Determining Case in Arabic: Learning Complex Linguistic Behavior Requires Complex Linguistic Features

Nizar Habash†, Ryan Gabbard‡, Owen Rambow†, Seth Kulick‡ and Mitch Marcus‡

†Center for Computational Learning Systems, Columbia University
New York, NY, USA
{habash,rambow}@cs.columbia.edu
‡Department of Computer and Information Science,University of Pennsylvania
Philadelphia, PA, USA
{gabbard,skulick,mitch}@cis.upenn.edu

Abstract

This paper discusses automatic determination of case in Arabic. This task is a major source of errors in full diacritization of Arabic. We use a gold-standard syntactic tree, and obtain an error rate of about 4.2%, with a machine learning based system outperforming a system using hand-written rules. A careful error analysis suggests that when we account for annotation errors in the gold standard, the error rate drops to 0.8%, with the hand-written rules outperforming the machine learning-based system.

1 Introduction

In Modern Standard Arabic (MSA), all nouns and adjectives have one of three cases: nominative (NOM), accusative (ACC), or genitive (GEN). What sets case in MSA apart from case in other languages is most saliently the fact that it is usually not marked in the orthography, as it is written using diacritics which are normally omitted. In fact, in a recent paper on diacritization, Habash and Rambow (2007) report that word error rate drops 9.4% absolute (to 5.5%) if the word-final diacritics (which include case) need not be predicted. Similar drops have been observed by other researchers (Nelken and Shieber, 2005; Zitouni et al., 2006). Thus, we can deduce that tagging-based approaches to case identification are limited in their usefulness, and if we need full diacritization for subsequent processing in a natural language processing (NLP) application (say, language modeling for automatic speech recognition (Vergyri and Kirchhoff, 2004)), we need to perform more complex syntactic processing to restore case diacritics. Options include using the output of a parser in determining case.

An additional motivation for investigating case in Arabic comes from treebanking. Native speakers of Arabic in fact are native speakers of one of the Arabic dialects, all of which have lost case (Holes, 2004). They learn MSA in school, and have no native-speaker intuition about case. Thus, determining case in MSA is a hard problem for everyone, including treebank annotators. A tool to catch case-related errors in treebanking would be useful.

In this paper, we investigate the problem of determining case of nouns and adjectives in syntactic trees. We use gold standard trees from the Arabic Treebank (ATB). We see our work using gold standard trees as a first step towards developing a system for restoring case to the output of a parser. The complexity of the task justifies an initial investigation based on gold standard trees. And of course, the use of gold standard trees is justified for our other objective, helping quality control for treebanking.

The study presented in this paper shows the importance of what has been called "feature engineering" and the issue of representation for machine learning. Our initial machine learning experiments use features that can be read off the ATB phrase structure trees in a straightforward manner. The literature on case in MSA (prescriptive and descriptive sources) reveals that case assignment in Arabic does not always follow standard assumptions about predicate-argument structure, which is what

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1084–1092, Prague, June 2007.

the ATB annotation is based on. Therefore, we transform the ATB so that the new representation is based entirely on case assignment, not predicate-argument structure. The features for machine learning that can now be read off from the new representation yield much better results. Our results show that we can determine case with an error rate of 4.2%. However, our results would have been impossible without a deeper understanding of the linguistic phenomenon of case and a transformation of the representation oriented towards this phenomenon.

Using either underlying representation, machine learning performs better than hand-written rules. However, a closer look at the errors made by the machine learning-derived classifier and the hand-written rules reveals that most errors are in fact treebank errors (between 69% and 86% of all errors for the machine learning-derived classifier and the hand-written rules, respectively). Furthermore, the machine learning classifier agrees more often with treebank errors than the hand-written rules do. This fact highlights the problem of machine learning (garbage in, garbage out), but holds out the prospect for improvement in the machine learning based classifier as the treebank is checked for errors and re-released.

In the next section, we describe all relevant linguistic facts of case in Arabic. Section 3 details the resources used in this research. Section 4 describes the preprocessing done to extract the relevant linguistic features from the ATB. Sections 5 and 6 detail the two systems we compare. Sections 7 and 8 present results and an error analysis of the two systems. And we conclude with a discussion of our findings in Section 9.

