Dependency Parsing of Modern Standard Arabic with Lexical and Inflectional Features

Yuval Marton*
Nuance Communications

Nizar Habash**
Center for Computational Learning
Systems, Columbia University

Owen Rambow†
Center for Computational Learning
Systems, Columbia University

We explore the contribution of lexical and inflectional morphology features to dependency parsing of Arabic, a morphologically rich language with complex agreement patterns. Using controlled experiments, we contrast the contribution of different part-of-speech (POS) tag sets and morphological features in two input conditions: machine-predicted condition (in which POS tags and morphological feature values are automatically assigned), and gold condition (in which their true values are known). We find that more informative (fine-grained) tag sets are useful in the gold condition, but may be detrimental in the predicted condition, where they are outperformed by simpler but more accurately predicted tag sets. We identify a set of features (definiteness, person, number, gender, and undiacritized lemma) that improve parsing quality in the predicted condition, whereas other features are more useful in gold. We are the first to show that functional features for gender and number (e.g., "broken plurals"), and optionally the related rationality ("humanness") feature, are more helpful for parsing than form-based gender and number. We finally show that parsing quality in the predicted condition can dramatically improve by training in a combined gold+predicted condition. We experimented with two transition-based parsers, MaltParser and Easy-First Parser. Our findings are robust across parsers, models, and input conditions. This suggests that the contribution of the linguistic knowledge in the tag sets and features we identified goes beyond particular experimental settings, and may be informative for other parsers and morphologically rich languages.

1. Introduction

For Arabic—as for other morphologically rich languages—the role of morphology is often expected to be essential in syntactic modeling, and the role of word order is less important than in morphologically poorer languages such as English. Morphology

* Nuance Communications, 505 First Ave. S, Suite 700, Seattle, WA 98104. E-mail: yuvalmarton@gmail.com.
** Center for Computational Learning, Columbia University. E-mail: habash@ccls.columbia.edu.
† Center for Computational Learning, Columbia University. E-mail: rambow@ccls.columbia.edu.

Submission received: October 1, 2011; revised submission received: June 16, 2012; accepted for publication: August 3, 2012.

interacts with syntax in two ways: agreement and assignment. In **agreement**, there is coordination between the morphological features of two words in a sentence based on their syntactic configuration (e.g., subject–verb or noun–adjective agreement in GENDER and/or NUMBER). In **assignment**, specific morphological feature values are assigned in certain syntactic configurations (e.g., CASE assignment for the subject or direct object of a verb).[1]

Parsing model design aims to come up with features that best help parsers learn the syntax and choose among different parses. The choice of optimal linguistic features depends on three factors: relevance, redundancy, and accuracy. A feature has **relevance** if it is useful in making an attachment (or labeling) decision. A particular feature may or may not be relevant to parsing. For example, the GENDER feature may help parse the Arabic phrase باب السيارة الجديد / الجديدة *bAb AlsyArħ Aljdyd/Aljdydħ* ('door the-car the-new$_{masc.sg/fem.sg}$ [lit.]),[2] using syntactic agreement: if *the-new* is masculine (الجديد), it should attach to the masculine *door*, resulting in the meaning 'the car's new door'; if *the-new* is feminine (الجديدة), it should attach to the feminine *the-car*, resulting in 'the door of the new car.' Conversely, the ASPECT feature does not constrain any syntactic decision. Even if relevant, a feature may not necessarily contribute to optimal performance because it may be **redundant** with other features that surpass it in relevance. For example, as we will see, the DET and STATE features alone both help parsing because they help identify the *idafa* construction, but they are redundant with each other and the DET feature is more helpful because it also helps with adjectival modification of nouns. Finally, the **accuracy** of automatically predicting the feature values (ratio of correct predictions out of all predictions) of course affects the value of a feature on unseen text. Even if relevant and non-redundent, a feature may be hard to predict with sufficient *accuracy* by current technology, in which case it will be of little or no help for parsing, even if helpful when its gold values are provided. As we will see, the CASE feature is very relevant and not redundant, but it cannot be predicted with high accuracy and overall it is not useful.

Different languages vary with respect to which features may be most helpful given various tradeoffs among these three factors. In the past, it has been shown that if we can recognize the relevant morphological features in assignment configurations well enough, then they contribute to parsing accuracy. For example, modeling CASE in Czech improves Czech parsing (Collins et al. 1999): CASE is relevant, not redundant, and can be predicted with sufficient accuracy. It has been more difficult showing that agreement morphology helps parsing, however, with negative results for dependency parsing in several languages (Eryigit, Nivre, and Oflazer 2008; Nivre, Boguslavsky, and Iomdin 2008; Nivre 2009).

In this article we investigate morphological features for dependency parsing of Modern Standard Arabic (MSA). For MSA, the space of possible morphological features is fairly large. We determine which morphological features help and why. We further determine the upper bound for their contribution to parsing quality. Similar to previous

1 Other morphological features, such as MOOD or ASPECT, do not interact with syntax at all. Note also that we do not commit to a specific linguistic theory with these terms; hence, other theoretical terms such as the Minimalist *feature checking* may be used here just as well.

2 All Arabic transliterations are presented in the HSB transliteration scheme (Habash, Soudi, and Buckwalter 2007): (alphabetically) *Abtθ jHxdðrzsšSDTĎ ς γfqklmnhwy* and the additional symbols: ' ء, Â أ, Ă إ, Ā آ, ŵ ؤ, ŷ ئ, ى ,ħ ة, ý ى, a ـَ, u ـُ, i ـِ, ~ ـّ, ã ـً, ũ ـٌ, ĩ ـٍ.

results, assignment features, specifically CASE, are very helpful in MSA, though only under gold conditions: Because CASE is rarely explicit in the typically undiacritized written MSA, it has a dismal accuracy rate, which makes it useless when used in a machine-predicted (real, non-gold) condition. In contrast with previous results, we show agreement features are quite helpful in both gold and predicted conditions. This is likely a result of MSA having a rich agreement system, covering both verb–subject and noun–adjective relations. The result holds for both the MaltParser (Nivre 2008) and the Easy-First Parser (Goldberg and Elhadad 2010).

Additionally, almost all work to date in MSA morphological analysis and part-of-speech (POS) tagging has concentrated on the morphemic form of the words. Often, however, the functional morphology (which is relevant to agreement, and relates to the meaning of the word) is at odds with the "surface" (form-based) morphology; a well-known example of this are the "broken" (irregular) plurals of nominals. We show that by modeling the functional morphology rather than the form-based morphology, we obtain a further increase in parsing performance (again, both when using gold and when using predicted POS and morphological features). To our knowledge, this work is the first to use functional morphology features in MSA processing.

As a further contribution of this article, we show that for parsing with predicted POS and morphological features, training on a combination of gold and predicted POS and morphological feature values outperforms the alternative training scenarios.

The article is structured as follows. We first present relevant Arabic linguistic facts, their representation in the annotated corpus we use, and variations of abstraction thereof in several POS tag sets (Section 2). We follow with a survey of related work (Section 3), and describe our basic experiments in Section 4. We first explore the contribution of various POS tag sets, (form-based) morphological features, and promising combinations thereof, to Arabic dependency parsing quality—in straightforward feature engineering design and combination heuristics. We also explore more sophisticated feature engineering for the determiner (DET) feature. In Section 5, we proceed to an extended exploration of functional features. This includes using functional NUMBER and GENDER feature values, instead of form-based values; using the non-form-based rationality (RAT) feature; and combinations thereof. We additionally consider the applicability of our results to a different parser (Section 6) and consider combining gold and predicted data for training (Section 7). Section 8 presents a result validation on unseen test data, as well as an analysis of parsing error types under different conditions. We conclude and provide a download link to our model in Section 9. Last, we include an appendix with further explorations of PERSON feature engineering, "binning" of Arabic number constructions according to their complex syntactic patterns, and embedding useful morphological features in the POS tag set. Much of Sections 2–5 was presented in two previous publications (Marton, Habash, and Rambow 2010, 2011). This article extends that previous work by:

1. evaluating *all* our parsing models in both gold and non-gold conditions (where before this was true for only select models in Sections 4–5),
2. using a newer version of our Arabic functional morphology resource (Section 5),
3. evaluating several of our most notable parsing models with an additional parser (Section 6),
4. exploring two additional training methods, as already mentioned above (Section 7), and

5. providing an extended discussion and comparison of several notable and best performing models, including analyses of their performance per dependency tag (Section 8).

2. Experimental Data and Relevant Linguistic Concepts

In this section, we present the linguistic concepts relevant to our discussion of Arabic parsing, and the data we use for our experiments. We start with the central concept of the morpheme followed by the more abstract concepts of the lexeme and lexical and inflectional features. Throughout this section, we use the term *feature* in its linguistic sense, as opposed to its machine learning sense that we use in Section 4. Discussions of the challenges of form-based (morpheme-based) versus functional features on the one hand, and morpho-syntactic interactions on the other hand, follow. Finally, we present the annotated corpus we use, and the various POS tag sets, that are extracted from this corpus (in varying degrees of abstraction and lexicalization), and which we use in the rest of the article.

2.1 Morphemes

Words can be described in terms of their morphemes (atomic units bearing meaning); in Arabic, in addition to concatenative prefixes and suffixes, there are templatic (non-contiguous) morphemes called **root** and **pattern**. The root is typically a triplet of consonants (a.k.a. *radicals*). The pattern is a template made of vowels, sometimes additional consonants, and place-holders for the root radicals. The root conveys some base meaning, which patterns may modify in various ways. A combination of a root and a pattern is called a **stem**. More on root and pattern can be found in Section 2.2. Arabic also includes a set of clitics that are tokenized in all Arabic treebanks, with the exception of the Arabic definite article, ال *Al+* ('the'), which is kept attached to the stem. We consider the definite article a prefix, and its presence affects the value of the DET feature in models containing it (see Section 4.3). An example of morphological analysis to the level of morphemes is the word يكاتبون *yu+kAtib+uwn* ('they correspond'); it has one prefix and one suffix (which at a deeper level may be viewed together as one circumfix), in addition to a stem composed of the root كتب *k-t-b* ('writing related') and the pattern *1A2i3*.[3]

2.2 Lexeme, Lexical Features, and Inflectional Features

Arabic words can also be described in terms of lexemes and inflectional features. We define the **lexeme** as the set of word forms that only vary inflectionally among each other. A **lemma** is one of these word forms, used for representing the lexeme word set. For example, Arabic verb lemmas are third-person masculine singular perfective. We explore using both a diacritized LEMMA feature, and an 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 semantic notions such as

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

causation and reflexiveness, among other things. 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* (see Section 4.4). Nominal lexemes can be further classified into two groups: denoting rational (i.e., human) entities, or irrational (i.e., non-human) entities. The rationality (or RAT) feature interacts with syntactic agreement and other inflectional features (discussed next); as such, we group it with those features in this article.

The *inflectional features* define the space of variations of the word forms associated with a lexeme. Words[4] vary along nine dimensions: GENDER, NUMBER, and PERSON (for nominals and verbs); ASPECT, VOICE, and MOOD (for verbs); and CASE, STATE (construct state, *idafa*), 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 earlier example from imperfective to perfective yields the word form كاتبوا *kAtab+uwA* ('they corresponded'), which differs in terms of prefix, suffix, and pattern.

