best feature combination to date, CORE12+DET+PNG+LMM. However, in subsequent feature combinations, we see that DET2 helps again, or at least, doesn't hurt: LAS goes up by 0.06% in conjunction with features LMM+PERSON +FN*NGR in Table 6.

CORE12 + ...	LAS	LAS_{diff}	UAS
CORE12 (repeated)	78.68	–	82.48
+PERSON (repeated)	78.74	0.06	82.45
+GENDER (repeated)	78.75	0.07	82.35
+NUMBER (repeated)	78.66	-0.02	82.39
+FN*GENDER	78.96^{++}	0.28	82.53
+FN*NUMBER	78.88^{+}	0.20	82.53
+FN*NUMDGTBIN	78.87	0.19	82.53
+FN*RATIONALITY	78.91^{+}	0.23	82.60
+FN*GNR	79.32^{++}	0.64	82.78
+PERSON+FN*GNR	79.34^{++}	0.66	82.82
+DET+LMM+PERSON+FN*NGR	80.47^{++}	1.79	83.57
+DET2+LMM+PERSON+FN*NGR	**80.53**$^{++}$	1.85	83.66
+DET2+LMM+PERSON+FN*NG	80.43^{++}	1.75	83.56
+DET2+LMM+PNG+FN*NGR	80.51^{++}	1.83	83.66
CATIBEX	79.74	–	83.30
+DET2+LMM +PERSON+FN*NGR	**80.83**$^{++}$	1.09	84.02
BW	72.64	–	77.91
+DET2+LMM +PERSON+FN*NGR	**74.40**$^{++}$	1.76	79.40

Table 6: Functional features: gender, number, rationality.

We also experimented with PERSON. We changed the values of proper names from "N/A" to "3" (third person), but it resulted in a similar or slightly decreased performance, so it was abandoned.

5.5 Functional feature values

The NUMBER and GENDER features we have used so far only reflect *surface* (as opposed to *functional*) values, e.g., broken plurals are marked as singular. This might have a negative effect on learning generalizations over the complex agreement patterns in MSA (see Section 3), beyond memorization of word pairs seen together in training.

Predicting functional features To predict functional GENDER, functional NUMBER and RATIONALITY, we build a simple maximum likelihood estimate (MLE) model using these annotations in the corpus created by Alkuhlani and Habash (2011). We train using the same training data we use throughout this paper. For all three features, we select the most seen value in training associated with the triple *word*-CATIBEX-*lemma*; we back off to CATIBEX-*lemma* and then to *lemma*. For gender and number, we further back off to the surface values; for rationality, we back off to the most common value (*irrational*). On our predicted dev set, the overall accuracy baseline of predicting correct functional *gender-number-rationality* using surface features is

85.1% (for all POS tags). Our MLE model reduces the error by two thirds reaching an overall accuracy of 95.5%. The high accuracy may be a result of the low percentage of words in the dev set that do not appear in training (around 4.6%).

Digit tokens (e.g., "4") are also marked singular by default. They don't show surface agreement, even though the corresponding number-word token (*Arbς ħ* 'four.fem.sing') would. We further observe that MSA displays complex agreement patterns with numbers (Dada, 2007). Therefore, we alternatively experimented with binning the digit tokens' NUMBER value accordingly:

- the number 0 and numbers ending with 00
- the number 1 and numbers ending with 01
- the number 2 and numbers ending with 02
- the numbers 3-10 and those ending with 03-10
- the numbers, and numbers ending with, 11-99
- all other number tokens (e.g., 0.35 or 7/16)

and denoted these experiments with NUMDGTBIN. Almost 1.5% of the tokens are digit tokens in the training set, and 1.2% in the dev set.[10]

Results using these new features are shown in Table 6. The first part repeats the CORE12 baseline. The second part repeats previous experiments with surface morphological features. The third part uses the new functional morphological features instead. The performance using NUMBER and GENDER increases by 0.21% and 0.22%, respectively, as we replace surface features with functional features. (Recall that there is no functional PERSON.) We then see that the change in the representation of digits does not help; in the large space of experiments we have performed, we saw some improvement through the use of this alternative representation, but not in any of the feature combinations that performed best and that we report on in this paper. We then use just the RATIONALITY feature, which results in an increase over the baseline. The combination of all three functional features (NUMBER, GENDER, RATIONALITY) provides for a nice cumulative effect. Adding PERSON hardly improves further.

In the fourth part of the table, we include the other features which we found previously to be helpful, namely DET and LMM. Here, using DET2 instead of DET (see Section 5.4) gives us a slight improvement, providing our best result using the CORE12 POS tagset: 80.53%. This is a 1.85% improvement over using only the CORE12 POS tags (an 8.7% error reduction); of this improvement, 0.3% absolute (35% relative) is due to the use of functional features. We then use the best configuration, but without the RATIONALITY feature; we see that this feature on its own contributes 0.1% absolute, confirming its place in Arabic syntax. In gold experiments which we do not report here, the contribution was even higher (0.6-0.7%). The last row in the fourth part of Table 6 shows that using both surface and functional variants of NUMBER and GENDER does not help (hurts, in fact); the functional morphology features carry sufficient information for syntactic disambiguation.

The last part of the table revalidates the gains achieved of the best feature combination using the two other POS tagsets mentioned in Section 3: CATIBEX (the best performing tagset with predicted values), and BW (the best POS tagset with gold values in Marton et al. (2010), but results shown here are with predicted values). The CATIBEX result of 80.83% is our overall best result. The result using BW reconfirms that BW is not the best tagset to use for parsing Arabic with current prediction ability.