2 Linguistic Facts

All Arabic nominals (common nouns, proper nouns, adjectives and adverbs) are inflected for case, which has three values in Arabic: nominative (NOM), accusative (ACC) or genitive (GEN). We know this from case agreement facts, even though the morphology and/or orthography do not necessarily always make the case realization overt. We discuss morphological and syntactic aspects of case in MSA in turn.

2.1 Morphological Realization of Case

The realization of nominal case in Arabic is complicated by its orthography, which uses optional diacritics to indicate short vowel case morphemes, and by its morphology, which does not always distinguish between all cases. Additionally, case realization in Arabic interacts heavily with the realization of definiteness, leading to different realizations depending on whether the nominal is indefinite, i.e., receiving *nunation* (تنوين), definite through the determiner *Al*+ (+ال) or definite through being the governor of an idafa possessive construction (إضافة). Most details of this interaction are outside the scope of this paper, but we discuss it as much as it helps clarify issues of case.

Buckley (2004) describes eight different classes of nominal case expression, which we briefly review. We first discuss the realization of case in morphologically singular nouns (including broken, i.e., irregular, plurals). *Triptotes* are the basic class which expresses the three cases in the singular using the three short vowels of Arabic: NOM is ـُ +*u*,[1] ACC is ـَ +*a*, and GEN is ـِ +*i*. The corresponding nunated forms for these three diacritics are: ـٌ +*ũ* for NOM, ـً +*ã* for ACC, and ـٍ +*ĩ* for GEN. Nominals not ending with Ta Marbuta (ة *ħ*) or Alif Hamza (اء *A'*) receive an extra Alif in the accusative indefinite case (e.g, كِتابًا *kitAbAã* 'book' versus كِتابَةً *kitAbaħã* 'writing').

Diptotes are like triptotes except that when they are indefinite, they do not express nunation and they use the ـَ +*a* suffix for both ACC and GEN. The class of diptotes is lexically specific. It includes nominals with specific meanings or morphological patterns (colors, elatives, specific broken plurals, some proper names with Ta Marbuta ending or location names devoid of the definite article). Examples include بيروت *bayruwt* 'Beirut' and أزرق *Âazraq*

[1] All Arabic transliterations are provided in the Habash-Soudi-Buckwalter transliteration scheme (Habash et al., 2007). This scheme extends Buckwalter's transliteration scheme (Buckwalter, 2002) to increase its readability while maintaining the 1-to-1 correspondence with Arabic orthography as represented in standard encodings of Arabic, i.e., Unicode, CP-1256, etc. The following are the only differences from Buckwalter's scheme (which is indicated in parentheses): Ā آ (|), Â أ (>), ŵ ؤ (&), Ǎ إ (<), ŷ ئ (}), ħ ة (p), θ ث (v), ð ذ (*), š ش ($), Ď ظ (Z), ς ع (E), γ غ (g), ý ى (Y), ã ـً (F), ũ ـٌ (N), ĩ ـٍ (K).

'blue'.

The next three classes are less common. The *invariables* show no case in the singular (e.g. nominals ending in long vowels: سوريا *suwryA* 'Syria' or ذكرى *ðikraý* 'memoir'). The *indeclinables* always use the ـَ +*a* suffix to express case in the singular and allow for nunation (معنًى *maςnaýã* 'meaning'). The *defective* nominals, which are derived from roots with a final radical glide (*y* or *w*), look like triptotes except that they collapse NOM and GEN into the GEN form, which also includes loosing their final glide: قاضٍ *qADĩ* (NOM,GEN) versus قاضيًا *qADiyAã* (ACC) 'a judge'.