2.3 Form-Based vs. Functional Features

Some inflectional features, specifically gender and number, are expressed using different morphemes in different words (even within the same POS). There are four *sound* gender-number suffixes in Arabic:[5] +ϕ *(null morpheme)* for masculine singular, ة+ +*ħ* for feminine singular, ون+ +*wn* for masculine plural, and ات+ +*At* for feminine plural. *Form-based* GENDER and NUMBER feature values are set only according to these four morphemes (and a few others, ignored for simplicity). There are exceptions and alternative ways to express GENDER and NUMBER, however, and *functional* feature values take them into account: Depending on the lexeme, 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 nominals that are not inflected using sound gender suffixes, for example, 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: *form-based* (a.k.a. surface, or illusory) features and *functional* features.[6]

Most available Arabic NLP tools and resources model morphology using form-based ("surface") inflectional features, and do not mark rationality; this includes the Penn Arabic Treebank (PATB) (Maamouri et al. 2004), the Buckwalter morphological analyzer (Buckwalter 2004), and tools using them such as the Morphological Analysis and Disambiguation for Arabic (MADA) toolkit (Habash and Rambow 2005; Habash, Rambow, and Roth 2012). The Elixir-FM analyzer (Smrž 2007) readily provides the

4 PATB-tokenized words; see Section 2.5.
5 We ignore duals, which are regular in Arabic, and case/state variations in this discussion for simplicity.
6 Note that the functional and form-based feature values for verbs always coincide.

functional inflectional number feature, but not full functional gender (only for adjectives and verbs but not for nouns), nor rationality. In this article, we use an in-house system which provides functional gender, number, and rationality features (Alkuhlani and Habash 2012). See Section 5.2 for more details.

2.4 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, however, 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. This exceptional agreement is orthogonal to the form-function inconsistency discussed earlier. In other words, having a sound or broken plural has no bearing on whether the noun is rational or not—and hence whether an adjectival modifier should agree with it by being feminine-singular or -plural. Note also that all agreement rules, including the exceptional agreement rules, refer to functional number and gender, not to form-based number and gender.

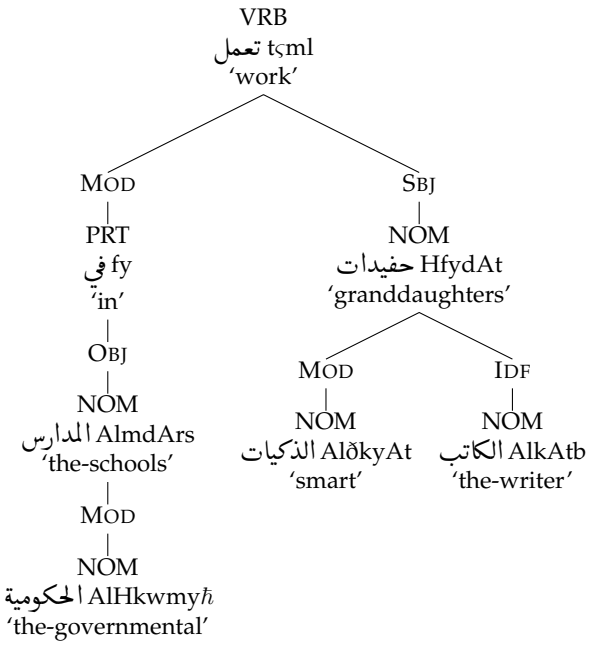

Figure 1
CATiB Annotation example. تعمل حفيدات الكاتب الذكيات في المدارس الحكومية *tçml HfydAt AlkAtb AlðkyAt fy AlmdArs AlHkwmyħ* ('The writer's smart granddaughters work for public schools'). The words in the tree are presented in the Arabic reading direction (from right to left).

MSA exhibits *assignment relations* in CASE and STATE marking. Different types of dependents have different CASE, for example, verbal subjects are always marked NOMINATIVE (for a discussion of case in MSA, see Habash et al. [2007]). STATE is a marker on nouns; when a noun heads an *idafa* construction, its STATE is ('construct'). CASE and STATE are rarely explicitly manifested in undiacritized MSA. The DET feature plays an important role in distinguishing between N-N construct (*idafa*), in which only the last noun bears the definite article,[7] and N-A (noun-adjectival modifier), in which both elements generally exhibit agreement in definiteness (and agreement in other features, too). Although only N-N may be followed by additional N elements in *Idafa* relation, both constructions may be followed by one or more adjectival modifiers.

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

2.5 Corpus, CATiB Format, and the CATIB6 POS Tag Set

We use the Columbia Arabic Treebank (CATiB) (Habash and Roth 2009). Specifically, we use the portion converted from Part 3 of the PATB 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 tag set consisting of six tags only (henceforth CATIB6). The tags are: **NOM** (non-proper nominals including nouns, pronouns, adjectives, and adverbs), **PROP** (proper nouns), **VRB** (active-voice verbs), **VRB-PASS** (passive-voice verbs), **PRT** (particles such as prepositions or conjunctions), and **PNX** (punctuation). CATiB uses 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 other PATB-based POS tag sets, see Sections 2.6 and 2.7.

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*+ ('the'). In all of the experiments reported in this article, we use the gold segmentation. Tokenization involves further decisions on the segmented token forms, such as spelling normalization, which we only briefly touch on here (in Section 4.1). An example CATiB dependency tree is shown in Figure 1. For the corpus statistics, see Table 1. For more information on CATiB, see Habash and Roth (2009) and Habash, Faraj, and Roth (2009).

2.6 Core POS Tag Sets

Linguistically, words have associated POS tags, e.g., "verb" or "noun," 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 tag set of the Buckwalter Morphological Analyzer (Buckwalter 2004) used in the PATB has a core POS set of 44 tags (CORE44) before morphological extension.[8] Cross-linguistically, a core set containing around 12 tags is often

7 We ignore the rare "false *idafa*" construction (Habash 2010, p. 102).

8 The 44 tags in CORE44 are based on the tokenized version of Arabic words. There are 34 untokenized core tags as used in MADA+TOKAN (Habash, Rambow, and Roth 2012).

assumed as a "universal tag set" (Rambow et al. 2006; Petrov, Das, and McDonald 2012). We have adapted the list from Rambow et al. (2006) for Arabic, and call it here CORE12. It contains the following tags: verb (V), noun (N), adjective (AJ), adverb (AV), proper noun (PN), pronoun (PRO), relative pronoun (REL), preposition (P), conjunction (C), particle (PRT), abbreviation (AB), and punctuation (PNX). The CATIB6 tag set can be viewed as a further reduction, with the exception that CATIB6 contains a passive voice tag (a morphological feature); this tag constitutes only 0.5% of the tags in the training, however.

2.7 Extended POS Tag Sets

The notion of "POS tag set" 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 tag set is still fairly small since English is morphologically impoverished). In PATB-tokenized MSA, the corresponding type of tag set (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 POS tags with complete morphology). Therefore, researchers have proposed tag sets for MSA whose size is similar to that of the English PTB tag set, as this has proven to be a useful size computationally. These tag sets are hybrids in the sense that they are neither simply the core POS, nor the complete morphologically enriched tag set, but instead they selectively enrich the core POS tag set with only certain morphological features. A more detailed discussion of the various available Arabic tag sets can be found in Habash (2010).

The following are the various tag sets we use in this article: (a) the core POS tag sets CORE44 and the newly introduced CORE12; (b) CATiB Treebank tag set (CATIB6) (Habash and Roth 2009) and its newly introduced extension of CATIBEX created using simple regular expressions on word form, indicating particular morphemes such as the prefix ال *Al+* or the suffix ون *+wn*; this tag set is the best-performing tag set for Arabic on predicted values as reported in Section 4; (c) the PATB full tag set with complete morphological tag (BW) (Buckwalter 2004); and two extensions of the PATB reduced tag set (PENN POS, a.k.a. RTS, size 24 [Diab, Hacioglu, and Jurafsky 2004]), both outperforming it: (d) Kulick, Gabbard, and Marcus (2006)'s tag set (KULICK), size 43, one of whose most important extensions is the marking of the definite article clitic, and (e) Diab and Benajiba's (in preparation) EXTENDED RTS tag set (ERTS), which marks gender, number, and definiteness, size 134.

3. 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 tag set for parsing Czech consists of a basic POS tag plus a CASE feature (when applicable). This tag set (size 58) outperforms the basic Czech POS tag set (size 13) and the complete tag set (size ≈3000+). They also report that the use of gender, number, and person features did not yield any improvements. The results for Czech are the opposite of our results for Arabic, as we will see. This may be due to CASE tagging having a lower error rate in Czech (5.0%) (Hajič and Vidová-Hladká 1998) compared with 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 (MSA), results have been reported on the PATB (Kulick, Gabbard, and Marcus 2006; Diab 2007; Green and Manning 2010), the Prague Dependency Treebank (PADT) (Buchholz and Marsi 2006; Nivre 2008) and the 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 treebank representations, even though all the experiments reported here were performed using the MaltParser.

Our results agree with previous work on Arabic and Hebrew in that marking the definite article is helpful for parsing. We go beyond previous work, however, and explore additional lexical and inflectional features. Previous work with MaltParser in Russian, Turkish, and Hindi showed gains with CASE but not with agreement features (Eryigit, Nivre, and Oflazer 2008; Nivre, Boguslavsky, and Iomdin 2008; Nivre 2009). Our work is the first to show gains using agreement in MaltParser and in Arabic dependency parsing, and the first to use functional features for this task. Furthermore, we demonstrate that our results carry over successfully to another parser, the Easy-First Parser (Goldberg and Elhadad 2010) (Section 6).

Hohensee and Bender (2012) have conducted a study on dependency parsing for 21 languages using features that encode whether the values for certain attributes are equal or not for a node and its governor. These features are potentially powerful, because they generalize to the very notion of agreement, away from the specific values of the attributes on which agreement occurs.[9] We expect this kind of feature to yield lower gains for Arabic, unless:

- one uses functional feature values (such as those used here for the first time in Arabic NLP),
- one uses yet another representation level to account for the otherwise non-identity agreement patterns of irrational plurals,
- one handles the loss of overt number agreement in constructions such as VS (where the verb precedes its subject), and
- one adequately represents the otherwise "inverse" number agreement (a phenomenon common to other Semitic languages, such as Hebrew, too).

4. Basic Parsing Experiments

We examined a large space of settings. In all our experiments, we contrasted the results obtained using machine-predicted input with the results obtained using gold input (the

9 We do not relate to specific results in their study because it has been brought to our attention that Hohensee and Bender (2012) are in the process of rechecking their code for errors, and rerunning their experiments (personal communication).

upper bound for using these features). We started by looking at individual features (including POS tag sets) and their prediction accuracy. We then explored various feature combinations in a hill-climbing fashion. We examined these issues in the following order:

1. the contribution of *POS tag sets* to the parsing quality, as a function of the amount of information encoded in the tag set, using (a) gold input, and (b) machine-predicted POS tags;
2. the contribution of numerous *inflectional features* in a controlled fashion, using (c) gold input and (d) machine-predicted input; (e) the prediction accuracy of each inflectional feature;
3. the contribution of the *lexical features* in a similar fashion, again using (f) gold input and (g) predicted input; (h) the prediction accuracy of each lexical feature;
4. (i) certain *feature combinations* and (j) the embedding of the best combination in the POS tag set; and
5. (k) further *feature engineering* of select useful features.