5.6 Validating results on unseen test set

Once experiments on the development set were done, we ran the best performing models on the previously unseen test set (Section 5.1). Table 7 shows that the same trends hold on this set as well.

Model	LAS	LAS_{diff}	UAS
CATIBEX	78.46	—	81.81
+DET2+LMM+PER+FN*NGR	**79.45**$^{++}$	**0.99**	**82.56**

Table 7: Results on unseen test set for models which performed best on dev set – predicted input.

6 Error Analysis

We analyze the attachment accuracy by attachment type. We show the accuracy for selected attachment types in Table 8. Using just CORE12, we see that some attachments (subject, modifications) are harder than others (objects, idafa). We see that by

[10] We didn't mark the number-words since in our training data there were less than 30 lemmas of less than 2000 such tokens, so presumably their agreement patterns can be more easily learned.

Features	SBJ	OBJ	MN	MP	IDF	Tot.
CORE12	67.9	90.4	72.0	70.3	94.5	78.7
CORE12 + LMM	68.8	90.4	72.6	70.9	94.6	79.0
CORE12 + DET2 +LMM+PNG	71.7	91.0	74.9	72.4	95.5	80.2
CORE12 + DET2 +LMM+PERS +FN*NGR	72.3	91.0	76.0	73.3	95.4	80.5

Table 8: Error analysis: Accuracy by attachment type (selected): subject, object, modification by a noun, modification (of a verb or a noun) by a preposition, idafa, and overall results (which match previously shown results)

adding LMM, all attachment types improve a little bit; this is as expected, since this feature provides a slight lexical abstraction. We then add features designed to improve idafa and those relations subject to agreement, subject and nominal modification (DET2, PERSON, NUMBER, GENDER). We see that as expected, subject, nominal modification (MN), and idafa reduce error by substantial margins (error reduction over CORE12+LMM greater than 8%, in the case of idafa the error reduction is 16.7%), while object and prepositional attachment (MP) improve to a lesser degree (error reduction of 6.2% or less). We assume that the relations not subject to agreement (object and prepositional attachment) improve because of the overall improvement in the parse due to the improvements in the other relations.

When we move to the functional features, we again see a reduction in the attachments which are subject to agreement, namely subject and nominal modification (error reductions over surface features of 2.1% and 4.4%, respectively). Idafa decreases slightly (since this relation is not affected by the functional features), while object stays the same. Surprisingly, prepositional attachment also improves, with an error reduction of 3.3%. Again, we can only explain this by proposing that the improvement in nominal modification attachment has the indirect effect of ruling out some bad prepositional attachments as well.

In summary, we see that not only do morphological features – and functional morphology features in particular – improve parsing, but they improve parsing in the way that we expect: those relations subject to agreement improve more than those that are not.

Last, we point out that MaltParser does not model generalized feature checking or matching directly, i.e., it has not learned that certain syntactic relations require identical (functional) morphological feature values. The gains in parsing quality reflect that the MaltParser SVM classifier has learned that the pairing of specific morphological feature values – e.g., *fem.sing.* for both the verb and its subject – is useful, with no generalization from each specific value to other values, or to general pair-wise value matching.

7 Conclusions and Future Work

We explored the contribution of different morphological (inflectional and lexical) features to dependency parsing of Arabic. We find that definiteness (DET), ϕ-features (PERSON, NUMBER, GENDER), and undiacritized lemma (LMM) are most helpful for Arabic dependency parsing on predicted input. We further find that functional morphology features and rationality improve over surface morphological features, as predicted by the complex agreement rules of Arabic. To our knowledge, this is the first result in Arabic NLP that uses functional morphology features, and that shows an improvement over surface features.

In future work, we intend to improve the prediction of functional morphological features in order to improve parsing accuracy. We also intend to investigate how these features can be integrated into other parsing frameworks; we expect them to help independently of the framework. We plan to make our parser available to other researchers. Please contact the authors if interested.

Acknowledgments

This work was supported by the DARPA GALE program, contract HR0011-08-C-0110. We thank Joakim Nivre for his useful remarks, Otakar Smrž for his help with Elixir-FM, Ryan Roth and Sarah Alkuhlani for their help with data, and three anonymous reviewers for useful comments. Part of the work was done while the first author was at Columbia University.

References

Sarah Alkuhlani and Nizar Habash. 2011. A corpus for modeling morpho-syntactic agreement in Arabic: gender, number and rationality. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, Portland, Oregon, USA.

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 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.

Ali Dada. 2007. Implementation of Arabic numerals and their syntax in GF. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pages 9–16, Prague, Czech Republic.

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.

Spence Green and Christopher D. Manning. 2010. Better Arabic parsing: Baselines, evaluations, and analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, pages 394–402, Beijing, China.

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 Association for Computational Linguistics (ACL)*, pages 573–580, Ann Arbor, Michigan.

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, Reem Faraj, and Ryan Roth. 2009. Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of MEDAR International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Nizar Habash. 2010. *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.

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.

Yuval Marton, Nizar Habash, and Owen Rambow. 2010. Improving Arabic dependency parsing with inflectional and lexical morphological features. In *Proceedings of Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL) at the 11th Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL) - Human Language Technology (HLT)*, Los Angeles, USA.

Jens Nilsson and Joakim Nivre. 2008. MaltEval: An evaluation and visualization tool for dependency parsing. In *Proceedings of the sixth International Conference on Language Resources and Evaluation (LREC)*, Marrakech, Morocco.

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.

Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of ACL-08: HLT, Short Papers*, pages 117–120, Columbus, Ohio, June. Association for Computational Linguistics.

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.