For the dual and sound plural, the situation is simpler, as there are no lexical exceptions. The duals and masculine sound plurals express number, case and gender jointly in single morphemes that are identifiable even if undiacritized: كاتِبُونَ *kAtib+uwna* 'writers$_{masc,pl}$' (NOM), كاتِبَان *kAtib+Ani* 'writers$_{masc,du}$' (NOM), كاتِبَتَانِ *kAtib+atAni* 'writers$_{fem,du}$' (NOM). The ACC and GEN forms are identical, e.g., كاتِبِينَ *kAtib+iyna* 'writers$_{masc,pl}$' (ACC,GEN). Finally, the dual and masculine sound plural do not express nunation. On the other hand, the feminine sound plural marks nunation explicitly, and all of its case morphemes are written only as diacritics, e.g., كاتِباتُ *kAtib+At+u* 'writers$_{fem,pl}$' (NOM).

2.2 Syntax of Case

Traditional Arabic grammar makes a distinction between *verbal clauses* (جمل فعلية) and *nominal clauses* (جمل آسمية). Verbal clauses are verb-initial sentences, and we (counter to the Arabic grammatical tradition) include copula-initial clauses in this group. The copula is كان *kAn* 'to be' or one of her sisters. Nominal clauses begin with a topic (which is always a nominal), and continue with a complement which is either a verbal clause, a nominal predicate, or a prepositional predicate. If the complement of a topic is a verbal clause, an inflectional subject morpheme or a resumptive object clitic pronoun replace the argument which has become the topic.

Arabic case system falls within the class of nominative-accusative languages (as opposed to ergative-absolutive languages). Some of the common behavior of case in Arabic with other languages includes:[2]

- NOM is assigned to subjects of verbal clauses, as well as other nominals in headings, titles and quotes.
- ACC is assigned to (direct and indirect) objects of verbal clauses, verbal nouns, or active participles; to subjects of small clauses governed by other verbs (i.e., "exceptional case marking" or "raising to object" contexts; we remain agnostic on the proper analysis); adverbs; and certain interjections, such as شكرًا *šukrAã* 'Thank you'.
- GEN is assigned to objects of prepositions and to possessors in idafa (possessive) construction.
- There is a distinction between case-by-assignment and case-by-agreement. In case-by-assignment, a specific case is assigned to a nominal by its case assigner; whereas in case-by-agreement, the modifying or conjoined nominal copies the case of its governor.

Arabic case differs from case in other languages in the following conditions, which relate to nominal clauses and numbers.

- The topic (independently of its grammatical function) is ACC if it follows the subordinating conjunction إنّ *Ăin∼a* (or any of her "sisters": لأنّ *liÂan∼a*, كأنّ *kaÂan∼a*, لكنّ *lakin∼a*, etc.). Otherwise, the topic is NOM.
- Nominal predicates are ACC if they are governed by the overt copula. They are also ACC if they are objects of verbs that take small clause complements (such as 'to consider'), unless the predicate is introduced by a subordinating conjunction. In all other cases, they are NOM.
- In constructions involving a nominal and a number (عِشرُونَ كاتِبًا *Eišruwna kAtibAã* 'twenty writers'), the head of the phrase for case assignment is the number, which receives whichever case the context assigns. The case of the nominal depends on the number. If the number is between 11 and 99, the nominal is

[2] Buckley (2004) describes in detail the conditions for each of the three cases in Arabic. He considers NOM to be the default case. He specifies seven conditions for NOM, 25 for ACC and two for GEN. Our summary covers the same ground as his description except that we omit the vocative use of nominals.

ACC by *tamiyz* (تمييز – lit. “specification”). Otherwise, the nominal is GEN by idafa.

3 The Data

We use the third section of the current version of the Arabic Treebank released by the Linguistic Data Consortium (LDC) (Maamouri et al., 2004). We use the division into training and devtest corpora proposed by Zitouni et al. (2006), further dividing their devtest set into two equal parts to give us a development and a test set. The training set has approximately 367,000 words, and the development and test sets each have about 33,000 words. In our training data, of 133,250 case-marked nominals, 66.4% are GEN, 18.5% ACC, and 15.1% NOM.

The ATB annotation in principle indicates for each nominal its case and the corresponding realization (including diacritics). The only systematic exception is that invariables are not marked at all with their unrealized case, and are marked as having NOCASE. We exclude all nominals marked NOCASE from our evaluations, as we believe that these nominals actually do have case, it is just not marked in the treebank, and we do not wish to predict the morphological realization, only the underlying case. In reporting results, we use accuracy on the number of nominals whose case is given in the treebank.