In Section 5 we explore using functional (instead of form-based) feature values. In Section 6 we repeat key experiments with another parser, illustrating the robustness of our findings across these frameworks. In Section 7 we explore alternative training methods, and their impact on key models.

All results are reported mainly in terms of labeled attachment accuracy score (the parent word and the type of dependency relation to it, abbreviated as LAS), which is also used for greedy (hill-climbing) decisions for feature combination. Unlabeled attachment accuracy score (UAS) and label accuracy (dependency relation regardless of parent, LS) are also given. For statistical significance, we use McNemar's test on non-gold LAS, as implemented by Nilsson and Nivre (2008). We denote $p < 0.05$ and $p < 0.01$ with $^{+}$ and $^{++}$, respectively.

4.1 Data Sets and Parser

For all the experiments reported in this article, we used the training portion of PATB Part 3 v3.1 (Maamouri et al. 2004), converted to the CATiB Treebank format, as mentioned in Section 2.5. We used the same training / devtest split as in Zitouni, Sorensen, and Sarikaya (2006); and we further split the devtest into two equal parts: a development (dev) set and a blind test set. For all experiments, unless specified otherwise, we used the dev set.[10] We kept the test unseen ("blind") during training and model development. Statistics about this split (after conversion to the CATiB dependency format) are given in Table 1.

For all experiments reported in this section we used the syntactic dependency parser MaltParser v1.3 (Nivre 2003, 2008; Kübler, McDonald, and Nivre 2009), a transition-based parser with an input buffer and a stack, which uses SVM classifiers

10 We use the term "dev set" to denote a non-blind test set, used for model development (feature selection and feature engineering). We do not perform further weight optimization (which, if done, is done on a separate "tuning set").

Table 1
Penn Arabic Treebank part 3 v3.1 data split.

split	# tokens	# sentences	sentence length (avg. # tokens)
training	341,094	11,476	29.7
dev	31,208	1,043	29.9
unseen test	29,944	1,007	29.7
TOTAL	402,246	13,526	29.7

to predict the next state in the parse derivation. All experiments were done using the Nivre "eager" algorithm.[11]

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),[12] 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], and so on. Kübler, McDonald, and Nivre (2009) describe a "typical" MaltParser model configuration of attributes and features.[13] Starting with it, in a series of initial controlled experiments, we settled on using buf[0-1] + stk[0-1] for word-forms, 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: stk[0], ldep(stk[0]), rdep(stk[0]), ldep(buf[0]), rdep(buf[0]) (where ldep and rdep are the left and right, respectively, dependents of the specified argument). This new MaltParser configuration resulted in gains of 0.3–1.1% in labeled attachment accuracy (depending on the POS tag set) 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, but it resulted in a small decrease in performance, so we settled on using non-normalized word-forms. All experiments reported here were conducted using this new configuration. To recap, it has the following MaltParser attributes (machine learning features): 4 word-form attributes, 7 POS tag attributes, and 5 deprel attributes (some of which are not useful for the Nivre "eager" algorithm), totaling 16 attributes and two more for every new feature described in Section 4.3 and on (e.g., CASE).

11 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 there to do better on Arabic, but in a preliminary experimentation, it did slightly worse than the "eager" one, perhaps due to the high percentage of right branching (left headed structures) in our Arabic training set—an observation already noted in Nivre (2008).

12 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)*.

13 It is slightly different from the default configuration.

Table 2
Parsing performance with each POS tag set, on gold and predicted input. LAS = labeled attachment accuracy (dependency + relation). UAS = unlabeled attachment accuracy (dependency only). LS = relation label prediction accuracy. LAS diff = difference between labeled attachment accuracy on gold and predicted input. POS acc = POS tag prediction accuracy.

tag set	gold			predicted			gold-pred.	POS	tag set
	LAS	UAS	LS	LAS	UAS	LS	LAS diff.	acc.	size
CATIB6	81.0	83.7	92.6	78.3	82.0	90.6	**–2.7**	**97.7**	6
CATIBEX	82.5	85.0	93.4	**79.7**	**83.3**	**91.4**	−2.8	**97.7**	44
CORE12	82.9	85.4	93.5	78.7	82.5	90.6	−4.2	96.3	12
CORE44	82.7	85.2	93.3	78.4	82.2	90.4	−4.3	96.1	40
ERTS	83.0	85.2	93.8	78.9	82.6	91.0	−4.0	95.5	134
KULICK	83.6	**86.0**	94.0	79.4	83.2	91.1	−4.2	95.7	32
BW	**84.0**	85.8	**94.8**	72.6	77.9	86.5	−11.4	81.8	430

4.2 The Effect of POS Tag Richness on Parsing Quality

In this section, we compare the effect on parsing quality of a number of POS tag sets varying in their richness, in both gold and predicted settings.

Gold POS tag values. We turn first to the contribution of POS information to parsing quality, as a function of the amount of information encoded in the POS tag set (i.e., the *relevance* of a tag set). A first rough estimation for the amount of information is the actual tag set size, as it appears in the training data. For this purpose we compared the POS tag sets discussed in sections 2.6 and 2.7. In optimal conditions (using gold POS tags), the richest tag set (BW) is indeed the best performer (84.0%), and the poorest (CATIB6) is the worst (81.0%). Mid-size tag sets are in the high (82%), with the notable exception of KULICK, which does better than ERTS, in spite of having one fourth the tag set size; moreover, it is the best performer in unlabeled attachment accuracy (86.0%), in spite of being less than tenth the size of BW. Our extended mid-size tag set, CATIBEX, was a mid-level performer as expected. Columns 2–4 in Table 2 show results with gold input, and the rightmost column shows the number of tag types actually occurring in the training data.

Predicted POS tag values. 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.[14] The more informative the tag set, the less accurate the tag prediction might be, so the effect on overall parsing quality is unclear. Put differently, we are interested in the tradeoff between *relevance* and *accuracy*. Therefore, we repeated the experiments with POS tags predicted by the MADA toolkit (Habash and Rambow 2005; Habash, Rambow, and Roth 2012)[15] (see Table 2,

14 Some parsers predict POS tags internally, instead of receiving them as input, but this is not the case in this article.

15 We use MADA v3.1 in all of our experiments. We note that MADA v3.1 was tuned on the same development set that we use for making our parsing model choices; ideally, we would have chosen a different development set for our work on parsing, but we thought it would be best to use MADA as a black box component (for past and future comparability), and did not have sufficient data to carve out from a second development set (while retaining a test set). We do not take this as a major concern for our results. In fact, although MADA was tuned to maximize its core POS accuracy (the untokenized version of CORE44), CORE44 did not yield best parsing quality on MADA-predicted input (see Table 2).

columns 5–7). It turned out that BW, the best gold performer but with lowest POS prediction accuracy (81.8%), suffered the biggest drop (11.4%) and was the worst performer with predicted tags. The simplest tag set, 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.

These results suggest that POS tag set accuracy is as important to parsing quality, if not more important, than its relevance. In other words, when designing a parsing model, one might want to consider that in the tradeoff, mediocre accuracy may be worse than mediocre relevance. Later we see a similar trend for other features as well (e.g., CASE in Section 4.3). In Section 7 we also present a training method that largely mitigates (but doesn't resolve) this issue of mediocre accuracy of relevant features.

4.3 Inflectional Features and Their Contribution to Parsing Quality

Experimenting with inflectional features is especially important in Arabic parsing, as it is morphologically rich. 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 tag set, CORE12, as baseline; using three different set-ups, we added nine inflectional features (with either gold values, or with values predicted by MADA): DET (presence of determiner), PERSON, ASPECT, VOICE, MOOD, GENDER, NUMBER, STATE, and CASE. For a brief reminder and examples for each feature, see the rightmost column in Table 3, or for more details refer back to Section 2.

In set-up *All*, we augmented the baseline model with all nine features (as nine additional MaltParser attributes); in set-up *Sep*, we augmented the baseline model with each of these features, one at a time, separately; and in set-up *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 repeating this heuristics through the least gainful feature. See Table 4.

Gold feature values. We applied the three setups (*All, Sep*, and *Greedy*) with gold POS tags and gold morphological tags, to examine the contribution of the morphological features in optimal conditions. The top left section of Table 4 shows that applying all inflectional features together yields gains over the baseline. Examining the contribution of each feature separately (second top left *Sep* section), we see that CASE, followed by STATE and DET, were the top contributors. Performance of CASE is the notable difference from the predicted conditions (see following discussion). No single feature outperformed the *All* set-up in gold. Surprisingly, only CASE and STATE helped in the *Greedy* set-up (85.4%, our highest result in gold), although one might expect feature DET to have helped, too (since it is highly relevant: It participates in agreement, and interacts with the *idafa* construction). This shows that there is *redundancy* in the information provided by DET on the one hand and CASE and STATE on the other, presumably because both sets of feature help identify the same construction, *idafa*.

Predicted feature values. We re-applied the three set-ups with predicted feature values (right-hand side half of Table 4). Set-up *All* hurts performance on the machine-predicted input. This can be explained if one examines the prediction accuracy of each feature (top half, third section of 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

Table 3
Prediction accuracy, value set sizes, descriptions, and value examples of features used in this work. Accuracy was measured over the development set. * = The set includes a "N/A" value(s).

feature	acc.	set size	comments and examples
normalized word-form	99.3	29,737	collapse certain spelling variations into a single representation, e.g., آلي *Āly* ('automatic') and إلى *Ālý* ('to') are collapsed into الي *Aly*
non-normalized word-form	98.9	29,980	'raw' input (except for PATB segmentation), e.g., the uncollapsed forms above
LEMMA (diacritized)	96.7	16,837	abstraction over inflected forms, e.g., the lemma of مَكَاتِب *makAtib* ('offices') is مَكتَب *maktab* ('office')
LMM	98.3	15,305	undiacritized lemma (lemma with vowels and other diacritics removed), e.g., مكتب *mktb* for the example above.
ROOT	98.4	9,646	further abstraction over inflection and patterns; typically a consonant triplet, a.k.a. radicals, e.g., ك ت ب *k t b* ('writing-related')
PATTERN	97.0	338	sequence of vowels and consonants with placeholders for the root radicals, e.g., *ma12a3* ('location-related'); typically a derivational modification to the basic meaning of the root, such as a location or instrument, but inflectional variations such as aspect, voice, number and gender also exist; we use the pattern of the lemma, not the inflected form, which may differ in cases such as broken plurals
DET	99.6	3*	presence of the determiner morpheme ال *Al*
PERSON	99.1	4*	first, second, or third person (or N/A)
ASPECT	99.1	5*	perfective, imperfective and imperative for verbs (or N/A)
VOICE	98.9	4*	active or passive voice for verbs (or N/A)
MOOD	98.6	5*	indicative, subjunctive, jussive for verbs (or N/A)
GENDER	99.3	3*	(form-based) masculine or feminine (or N/A)
NUMBER	99.5	4*	(form-based) singular, dual, or plural (or N/A)
STATE	95.6	4*	construct (head of *idafa*), definite, or indefinite (or N/A)
CASE	86.3	5*	nominative, accusative or genitive (or N/A)
NUMDGT	99.5	7*	a NUMBER feature with digit token representation; see Section A.3
NUMDGTBIN	99.5	12*	a NUMBER feature with number 'binning' according to syntactic agreement patterns; see Section A.3
FNNUM	99.2	6*	a functional NUMBER feature, using ElixirFM; see Section 5.1
FNNUMDGT	99.2	7*	a functional NUMBER feature with digit token representation, using ElixirFM; see Sections 5.1 and A.3
FNNUMDGTBIN	99.2	12*	a functional NUMBER feature with number 'binning' according to syntactic agreement patterns, using ElixirFM; see Sections 5.1 and A.3
FN*GENDER	98.6	6*	a functional GENDER feature, using our in-house resource; see Section 5.2
FN*NUM	99.0	7*	a functional NUMBER feature, using our in-house resource; see Section 5.2
FN*NUMDGTBIN	99.0	13*	a functional NUMBER feature with number 'binning' according to syntactic agreement patterns, using our in-house resource; see Sections 5.2 and A.3
RAT	95.6	5*	rationality (humanness) feature; rational, irrational, ambiguous, unknown or N/A; using our in-house resource; see Section 5.2
PNG	–	–	abbrev. for PERSON, NUMBER, and GENDER (a.k.a. ϕ-features); similarly for PG
FN*NGR	–	–	abbrev. for functional NUMBER, GENDER, and RAT; similarly for FN*NG

top contributors, that is, highly relevant, in optimal (gold) conditions (see previous paragraph). The determiner feature (DET), followed by the STATE feature, were top individual contributors in set-up *Sep*. Adding the features that participate in agreement, namely, DET and the PNG features (PERSON, NUMBER, GENDER), in the *Greedy* set-up, yielded a 1.4% gain over the CORE12 baseline. These results suggest that for a successful feature combination, one should take into account not only the relevance of the features, but also their accuracy.

Table 4
CORE12 POS tag set with morphological inflectional features. Left half: Using gold POS tag and feature values. In it: Top part (*All*): Adding all nine inflectional features to CORE12. Second part (*Sep*): Adding each feature separately to CORE12. Third part (*Greedy*): Greedily adding next best feature from *Sep*, and keeping it if improving score. Right half: Same as left half, but with predicted POS tag and feature values. Statistical significance tested only on predicted (non-gold) input, against the CORE12 baseline.

	gold POS and feature values				predicted POS and feature values			
Set-up	**CORE12+...**	LAS	UAS	LS	**CORE12+...**	LAS	UAS	LS
All	(baseline repeated)	82.9	85.4	93.5	(baseline repeated)	**78.7**	82.5	90.6
	+ all 9 infl. features	**85.2**	86.6	95.3	+ all 9 infl. features	77.9	82.1	90.0
Sep	+CASE	**84.6**	86.3	95.0	+DET	**79.8**$^{++}$	83.2	91.5
	+STATE	84.2	86.4	94.4	+STATE	79.4^{++}	82.9	91.2
	+DET	84.0	86.2	94.2	+GENDER	78.8	82.4	90.8
	+NUMBER	83.1	85.5	93.6	+PERSON	78.7	82.5	90.7
	+PERSON	83.1	85.4	93.7	+NUMBER	78.7	82.4	90.6
	+VOICE	83.1	85.4	93.6	+VOICE	78.6	82.4	90.6
	+MOOD	83.1	85.5	93.5	+ASPECT	78.6	82.4	90.5
	+ASPECT	83.0	85.4	93.5	+MOOD	78.5	82.4	90.5
	+GENDER	83.0	85.2	93.6	+CASE	75.8	80.2	88.5
Greedy	+CASE+STATE	**85.4**	86.9	95.5	+DET+STATE	79.4^{++}	82.8	91.2
	+CASE+STATE+DET	85.2	86.7	95.4	+DET+GENDER	79.9^{++}	83.2	91.7
	+CASE+STATE+NUMBER	85.4	86.9	95.5	+DET+GENDER+PERSON	79.9^{++}	83.2	91.7
	+CASE+STATE+PERSON	85.3	86.8	95.4	+DET+PNG	**80.1**$^{++}$	83.3	91.8
	+CASE+STATE+VOICE	85.3	86.8	95.4	+DET+PNG+VOICE	80.0^{++}	83.2	91.7
	+CASE+STATE+MOOD	85.2	86.7	95.4	+DET+PNG+ASPECT	80.0^{++}	83.2	91.8
	+CASE+STATE+ASPECT	85.2	86.8	95.4	+DET+PNG+MOOD	80.0^{++}	83.2	91.8
	+CASE+STATE+GENDER	85.3	86.8	95.4	—			

4.4 Lexical Features and Their Contribution to Parsing Quality

Next, we experimented with adding the lexical features, which involve semantic abstraction to some degree: the diacritized LEMMA, the undiacritized lemma (LMM), the ROOT, and the PATTERN (which is the pattern of the LEMMA). A notable advantage of lexical abstraction is that it reduces data sparseness, and explicitly ties together semantically related words. We experimented with the same set-ups as above: *All, Sep*, and *Greedy*.

Gold feature values. The left-hand side half of Table 5 shows that adding all four features yielded gains similar to adding a lemma feature separately. With gold tags, however, no proper subset of the lexical features beats the set of all lexical features.

Predicted feature values. The right-hand side of Table 5 shows that adding all four features yielded a minor gain in set-up *All*. LMM was the best single contributor, closely followed by ROOT in *Sep*. CORE12+LMM+ROOT (with or without LEMMA) was the best greedy combination in set-up *Greedy*, and also provides the best performance of all experiments with lexical features only. Due to the high redundancy of LEMMA and LMM (only 0.01% absolute gain when adding LEMMA in the *Greedy* set-up, which appears larger only due to rounding in the table), we do not consider LEMMA in feature combinations from this point on. Note, however, that LEMMA—and all the lexical features—are predicted with high accuracy (top half, second section of Table 3).

Table 5
Models with lexical morpho-semantic features. Top: Adding all lexical features together on top of the CORE12 baseline. Center: Adding each feature separately. Bottom: Greedily adding best features from previous part, on predicted input. Statistical significance tested only on predicted (non-gold) input, against the CORE12 baseline.

set-up	CORE12+...	gold			predicted		
		LAS	UAS	LS	LAS	UAS	LS
All	CORE12 (baseline repeated)	82.9	85.4	93.5	78.7	82.5	90.6
	+ all lexical features	**83.4**	85.5	93.9	**78.9**	82.5	90.8
Sep	+LMM (lemma without diacritics)	83.3	85.5	93.8	**79.0**$^{+}$	82.5	90.8
	+ROOT	83.2	85.5	93.7	78.9^{+}	82.6	90.7
	+LEMMA	**83.4**	85.5	93.8	78.8	82.4	90.7
	+PATTERN	83.1	85.5	93.6	78.6	82.4	90.6
Greedy	+LMM+ROOT	83.3	85.5	93.9	79.0^{++}	82.6	90.9
	+LMM+ROOT+LEMMA	83.3	85.4	93.8	**79.1**$^{++}$	82.6	90.9
	+LMM+ROOT+PATTERN	**83.4**	85.5	93.9	78.9	82.6	90.8

4.5 Inflectional and Lexical Feature Combination and Its Contribution to Parsing Quality

We now combine morphological and lexical features. Following the same greedy heuristic as in the previous sections, we augmented the best inflection-based model CORE12+DET+PNG with lexical features, and found that the undiacritized lemma (LMM) improved performance on predicted input (80.2%) (see Table 6). Adding more lexical features does not help, however, suggesting that some of the information in the lexical features is redundant with the information in the morphological features. See the Appendix, Section A.1, for our attempt to extend the tag set by embedding the best feature combination in it.

4.6 Additional Feature Engineering

So far we have experimented with morphological feature values as extracted from the PATB (gold) or predicted by MADA; we also used the same MaltParser feature configuration for all added features (i.e., stk[0] + buf[0]). It is likely, however, that from a machine-learning perspective, representing similar categories with the same tag, or

Table 6
Models with inflectional and lexical morphological features together (predicted value-guided heuristic). Statistical significance tested only on predicted input, against the CORE12 baseline.

tag set	gold			predicted		
	LAS	UAS	LS	LAS	UAS	LS
CORE12+DET+PNG (rep.)	84.2	86.2	94.5	**80.1**$^{++}$	83.3	91.8
CORE12+DET+PNG+LMM	**84.4**	**86.4**	**94.6**	**80.2**$^{++}$	**83.3**	**91.9**
CORE12+DET+PNG+LMM +ROOT	84.3	86.3	94.6	80.1^{++}	83.3	91.8
CORE12+DET+PNG+LMM +PATTERN	84.4	86.3	94.6	80.0^{++}	83.2	91.8

Table 7
Models with re-engineered DET and PERSON inflectional features. Statistical significance tested only on predicted input, against the CORE12 baseline.

model (POS tag set and infl. feature)	gold			predicted		
	LAS	UAS	LS	LAS	UAS	LS
CORE12+DET (repeated)	84.0	86.2	94.2	79.8^{++}	83.2	91.5
CORE12+DET2	**84.1**	86.4	94.3	$\mathbf{80.1}^{++}$	83.5	91.7
CORE12+DET+PNG+LMM (repeated)	84.4	86.4	94.6	80.2^{++}	83.3	91.9
CORE12+DET2+PNG+LMM	**84.6**	86.5	94.7	$\mathbf{80.2}^{++}$	83.4	91.9

taking into account further-away tokens in the sentence, may be useful for learning. Therefore, we next experimented with modifying some inflectional features that proved most useful in predicted input.

As DET may help disambiguate N-N / N-A constructions (and N-N-N, N-A-A, ..., see Section 2), we attempted modeling the DET values of previous and next elements (as MaltParser's stk[1] + buf[1], in addition to the modeled stk[0] + buf[0]). This variant, denoted DET2, indeed helps: When added to the CORE12 baseline model, DET2 improves non-gold parsing quality by more than 0.3%, compared to DET, as shown in Table 7. This variant yields a small improvement also when used in combination with the PNG and LMM features, as shown in the second part of Table 7—but only in gold. These results suggest an intricate interaction between the extended relevance of the determiner feature, and its redundancy with the PNG features (and note that all features involved are predicted with high accuracy). A possible explanation might be that form-based feature representation is inherently inadequate here, and therefore its high accuracy may not be very indicative. We explore non-form-based (functional) feature representation in Section 5. For more on our feature engineering, see the Appendix, Section A.2.

5. Parsing Experiments with Functional Features

Section 4 explored the contribution of various POS tag sets, (form-based) morphological features, and promising combinations thereof, to Arabic dependency parsing quality—in straightforward feature engineering design and combination heuristics. This section explores more sophisticated feature engineering: using functional NUMBER and GENDER feature values, instead of form-based values; using the non-form-based rationality (RAT) feature; and combinations thereof. For additional experiments regarding alternative representation for digit tokens, and the "binning" Arabic number constructions according to their complex syntactic patterns, see the Appendix, Section A.3.

5.1 Functional Feature Representation for Broken Plurals (using ElixirFM)

The NUMBER feature we have thus far extracted from PATB with MADA only reflects *form-based* (as opposed to *functional*) values, namely, broken plurals are marked as singular. This might have a negative effect for learning generalizations over the complex agreement patterns in MSA, beyond memorization of word pairs seen together in

training. To address this issue, one can use the Arabic morphological tool ElixirFM[16] (Smrž 2007). For each given word form, it outputs a list of possible analyses, each containing a lemma and a functional NUMBER (and other features). We replaced the surface NUMBER value for all nominals marked as singular in our data with ElixirFM's functional value, using the MADA-predicted lemma to disambiguate multiple ElixirFM analyses. These experiments are denoted with FNNUM. In training, of the lemma types sent to ElixirFM for analysis, about 20% received no analysis (OOV). A manual observation of a small sample revealed that at least half of those were proper names (and hence their NUMBER value would have stayed singular). Almost 9% of the ElixirFM-analyzed types (over 7% of the tokens) changed their NUMBER value. In the dev set, the OOV rate was less than 9%, and almost 11% of the ElixirFM-analyzed types changed their NUMBER value. This amounts to 4.4% of all tokens.