While the ATB does not contain explicit information about headedness in its phrase structure, we can say that the syntactic annotations in the ATB are roughly based on predicate-argument structure. For example, for the structure shown in Figure 1, the “natural” interpretation is that the head is احتراقُ *AHtrAqu* ‘burning’, with a modifier منزلاً *mnzlAã* ‘house’, which in turn is modified by a QP whose head is (presumably) the number 20, which is modified by اكثرِ *Akθri* ‘more’ and من *mn* ‘than’. This dependency structure is shown on the left in Figure 2. Another annotation detail relevant to this paper is that the ATB marks the topic of a nominal clause as “SBJ” (i.e., as a subject) except when the predicate is a verbal clause; then it is marked as TPC. We consider these two cases to be the same case and relabel all such cases as TPC.

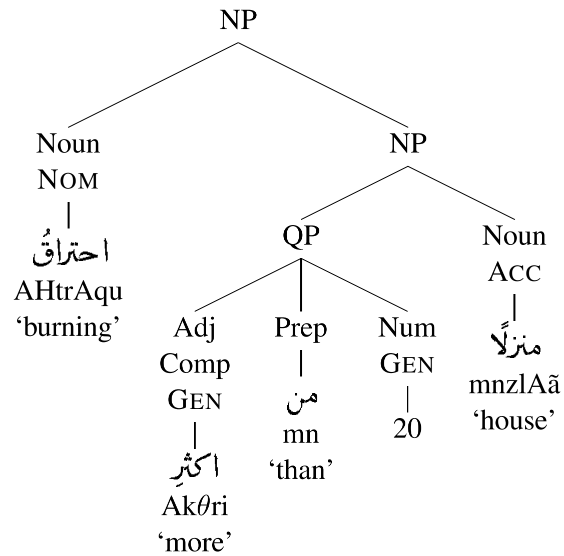

Figure 1: The representation of numbers in the Arabic Treebank, for a subject NP meaning ‘the burning of more than 20 houses’

4 Determining the Case Assigner

Case assignment is a relationship between two words: one word (the case governor or assigner) assigns case to the other word (the case assignee). Because case assignment is a relationship between words, we switch to a dependency-based version of the treebank. There are many possible ways to transform a phrase structure representation into a dependency representation; we explore two such conversions in the context of this paper. Note that if we had used the Prague Arabic Dependency Treebank (Smrž and Hajič, 2006) instead of the ATB, we would not have had to convert to dependency, but we still would have had to analyze whether the dependencies are the ones we need for modeling case assignment, possibly having to restructure the dependencies.

For determining the dependency relations that determine case assignment, we start out by using a standard head percolation algorithm with the following parameters: Verbs head all the arguments in VPs; prepositions head the PP arguments; and the first nominal in an NP or ADJP heads those structures. Non-verbal predicates (NPs, ADJPs or PPs) head their subjects (topics). The subordinating conjunction إنّ *Ăin∼a* is governed by what follows it. The overt copula كان *kAn* governs both topic and predicate. Conjunctions are headed by what they follow and head what they precede (with the exception of the common sentence initial conjunction و+ *w+* 'and', which is headed by the sentence it introduces). We will call the result of this algorithm the **Basic Case Assigner Identification Algorithm**, or **Basic Representation** for short.

After initial experiments with both hand-written rules and machine learning, we extend the Basic Representation in order to account for the special case assigning properties of numbers in Arabic by adding additional head percolation parameters and restructuring rules to handle the structure of NPs in the ATB. This is because the current ATB representation is not useful in some cases for representing case assignment. Consider the structure in Figure 1. Here, the head of the NP is the noun احتراقُ *AHtrAqu* 'burning', which has NOM because the NP is a subject (the verb is not shown). The QP's first member, اكثرِ *Akθri* 'more' is GEN because it is in an idafa construction with the noun احتراقُ *AHtrAqu*. اكثرِ *Akθri* is modified by the preposition من *mn* 'than' which assigns GEN to the number 20 (which is written in Arabic numerals and thus does not show any case at all). The noun منزلاً *mnzlAã* 'house' is in a tamyiz relation with the number 20 which governs it, and thus it is ACC. It is clear that the phrase structure chosen for the ATB does not represent these case-assignment relations in a direct manner.