We used ElixirFM to determine the values for FNNUM, the functional number feature. We used this feature in our best model so far, CORE12+DET+PNG+LMM, instead of the form-based NUMBER feature.[17] The ElixirFM-based models yielded small gains of up to 0.1% over this best model on predicted input. We then modified the ElixirFM-based best model to use the enhanced DET2 feature. This variation yielded a similarly small gain, altogether less than 0.2% from its ElixirFM-free counterparts.

5.2 Functional Gender and Number Features, and the Rationality Feature

The ElixirFM lexical resource used previously provided functional NUMBER feature values but no functional GENDER values, nor RAT (rationality, or humanness) values. To address this issue, we use a version of the PATB3 training and dev sets manually annotated with functional gender, number, and rationality (Alkuhlani and Habash 2011).[18] This is the first resource providing all three features (ElixirFm only provides functional number, and to some extent functional gender). We conducted experiments with gold features to assess the potential of these features, and with predicted features, obtained from training a simple maximum likelihood estimation classifier on this resource (Alkuhlani and Habash 2012).[19] The first part of Table 8 shows that the RAT (rationality) feature is very relevant (in gold), but suffers from low accuracy (no gains in machine-predicted input). The next two parts show the advantages of functional gender and number (denoted with a FN* prefix) over their surface-based counterparts. The fourth part of the table shows the combination of these functional features with the other features that participated in the best combination so far (LMM, the extended DET2, and PERSON); without RAT, this combination is at least as useful as its form-based counterpart, in both gold and predicted input; adding RAT to this combination yields 0.4% (absolute) gain in gold, offering further support to the relevance of the rationality feature, but a slight decrease in predicted input, presumably due to insufficient accuracy again. The last part of the table revalidates the gains achieved with the best controlled feature combination, using CATIBEX—the best performing tag set with predicted input. Note, however, that the 1% (absolute) advantage of CATIBEX (without additional features) over the morphology-free CORE12 on machine-predicted input (Table 2) has

16 http://sourceforge.net/projects/elixir-fm.

17 We also applied the manipulations described in Section A.3 to FNNUM, giving us the variants FNNUMDGT and FNNUMDGTBIN, which we tested similarly.

18 In this article, we use a newer version of the corpus by Alkuhlani and Habash (2011) than the one we used in Marton, Habash, and Rambow (2011).

19 The paper by Alkuhlani and Habash (2012) presents additional, more sophisticated models that we do not use in this article.

Table 8
Models with functional features: GENDER, NUMBER, rationality (RAT). FN* = functional feature(s) based on Alkuhlani and Habash (2011); GN = GENDER+NUMBER; GNR = GENDER+NUMBER+RAT. Statistical significance tested only for CORE12+... models on predicted input, against the CORE12 baseline.

model (POS tag set and features)	gold			predicted		
	LAS	UAS	LS	LAS	UAS	LS
CORE12 (repeated)	82.9	85.4	93.5	**78.7**	82.5	90.6
+FN*RATIONAL	**83.7**	85.8	94.0	78.7	82.5	90.7
+GENDER (repeated)	83.0	85.2	93.6	78.8	82.4	90.8
+FN*GENDER	**83.3**	85.5	93.7	**78.9**$^{+}$	82.6	90.9
+NUMBER (repeated)	83.1	85.5	93.6	78.7	82.4	90.6
+FN*NUMBER	83.3	85.6	93.7	78.9^{+}	82.5	90.7
+DET2+LMM+PNG (repeated)	84.6	86.5	94.7	80.2^{++}	83.4	91.9
+DET2+LMM+PERSON +FN*NGR	**85.0**	86.7	94.9	80.3^{++}	83.7	91.6
+DET2+LMM+PERSON +FN*NG	84.6	86.5	94.7	**80.4**$^{++}$	83.5	91.9
CATIBEX+DET2+LMM+PERSON+FN*NGR	**84.1**	85.9	94.4	80.7	84.0	91.9
CATIBEX+DET2+LMM+PERSON+FN*NG	83.5	85.4	94.1	**80.7**	83.7	92.2

shrunk with these functional feature combinations to 0.3%. We take it as further support to the relevance of our functional morphology features, and their partial redundancy with the form-based morphological information embedded in the CATIBEX POS tags.

6. Evaluation of Results with Easy-First Parser

In this section, we validate the contribution of key tag sets and morphological features—and combinations thereof—using a different parser: the Easy-First Parser (Goldberg and Elhadad 2010). As in Section 4, all models are evaluated on both gold and non-gold (machine-predicted) feature values.

The Easy-First Parser is a shift-reduce parser (as is MaltParser). Unlike MaltParser, however, it does not attempt to attach arcs "eagerly" as early as possible (as in previous sections), or at the latest possible stage (an option we abandoned early on in preliminary experiments). Instead, the Easy-First Parser keeps a stack of partially built treelets, and attaches them to one another in order of confidence (from high confidence, "easy" attachment, to low, as estimated by the classifier). Labeling the relation arcs is done in a second pass, with a separate training step, after all attachments have been decided (the code for which was added after the publication of Goldberg and Elhadad (2010), which only included an unlabeled attachment version).

Setting machine-learning features for Easy-First Parser is not as simple and elegant as for MaltParser, but it gives the feature designer greater flexibility. For example, the POS tag can be dynamically split (or not) according to the token's word-form and/or the already-built attachment treelets, whereas in MaltParser, one can meld several features into a single complex feature only if applied unconditionally to all tokens. The Easy-First Parser's first version comes with the code for the features used in its first publication. These include POS tag splitting and feature melding for prepositional attachment chains (e.g., parent-preposition-child). For greater control of the contribution of the various POS tag and morphological features in the experiments, and for

a better "apples-to-apples" comparison with MaltParser (as used here), we disabled these features, and instead used features (and selected feature melding) that were as equivalent to MaltParser as possible.

Table 9 shows results with Easy-First Parser. Results with Easy-First Parser are consistently higher than the corresponding results with MaltParser, with similar trends for the various features' contribution: Functional GENDER and NUMBER features contribute more than their form-based counterparts, in both gold and predicted conditions; rationality (RAT) as a single feature on top of the POS tag set helps in gold (and with Easy-First Parser, also in predicted conditions)—but when used in combination with PERSON, LMM, functional GENDER, and NUMBER, it actually slightly lowers parsing scores in predicted conditions (but with Easy-First Parser, it helps in gold conditions); DET is the most useful single feature in predicted conditions (from those we tried here); and the best performing model in predicted conditions is the same as with MaltParser: CORE12+DET+LMM+PERSON+FN*NG.[20]

As before, we see that the patterns of gain achieved with the "morphology-free" CORE12 hold also for CATIBEX, the best performing tag set on predicted input. Interestingly, with this parser, the greater 1.6% (absolute) advantage of CATIBEX (without additional features) over the morphology-free CORE12 on machine-predicted input (compare with only 1% in MaltParser in Table 2) has shrunk *completely* with these functional feature combinations. This suggests that Easy-First Parser is more resilient to accuracy errors (presumably due to its design to make less ambiguous decisions earlier), and hence can take better advantage of the relevant information encoded in our functional morphology features.

7. Combined Gold and Predicted Features for Training

So far, we have only evaluated models trained on gold POS tag set and morphological feature values. Some researchers, however, including Goldberg and Elhadad (2010), train on *predicted* feature values instead. It makes sense that training on predicted features yields better scores for evaluation on predicted features, since the training better resembles the test. But we argue that it also makes sense that training on a *combination* of gold and predicted features (one copy of each) might do even better, because good predictions of feature values are reinforced (since they repeat the gold patterns), whereas noisy predicted feature values are still represented in training (in patterns that do not repeat the gold).[21] To test our hypothesis, we start this section by comparing three variations:

- Training on gold feature values (as has been the case so far)
- Training on predicted feature values (as in Goldberg and Elhadad 2010)
- Training on the novel combination of gold and predicted features (denoted below as *g+p*)

20 Recall that DET2 was only defined for MaltParser, and not for the Easy-First Parser.

21 Although conceived independently, this hypothesis resembles *self-training* (McClosky, Charniak, and Johnson 2006), where the parser is re-trained on its own predicted parsing output, together with the original labeled training data. Note, however, that we re-train on gold and predicted feature values (e.g., POS tag, GENDER, or NUMBER), but we always use gold training data for HEAD and DEPREL. In both cases the parsers seem to benefit from training data (features) that better resemble the test data, while retaining bias toward the gold and correctly predicted data.

Table 9
Select models trained using the Easy-First Parser. Statistical significance tested only for CORE12... models on predicted input: significance of the Easy-First Parser CORE12 baseline model against its MaltParser counterpart; and significance of all other CORE12+... models against the Easy-First Parser CORE12 baseline model.

model (POS tag set and features)	gold			predicted		
	LAS	UAS	LS	LAS	UAS	LS
CORE12 (MaltParser baseline, repeated)	82.9	85.4	93.5	78.7	82.5	90.6
CORE12 (Easy-First Parser)	**83.5**	86.0	93.9	**79.6**$^{++}$	83.5	91.3
CORE12+NUMBER	83.3	85.7	94.0	79.5	83.4	91.3
CORE12+FN*NUMBER	**83.5**	85.9	94.0	**79.8**	83.6	91.4
CORE12+GENDER	83.5	86.0	94.0	79.5	83.5	91.3
CORE12+FN*GENDER	**83.6**	86.1	94.0	**79.7**	83.6	91.3
CORE12+RAT	84.2	86.4	94.4	79.6	83.6	91.3
CORE12+DET	**84.3**	86.7	94.5	**80.6**$^{++}$	84.1	92.2
CORE12+LMM	83.6	85.8	94.1	79.7	83.5	91.5
CORE12+DET+LMM+PNG	84.8	86.9	94.9	81.1^{++}	84.4	92.3
CORE12+DET+LMM+PERSON+FN*NG	84.9	86.9	94.8	**81.4**$^{++}$	84.7	92.4
CORE12+DET+LMM+PERSON+FN*NGR	**85.1**	87.1	94.9	81.2^{++}	84.7	92.1
CATIBEX	83.1	85.6	94.0	81.2	84.6	92.5
CATIBEX+DET+LMM+PERSON+FN*NG	83.5	85.8	94.2	**81.4**	84.6	92.7
CATIBEX+DET+LMM+PERSON+FN*NGR	**83.9**	85.9	94.7	81.1	84.6	92.5

The first two parts of Table 10 show that, as expected, training on gold feature values yields better scores when evaluated on gold, too (although later we see this is not always the case). More interestingly, when evaluated on predicted feature values, training on predicted feature values yields better parsing scores than when training on gold, and training on g+p yields best scores, in support of our hypothesis. Therefore, in the rest of the table (and in the rest of the experiments), we apply the g+p training variant to the best models so far, both in MaltParser and Easy-First Parser. The next part in Table 10 shows that this trend is consistent also with the best feature combinations so far. Interestingly, the RAT feature contributes to improvement only in the g+p condition, presumably because of its low prediction accuracy.

In Table 11, we repeated most of these experiments with other tag sets: CATIBEX and BW (best performers on predicted and gold input, respectively). We can see in this table that the same trends hold for these POS tag sets as well. Interestingly, the "morphology-free" CORE12 (in Table 10) outperforms CATIBEX here (Table 11), making CORE12+DET2+LMM+PERSON+FN*NGR our best MaltParser model on predicted feature values. Similarly, the Easy-First Parser model CORE12+DET+LMM+PERSON+FN*NG outperforms its CATIBEX counterpart (CATIBEX+DET+LMM+PERSON+FN*NG), resulting in our best model on the dev set in machine-predicted condition (82.7%).[22]

The richest POS tag set, BW, which is also the worst predicted tag set and worst performer on predicted input, had the most dramatic gains from using g+p: more than

22 See Section 9 for download information.

Table 10
Alternatives to training on gold-only feature values. Top: Select MaltParser CORE12+... models re-trained on predicted or gold + predicted feature values. Bottom: Similar models to the top half, with the Easy-First Parser. Statistical significance tested only for CORE12+... models on predicted input: significance of the MaltParser models from the MaltParser CORE12 baseline model, and significance of the Easy-First Parser models from the Easy-First Parser CORE12 baseline.

model (POS tag set and features)	gold			predicted		
	LAS	UAS	LS	LAS	UAS	LS
MaltParser:						
CORE12 (gold train, repeated)	**82.9**	85.4	93.5	78.7	82.5	90.6
CORE12 predicted train	82.4	85.0	93.2	79.8^{++}	83.2	91.4
CORE12 g+p	82.7	85.2	93.5	**80.0**$^{++}$	83.4	91.6
CORE12+DET+LMM+PNG (gold train, repeated)	**84.4**	86.4	94.6	80.2^{++}	83.3	91.9
CORE12+DET+LMM+PNG predicted train	84.1	86.1	94.3	81.6^{++}	84.4	92.8
CORE12+DET+LMM+PNG g+p	84.2	86.1	94.5	**81.7**$^{++}$	84.5	92.9
CORE12+DET2+LMM+PERSON+FN*NGR (gold train, repeated)	**85.0**	86.7	94.9	80.3^{++}	83.7	91.6
CORE12+DET2+LMM+PERSON+FN*NG (gold train, repeated)	84.6	86.5	94.7	80.4^{++}	83.5	91.9
CORE12+DET2+LMM+PERSON+FN*NG g+p	84.4	86.3	94.6	81.8^{++}	84.6	93.0
CORE12+DET2+LMM+PERSON+FN*NGR g+p	84.7	86.5	94.7	**81.9**$^{++}$	84.7	93.0
Easy-First Parser:						
CORE12 (gold train, repeated)	83.5	86.0	93.9	79.6	83.5	91.3
CORE12 g+p	**83.6**	86.1	94.1	**80.8**$^{++}$	84.4	92.3
CORE12+DET+LMM+PNG g+p	84.8	86.9	94.7	82.5^{++}	85.5	93.3
CORE12+DET+LMM+PERSON+FN*NG g+p	84.9	86.9	94.9	**82.7**$^{++}$	85.7	93.3
CORE12+DET+LMM+PERSON+FN*NGR g+p	**85.2**	87.2	95.0	82.6^{++}	85.7	93.2

5% (absolute) for LAS on predicted input with MaltParser (and over 3% with Easy-First Parser). Although much improved, BW models' performance still lags behind the leading models.

The results in Tables 10 and 11 suggest that our g+p training method is superior to the alternatives (independently of parser choice) due to making the parser more resilient to lower accuracy in the input. It also suggests that g+p training enables the parser to better exploit relevant data when represented in "cleaner" separate features, as opposed to when the POS tags are split into ambiguous form-based cases as in CATIBEX. Future experimentation is needed in order to test this latter conjecture.

8. Result Validation and Discussion

8.1 Validating Results on an Unseen Test Set

Once experiments on the development set were done, we ran the best performing form-based non-gold-based models from Section 4 on a previously unseen test set. This set

Table 11
Alternatives to training on gold-only feature values for CATIBEX and BW tag sets. Top: Select MaltParser models re-trained on predicted or gold + predicted feature values. Bottom: Similar models to the top half, with the Easy-First Parser. (Statistical significance was tested only for CORE12+... models – none here).

model (POS tag set and features)	gold			predicted		
	LAS	UAS	LS	LAS	UAS	LS
MaltParser:						
CATIBEX (gold train, repeated)	82.5	85.0	93.4	79.7	83.3	91.4
CATIBEX g+p	82.3	84.8	93.3	80.4	83.6	92.0
CATIBEX+DET2+LMM+PERSON+FN*NG (gold train, repeated)	83.5	85.4	94.1	80.7	83.7	92.2
CATIBEX+DET2+LMM+PERSON+FN*NGR (gold train)	**84.1**	85.9	94.4	80.7	84.0	91.9
CATIBEX+DET2+LMM+PERSON+FN*NG g+p	83.2	85.3	93.9	81.3	84.2	92.6
CATIBEX+DET2+LMM+PERSON+FN*NGR g+p	**83.7**	85.7	94.2	**81.4**	84.3	92.6
BW (gold train, repeated)	84.0	85.8	94.8	72.6	77.9	86.5
BW g+p	83.9	85.7	94.7	77.8	81.4	90.3
BW+DET2+LMM+PERSON+FN*NG g+p	84.8	86.4	95.1	79.4	82.6	91.2
BW+DET2+LMM+PERSON+FN*NGR g+p	**85.1**	86.6	95.2	**79.5**	82.7	91.2
Easy-First Parser:						
CATIBEX (gold train, repeated)	**83.1**	85.6	94.0	81.2	84.6	92.5
CATIBEX g+p	82.5	85.1	93.8	**81.2**	84.4	92.9
CATIBEX+DET+LMM+PERSON+FN*NG (gold train, repeated)	83.5	85.8	94.2	81.4	84.6	92.7
CATIBEX+DET+LMM+PERSON+FN*NGR (gold train, repeated)	83.9	85.9	94.7	81.1	84.6	92.5
CATIBEX+DET+LMM+PNG g+p	83.4	85.7	94.3	82.1	85.0	93.3
CATIBEX+DET+LMM+PERSON+FN*NG g+p	83.6	85.8	94.4	82.0	84.9	93.4
CATIBEX+DET+LMM+PERSON+FN*NGR g+p	**83.9**	86.0	94.6	**82.2**	85.3	93.2
BW (gold train, repeated)	84.9	86.6	95.6	77.5	82.2	90.1
BW g+p	84.4	86.2	95.3	80.7	84.1	92.5
BW+DET+LMM+PERSON+FN*NG g+p	84.8	86.5	95.5	81.1	84.2	92.8
BW+DET+LMM+PERSON+FN*NGR g+p	**85.1**	86.7	95.6	**81.2**	84.4	92.9

Table 12
Results on PATB3-TEST for form-based models which performed best on PATB3-DEV – predicted input. Statistical significance tested on the PATB3-TEST set, only for MaltParser CORE12+... models against the MaltParser CORE12 baseline model output.

model (POS tag set and morph. features)	LAS	UAS	LS
CORE12	77.3	81.0	90.1
CORE12+DET+PNG	78.6	81.7	91.1
CORE12+DET+LMM+PNG	**79.1**$^{++}$	**82.1**	**91.4**
CATIBEX	78.5	81.8	91.0
CATIBEX+DET+LMM+PNG	**79.3**	**82.4**	**91.6**

is the test split of part 3 of the PATB (hereafter PATB3-TEST; see Table 1 for details). Table 12 shows that the same trends held on this set too, with even greater relative gains, up to almost 2% absolute gains.

We then also revalidated the contribution of the best performing models from Sections 5–7 on PATB3-TEST. Here, too, the same trends held. Results are shown in Table 13.

8.2 Best Results on Length-Filtered Input

For better comparison with work of others, we adopt the suggestion made by Green and Manning (2010) to evaluate the parsing quality on sentences up to 70 tokens long. We report these filtered results in Table 14. Filtered results are consistently higher (as expected). Results are about 0.9% absolute higher on the development set, and about 0.6% higher on the test set. The contribution of the RAT feature across sets is negligible (or small and unstable), resulting in less than 0.1% absolute loss on the dev set, but about 0.15% gain on the test set. For clarity and conciseness, we only show the best model (with RAT) in Table 14.

8.3 Error Analysis

We perform two types of error analyses. First, we analyze the attachment accuracy by attachment relation type on PATB3-DEV. Our hypothesis is that the syntactic relations which are involved in agreement or assignment configurations will show an improvement when the relevant morphological features are used, but other syntactic

Table 13
Results on PATB3-TEST for models that performed best on PATB3-DEV – predicted input. Using MaltParser, unless indicated otherwise. g+p = trained on combination of gold and predicted input (instead of gold-only). Statistical significance tested only for CORE12+... models: For MaltParser CORE12+... models against the MaltParser CORE12 baseline model output, and for Easy-First Parser CORE12+... models against the Easy-First Parser CORE12 baseline model output.

POS tag set	LAS	UAS	LS
CORE12 (repeated)	77.3	81.0	90.1
CORE12+DET2+LMM+PG+FNNUMDGTBIN	79.3^{++}	82.3	91.4
CORE12+DET2+LMM+PERSON +FN*NGR	78.9^{+}	82.3	91.0
CORE12+DET2+LMM+PERSON +FN*NG	79.1^{++}	82.1	91.4
CORE12+DET2+LMM+PERSON +FN*NGR g+p, Easy-First Parser	$\mathbf{81.0}^{++}$	**84.0**	92.7
CORE12+DET2+LMM+PERSON +FN*NG g+p, Easy-First Parser	80.9^{++}	83.9	**92.8**
CATIBEX	78.5	81.8	91.0
CATIBEX+DET2+LMM+PG+FNNUMDGTBIN	79.4	82.5	91.6
CATIBEX+DET2+LMM+PERSON +FN*NGR	79.3	82.6	91.3
CATIBEX+DET2+LMM+PERSON +FN*NG	79.3	82.4	91.5
CATIBEX+DET2+LMM+PERSON +FN*NGR g+p, Easy-First Parser	79.5	**83.0**	91.9
CATIBEX+DET2+LMM+PERSON +FN*NG g+p, Easy-First Parser	**79.6**	82.8	**92.1**
BW	72.1	77.2	86.3
BW+DET2+LMM+PERSON +FN*NGR g+p, Easy-First Parser	79.6	82.7	92.2
BW+DET2+LMM+PERSON +FN*NG g+p, Easy-First Parser	**79.7**	**82.9**	**92.3**

Table 14
Results for best performing model on PATB3-DEV and PATB3-TEST for sentences up to 70 tokens long (predicted input).

model	evaluated on	LAS	UAS	LS
CORE12+DET+LMM+PERSON+FN*NGR g+p, Easy-First Parser	PATB3-DEV	83.6	86.5	93.5
CORE12+DET+LMM+PERSON+FN*NGR g+p, Easy-First Parser	PATB3-TEST	81.7	84.6	92.8

relations will not. Second, we analyze the grammaticality of the obtained parse trees with respect to agreement and assignment phenomena. Here, our hypothesis is that when using morphological features, the grammaticality of the obtained parse trees will increase.