To create the appropriate head relations for case determination, we flatten all QPs and use a set of simple deterministic rules to create the more appropriate structure which expresses the chain of case assignments. In our development set, 5.8% of words get a new head using this new head assignment. We call this new representation the **Revised Representation**. Figure 2 shows the dependency representation corresponding to the phrase structure in Figure 1.

We make use of all dash-tags provided by the ATB as arc labels and we extend the label set to explicitly mark objects of prepositions (POBJ), possessors in idafa construction (IDAFA), conjuncts (CONJ) and conjunctions (CC), and the accusative specifier, tamyiz (TMZ). All other modifications receive the label (MOD).

5 Hand Written Rules

Our first system is based on hand-written rules (henceforth, we refer to this system as the rule-based system). We add two features to nominals in the tree: (1) we identify if a word governs a subordinating conjunction إِنَّ *Ǎin∼a* or any of its sisters; and (2) we also identify if a topic of a nominal sentence has an *Ǎin∼a* sibling.

The following are the simple hand written rules we use:

- RULE 1: The default case assigned is ACC for all words.
- RULE 2: Assign NOM to nominals heading the tree and those labeled HLN (headline) or TTL (title).
- RULE 3: Assign GEN to nominals with the labels POBJ or IDAFA.
- RULE 4: Assign NOM to nominals with the label PRD if NOT headed by a verbal (verb or deverbal noun) or if it has an *Ǎin∼a* child.
- RULE 5: Assign NOM to nominal topics that do not have an *Ǎin∼a* sibling.
- RULE 6: All case-unassigned children of nominal parents (and conjunctions), whose label is MOD, CONJ or CC, copy the case of their parent. Conjunctions carry the case temporarily to pass on agreement. Verbs do not pass on agreement.

The first rule is applied to all nodes. The second to fifth rules are case-by-assignment rules applied in an if-else fashion (no overwriting is done). The last rule is a case-by-agreement rule. All non-nominals receive the case NA.

6 Machine Learning Experiments: The Statistical System

Our second system uses statistical machine learning. This system consists of a core model and an agreement model, both of which are linear classifiers trained using the maximum entropy technique. We implement this system using the MALLET toolbox (McCallum, 2002). The core model is used to classify all words whose label in the dependency representation is *not* MOD (case-by-assignment); whereas, the agreement model is used to classify all words

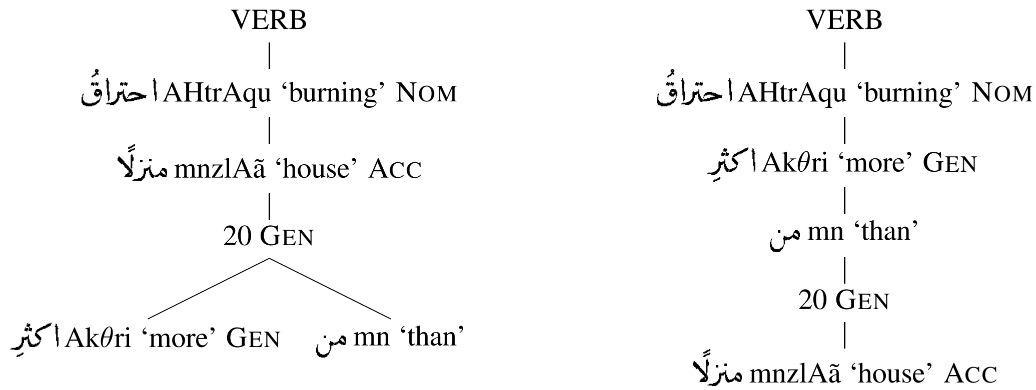

Figure 2: Two possible dependency trees for the phrase structure tree in Figure 1, meaning ‘burning of more than 20 houses’; the tree on the left, our Basic Representation, represents a standard predicate-argument-modification style tree, while the tree on the right represents the chain of case assignment and is our Revised Representation

whose label is `MOD` (case-by-agreement). We handle conjunctions in the statistical system differently from the rule-based system: we resolve conjunctions so that conjoined words are labeled exactly the same. For example, in *John and Mary went to the store*, both *John* and *Mary* would have the subject label, even though *Mary* has a conjunction label in the raw dependency tree. Both models are trained only on those words which are marked for case in the treebank.