Attachment accuracy by relation type. Our first hypothesis is illustrated in Figure 2. On the left, we see the parse provided by our baseline system (MaltParser using only CORE12), which has two errors:

- The node labeled أيام *ÂyAm* ('days') should be the subject, not the object, of the main verb مرت *mrt* ('passed'). Morphologically, أيام *ÂyAm* is masculine plural, and مرت *mrt* ('passed') is feminine singular, obeying the agreement pattern under which a non-rational subject following the verb always triggers a feminine singular verbal form.
- The node labeled المهندس *Almhnds* ('the engineer') should not be in an *idafa* (genitive construction) dependency with its governor الزميل *Alzmyl* ('the colleague'), but in a modifier relation (a sort of apposition, in this case). This must be the case because a noun that is the head of an *idafa* construction cannot have a definite determiner, as is the case here.

Both errors could be corrected (to the correct form as in our best model, on the right-hand side of Figure 2) if functional morphological features were available to the parser, including the rationality feature, and if the parser could learn the agreement rule for non-rational subjects, as well as the requirement that the head of an *idafa* construction cannot have a definite article.

Our first hypothesis is generally borne out. We discuss three conditions in more detail:

1. Using morphological features with the MaltParser and training on gold tags (Table 15).
2. Using morphological features with the MaltParser and training on a combination of gold and predicted tags (Table 16).
3. Using morphological features with the Easy-First parser and training on a combination of gold and predicted tags (Table 17).

In all cases, for controlled investigation, we compare the error reduction resulting from adding morphological features to a "morphology-free" baseline, which in all cases we take to be the MaltParser trained on the gold CORE12, and evaluated on

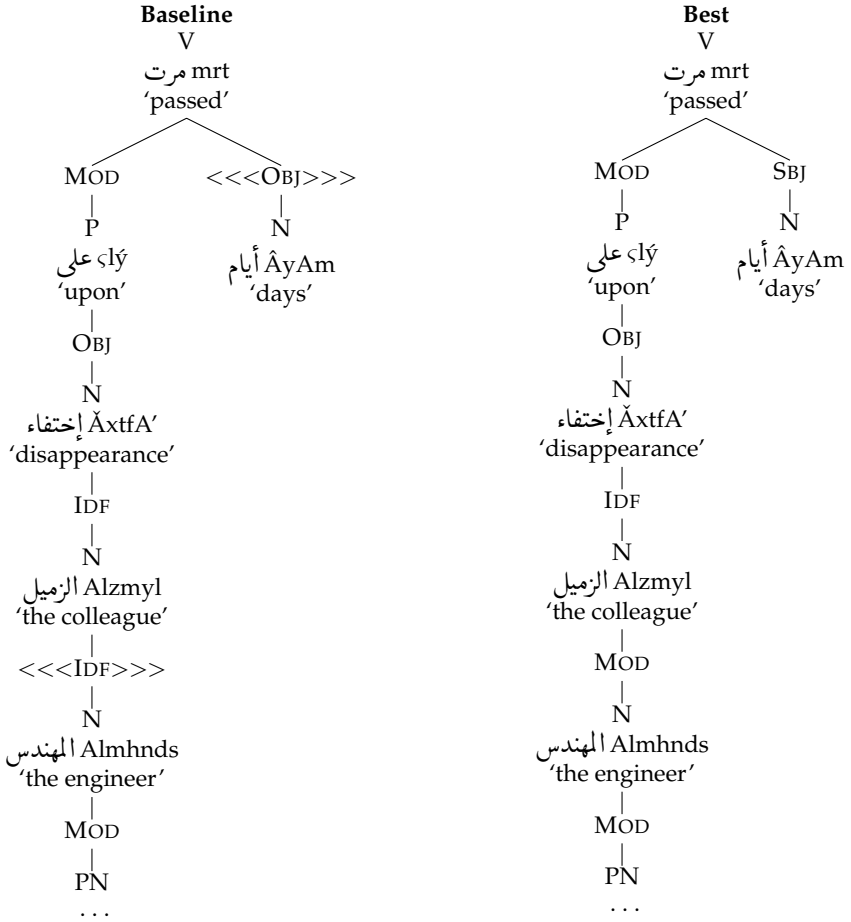

Figure 2
Error analysis example. . . . مرت أيام على إختفاء الزميل المهندس *mrt ÂyAm ʕlý ĂxtfA' Alzmyl Almhnds* . . . ('Several days have passed since the disappearance of the colleague the engineer . . . '), as parsed by the baseline system using only CORE12 (left) and as using the best performing model (right). Bad predictions are marked with <<< . . . >>>. The words in the tree are presented in the Arabic reading direction (from right to left).

machine-predicted input (except for Table 17, where the Easy-First Parser is trained and evaluated instead).

We start out by investigating the behavior of MaltParser, using all gold tags for training. The accuracy by relation type is shown in Table 15. Using just CORE12, we see that some attachments (subject, modifications) are harder than others (objects, *idafa*). We see that by adding LMM, all attachment types improve a little bit; this is as expected, because 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, and *idafa* reduce error by substantial margins (error reduction over CORE12 is greater than 10%; in the case of *idafa* it is 21.8%), and all other relations (including object and prepositional attachment) improve to a lesser degree (error reduction of 7.1% or less). We assume that the non-agreement relations (object

Table 15
Training the MaltParser on gold tags, accuracy by gold attachment type (selected): subject, object, modification (of a verb or a noun) by a noun, modification (of a verb or a noun) by a preposition, *idafa*, and overall results (repeated).

model (POS and morphological features)	SBJ	OBJ	MOD-N	MOD-Prep	IDF	total
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+PERSON +FN*NG	72.3	91.0	75.6	72.7	95.5	80.4
CORE12 + DET2+LMM+PERSON +FN*NGR	71.9	91.2	74.5	73.2	95.3	80.2

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, using functional number and gender, we see a further reduction in the agreement-related attachments, namely, subject and nominal modification (error reductions over baseline of 13.7% and 12.9%, respectively). *Idafa* decreases slightly (because this relation is not affected by the functional features), whereas object stays the same. Surprisingly, prepositional attachment also improves, with an error reduction of 8.1%. 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.

We then add the rationality feature (last line of Table 15). We now see that all relations affected by agreement or assignment perform worse than without the rationality feature. In contrast, all other relations improve. The decrease in performance can be explained by the fact that the rationality (RAT) feature is not predicted with high accuracy; because it interacts directly with agreement, and because we are training on gold annotation, the models trained do not correspond to the seen data. We expect rationality to contribute when we look at training that includes predicted features. (We have no explanation for the improvement in the other relations.)

We now turn to training the MaltParser on a combination of gold and predicted POS and morphological feature values (g+p; Section 7). The accuracy by relation is shown in Table 16. The table repeats (in the first row) the results for the MaltParser trained only using gold CORE12 features. First, we see that using the same single feature, but training on gold and predicted tags, we obtain an across-the-board improvement,

Table 16
Training the MaltParser on gold and predicted tags, accuracy by gold attachment type (selected): subject, object, modification (of a verb or a noun) by a noun, modification (of a verb or a noun) by a preposition, *idafa*, and overall results (repeated).

model (POS and morphological features)	SBJ	OBJ	MOD-N	MOD-Prep	IDF	total
CORE12	67.9	90.4	72.0	70.3	94.5	78.7
CORE12 g+p	70.9	91.0	73.5	70.2	94.7	80.0
CORE12 + DET+LMM+PNG g+p	73.7	91.6	76.6	72.8	96.3	81.7
CORE12 + DET2+LMM+PERSON +FN*NG g+p	74.3	91.8	77.4	72.9	95.2	81.8
CORE12 + DET2+LMM+PERSON +FN*NGR g+p	74.8	91.6	77.4	73.5	95.5	81.9

Table 17
Training the Easy-First Parser on gold and predicted tags, accuracy by gold attachment type (selected): subject, object, modification (of a verb or a noun) by a noun, modification (of a verb or a noun) by a preposition, *idafa*, and overall results (repeated).

model (POS and morphological features)	SBJ	OBJ	MOD-N	MOD-Prep	IDF	total
CORE12	70.8	90.7	73.1	71.4	94.2	79.6
CORE12 g+p	73.3	91.2	74.6	71.4	95.0	80.8
CORE12 + DET+LMM+PERSON+FN*NG g+p	76.4	91.9	77.9	73.2	96.2	82.7
CORE12 + DET+LMM+PERSON+FN*NGR g+p	76.2	91.9	78.1	73.2	95.9	82.6

with error reductions between 3.6% and 9.3%, with no apparent patterns. (Prepositional modifications even show a slight decline in attachment accuracy). This row (using only CORE12 and training on gold and predicted) now becomes our baseline for subsequent discussion of error reduction. If we then add the form-based features, we again find that the error rate decrease for subject, nominal modification, and *idafa* (the relations affected by agreement and assignment) is greater than that for the other relations; with this training corpus, however, the separation is not as stark, with subject decreasing its error rate by 9.6% and prepositional modification by 8.7%. Notably, *idafa* shows the greatest error rate reduction we have seen so far: 30.2%. When we turn to functional features, we again see a further increase in performance across the board. And, as expected, the penalty for using the rationality feature disappears because we have trained on predicted features as well. In fact the improvement due to rationality specifically benefits the relations affected by agreement and assignment, with subject reducing error by 13.4% now, nominal modification by 14.7%, and *idafa* by 34.0%. The tree on the right in Figure 2 is the parse tree returned by this model, and both the subject and the *idafa* relation are correctly analyzed. Note that the increase in the accuracy of *idafa* is probably not related to the interaction of syntax and morphology in assignment, because assignment in the *idafa* construction is not affected by rationality. Instead, we suspect that the parser can exploit the very different profile of the rationality feature in the dependent node of the *idafa* and modification constructions. Looking just at nominals, we see in the gold corpus that 62% of the dependents in a modification relation have no inherent rationality (this is the case notably for adjectives), whereas this number for *idafa* is only 18%. In contrast, the dependent of an *idafa* is irrational 66% of the time, whereas for modification that number is only 16%.

Finally, we turn to the use of the Easy-First Parser (Section 6). The accuracy by relation is shown in Table 17. When we switch from MaltParser to Easy-First Parser, we get an overall error reduction of 4.2%, which is reflected fairly evenly among the relations, with two outliers: subjects improve by 9.0%, whereas *idafa* increases its error rate by 5.5%! We do not have an immediate analysis for this behavior, because *idafa* is usually considered an "easy" relation (no word can intervene between the linked words), as reflected in the high accuracy numbers for this relation. Furthermore, when we inspect the unlabeled accuracy scores (not shown here), we see that the unlabeled attachment score for *idafa* also decreases. Thus, we must reject a plausible hypothesis, namely, that the parser gets the relations right but the labeler (which in the Easy-First Parser is a separate, second-pass module) gets the labels wrong. When we train the Easy-First Parser on gold and predicted, we see a similar improvement pattern over just training on gold as we did with the MaltParser; one exception is that the *idafa* relation improves greatly again. Finally, we add the functional morphological features (training

on gold and predicted). Again, the pattern we observe (by comparing error reduction against using Easy-First Parser trained only using CORE12 on gold and predicted) are very similar to the pattern we observed with the MaltParser in the same conditions. One difference stands out, however: whereas the MaltParser can exploit the rationality feature when trained on gold and predicted, the Easy-First Parser cannot. Object and prepositional modification perform identically with or without rationality, but subject and *idafa* perform worse; only nominal modification performs better (with overall performance decreasing). If we inspect the unlabeled attachment scores for subjects, we do detect an increase in accuracy (from 85.0% to 85.4%); perhaps the parser can exploit the rationality feature, but the labeler cannot.

Grammaticality of parse trees. We now turn to our second type of error analysis, the evaluation of the grammaticality of the parse trees in terms of gender and number agreement patterns. We use the agreement checker code developed by Alkuhlani and Habash (2011) and evaluate our baseline (MaltParser using only CORE12), best performing model (Easy-First Parser using CORE12 + DET+LMM+PERSON+FN*NGR g+p), and the gold reference. The agreement checker verifies, for all verb–nominal subject relations and noun–adjective relations found in the tree, whether the agreement conditions are met or not. The accuracy number reflects the percentage of such relations found which meet the agreement criteria. Note that we use the syntax given by the tree, not the gold syntax. For all three trees, however, we used gold morphological features for this evaluation even when those features were not used in the parsing task. This is because we want to see to what extent the predicted morphological features help find the correct syntactic relations, not whether the predicted trees are intrinsically coherent given possibly false predicted morphology. The results can be found in Table 18. We note that the grammaticality of the gold corpus is not 100%; this is approximately equally due to errors in the checking script and to annotation errors in the gold standard. We take the given grammaticality of the gold corpus as a topline for this analysis. Nominal modification has a smaller error band between baseline and gold compared with subject–verb agreement. We assume this is because subject–verb agreement is more complex (it depends on their relative order), and because nominal modification can have multiple structural targets, only one of which is correct, although all, however, are plausible from the point of view of agreement. The error reduction relative to the gold topline is 62% and 76% for nominal agreement and verb agreement, respectively. Thus, we see that our second hypothesis—that the use of morphological features will reduce grammaticality errors in the resulting parse trees with respect to agreement phenomena—is borne out.

In summary, we see that not only do morphological (and functional morphological features in particular) improve parsing, but they improve parsing in the way

Table 18
Analysis of grammaticality of agreement relations between verb and subject and between a noun and a nominal modifier (correct agreement in percent).

model (POS and morphological features)	noun-modifier	subject-verb
Gold	97.8	98.1
MaltParser using CORE12	95.2	88.6
Easy-First Parser using CORE12 + DET+LMM+PERSON+FN*NGR g+p	96.8	95.8

that we expect: (a) those relations affected by agreement and assignment contribute more than those that are not, and (b) agreement errors in the resulting parse trees are reduced.

9. Conclusions and Future Work

We explored the contribution of different morphological features (both inflectional and lexical) to dependency parsing of Arabic. Starting with form-based morphological features, we find that definiteness (DET), PERSON, NUMBER, GENDER, and undiacritized lemma (LMM) are most helpful for Arabic dependency parsing on predicted (non-gold) input. We further find that functional gender, number, and rationality features (FN*GENDER, FN*NUMBER, RAT) improve over form-based-only morphological features, as expected when considering the complex agreement rules of Arabic. To our knowledge, this is the first result in Arabic NLP using functional morphological features, and showing an improvement over form-based features.

This article presented a large number of results. We summarize them next.

1. We observe a tradeoff among the three factors (relevance, redundancy, and accuracy) of morphological features in parsing quality. The best performing tag set (BW) under the gold condition (i.e., it is very relevant) is worst under the machine-predicted condition, because of its dismal prediction accuracy rate. The tag set with highest prediction accuracy (CATIB6) does not necessarily yield the best results in dependency parsing accuracy, because it is not very relevant. A simple extension of CATIB6, however, that improves its relevance (CATIBEX) but retains sufficient accuracy improves the overall parsing quality.

2. Lexical features do help parsing, and the most helpful in predicted condition is the undiacritized lemma (LMM) feature. Although LMM is more ambiguous than the diacritized LEMMA feature, it has half the error rate of LEMMA which makes it a more reliable (accurate) feature. When using LMM, LEMMA is highly redundant (and vice versa).

3. GENDER and NUMBER and their functional variants are the most useful for parsing in predicted condition. This is a result of their high relevance and their high prediction accuracy. In contrast, CASE and STATE are the best performers in the gold condition (i.e., highly relevant) but not in the predicted condition (where CASE is actually the worst feature). The rationality (RAT) feature is more helpful in the gold condition, which suggests it is relevant; its associated parsing results in predicted condition are not as good, however. Presumably, this is because of its lower prediction accuracy.

4. When evaluating in the machine-predicted input condition, training on data with gold *and* predicted morphological features (g+p training) consistently improves results over training on gold. This novel technique most likely addresses the negative effect of feature prediction error by introducing the common errors to the parsing model in training. A side effect of it is that using correct predictions by the parser is reinforced, because constructions with correctly predicted values appear twice as often in g+p training.

5. All of these results carry over successfully to another parser (Easy-First Parser), suggesting the insights are not specific to MaltParser.

6. Our best model was trained with the Easy-First Parser, containing the following features: CORE12+DET+LMM+PERSON+FN*NGR, with g+p feature values for training. We make this model available, together with the source code.[23]

Although we only experimented with Arabic dependency parsing, we believe that the evaluation framework we presented and many of our conclusions will carry over to other languages (particularly, Semitic and morphology-rich languages) and syntactic representations (e.g., phrase structure). Some of our conclusions are more language independent (e.g., those involving the use of predicted training conditions).

In future work, we intend to improve the prediction of functional morphological features—especially RAT—in order to improve dependency parsing accuracy in predicted condition. We also intend to investigate how these features can be integrated into other parsing frameworks; we expect them to help independently of the framework. The ability to represent the relevant morphological information in a manner that is useful to attachment decisions is, of course, crucial to improving parsing quality.

Acknowledgments

This work was supported by the DARPA GALE program, contract HR0011-08-C-0110. Y. Marton performed most of the work on this paper while he was at the Center for Computational Learning Systems at Columbia University and at the IBM Watson Research Center. We thank Joakim Nivre for his useful remarks, Ryan Roth for his help with MADA, and Sarah Alkuhlani for her help with functional features. We also thank three anonymous reviewers for thoughtful comments.

References

Alkuhlani, Sarah 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)*, pages 357–362, Portland, OR.

Alkuhlani, Sarah and Nizar Habash. 2012. Identifying broken plurals, irregular gender, and rationality in Arabic text. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 675–685, Avignon.

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

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

Collins, Michael, 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)*, pages 505–512, College Park, MD.

Cowan, Brooke 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, Morristown, NJ.

Dada, Ali. 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.

Diab, Mona. 2007. Towards an optimal POS tag set for modern standard Arabic processing. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pages 91–96, Borovets.

Diab, Mona and Yassine Benajiba. (in preparation). From raw text to base phrase chunks: The new generation of AMIRA Tools for the processing of Modern Standard Arabic.

Diab, Mona, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of Arabic text: From raw text to base phrase

23 Available for downloading at `http://www1.ccls.columbia.edu/ CATiB/parser`.

chunks. In *Proceedings of the 4th Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL) - Human Language Technology (HLT)*, pages 149–152, Boston, MA.

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

Goldberg, Yoav and Michael Elhadad. 2010. An efficient algorithm for easy-first non-directional dependency parsing. In *Proceedings of Human Language Technology (HLT): The North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 742–750, Los Angeles, CA.

Green, Spence 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.

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

Habash, Nizar, 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*, pages 125–135, Cairo.

Habash, Nizar, Ryan Gabbard, Owen Rambow, Seth Kulick, and Mitch Marcus. 2007. Determining case in Arabic: Learning complex linguistic behavior requires complex linguistic features. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1,084–1,092, Prague.

Habash, Nizar 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, MI.

Habash, Nizar, Owen Rambow, and Ryan Roth. 2012. MADA+TOKAN Manual. Technical Report CCLS-12-01, Columbia University, New York, NY.

Habash, Nizar and Ryan Roth. 2009. CATiB: The Columbia Arabic treebank. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 221–224, Suntec.

Habash, Nizar, 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*, pages 15–22. Springer, Berlin.

Hajič, Jan 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, Stroudsburg, PA.

Hohensee, Matt and Emily M. Bender. 2012. Getting more from morphology in multilingual dependency parsing. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 315–326, Montréal.

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

Kulick, Seth, 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.

Maamouri, Mohamed, 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.

Marton, Yuval, 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)*, pages 13–21, Los Angeles, CA.

Marton, Yuval, Nizar Habash, and Owen Rambow. 2011. Improving Arabic dependency parsing with lexical and inflectional surface and functional features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1,586–1,596, Portland, OR.

McClosky, David, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL) - Human Language Technology (HLT)*, pages 152–159, Brooklyn, New York.

Nilsson, Jens and Joakim Nivre. 2008. MaltEval: An evaluation and visualization tool for dependency parsing. In *Proceedings of the sixth Conference on Language Resources and Evaluation (LREC)*, pages 161–166, Marrakech.

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

Nivre, Joakim. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

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

Nivre, Joakim, 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, Manchester.

Nivre, Joakim, 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.

Petrov, Slav, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, pages 2,089–2,096.

Rambow, Owen, Bonnie Dorr, David Farwell, Rebecca Green, Nizar Habash, Stephen Helmreich, Eduard Hovy, Lori Levin, Keith J. Miller, Teruko Mitamura, Florence Reeder, and Siddharthan Advaith. 2006. Parallel syntactic annotation of multiple languages. In *Proceedings of the Fifth Conference on Language Resources and Evaluation (LREC)*, pages 559–564, Genoa.

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

Tsarfaty, Reut 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.

Zitouni, Imed, 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 Association for Computational Linguistics (ACL)*, pages 577–584, Sydney.

A. Appendix: Additional Feature Engineering

The following sections describe additional experiments, with negative or small gains, presented here for completeness.

A.1 Embedding Morphological Features Within the POS Tags

After discovering our best form-based feature combination, we explored whether morphological data should be added to an Arabic parsing model as stand-alone machine learning features, or whether they should be used to enhance and extend a POS tag set. We created a new POS tag set, CORE12EX, size 81 (and 96.0% prediction accuracy), by extending the CORE12 tag set with the features that most improved the CORE12 baseline: DET and the PNG-features. But CORE12EX did worse than its non-extended (but feature-enhanced) counterpart, CORE12+DET+PNG. Another variant, CORE12EX+DET+PNG, which used both the extended tag set and the additional DET and PNG-features, did not improve over CORE12+DET+PNG either.

A.2 Extended PERSON Feature

After extending the determiner feature (DET2), the next gainful feature that we could alter was PERSON. We changed the values of proper names from "N/A" to "3" (third-person). But this change resulted in a slight decrease in performance, so it was abandoned.

A.3 Digit Tokens and Number Binning

Digit tokens (e.g., 4, as opposed to four) are marked singular by default. They don't show surface agreement with a noun, even though the corresponding number-word token would. Therefore we replaced the digit tokens' NUMBER value with "N," and denoted these experiments with NUMDGT.[24]

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)

We 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.

Number binning did not have a consistent contribution in either gold or predicted value conditions (results not shown), so it was abandoned as well.

24 We didn't mark the number-words because in our training data there were fewer than 30 lemmas of fewer than 2,000 such tokens, and hence presumably their agreement patterns can be more easily learned.