6.1 The Core Model

The core model uses the following features of a word:

- the word’s POS tag;
- the conjunction of the word’s POS tag and its arc label;
- the word’s last length-one and length-two suffixes (to model written case morphemes);
- the conjunction of the word’s arc label, its POS tag, and its parent’s POS tag;
- if the word is the object of a preposition, the preposition it is the object of;
- whether the word is a PRD child of a verb (with the identity of that verb conjoined if so);
- if the word has a sister which is a subordinating conjunction, and if so, that conjunction conjoined with its arc label;
- whether the word is in an embedded clause conjoined with its arc label under the verb of the embedded clause;
- if the word is a PRD child of a verb, the verb;
- the word’s left sister’s POS tag conjoined with this word’s arc label and its sister’s arc label;
- whether the word’s sister depends on the word or something else;
- and the left sister’s terminal symbol.

Arabic words which do not overtly show case are still determined for purposes of resolving agreement. The classifier is applied to these cases at runtime anyway.

6.2 The Agreement Model

The agreement model uses the following features of a word:

- the word itself;
- the word’s last length-one and length-two suffixes;
- and the conjunction of the word’s POS tag and the case of what it agrees with.

Since words may get their case by agreement with other words which themselves get their case by agreement, the agreement model is applied repeatedly until case has been determined for all words.

System	Basic	Revised
Rule-based	93.5	94.4
Statistical	**94.0**	**95.8**

Table 1: Accuracies of various approaches on the test set in both basic and revised dependency representations.

7 Results

The performance of our two systems on the test data set is shown in table 1. There are three points to note: first, even in the basic representation, the statistical system reduces error over the rule-based system by 7.7%. Second, the revised representation helps tremendously, resulting in a 13.8% reduction in error for the rule-based system and 30% for the statistical system. Finally, the statistical system gains much more than the rule-based system from the improved representation, increasing the gap between them to a 25% reduction in error.

8 Error Analysis

We took a sample of 105 sentences (around 10%) from our development data prepared in the revised representation. Our rule-based system accuracy for the sample is about 94.1% and our statistical system accuracy is 96.2%. Table 2 classifies the different types of errors found. The first and second rows list the errors made by the statistical and rule-based systems, respectively. The third row lists errors made by the statistical system only. The fourth row lists errors made by the rule-based system only. And the fifth row lists errors made by both. The second column indicates the count of all errors. The rest of the columns specify the error types as: system errors, gold POS errors or gold tree errors. The gold POS and tree errors are treebank errors that misguide our systems. They represent 69% of all statistical system errors and 86% of all rule-based system errors. Gold POS errors represent around 35-40% of all gold errors. They most commonly include the wrong POS tag or the wrong case. One example of such errors is the mis-annotation of the ACC case to a GEN for a diptote nominal (which are indistinguishable out of context). Gold tree errors are primarily errors in the dash-tags used (or missing) in the treebank or attachment errors that are inconsistent with the gold POS tag.

The rule-based system errors involve various constructions that were not addressed in our study, e.g. flat adjectival phrases or non S constructions at the highest level in a tree (e.g. FRAG or NP). The majority of the statistical system errors involve agreement decisions and incorrect choice of case despite the presence of the dash-tags. The ratio of system errors for the statistical system is 31% (twice as much as those of the rule-based system's 14%). Thus, it seems that the statistical system manages to learn some of the erroneous noise in the treebank.

9 Discussion

9.1 Accomplishments

We have developed a system that determines case for nominals in MSA. This task is a major source of errors in full diacritization of Arabic. We use a gold-standard syntactic tree, and obtain an error rate of about 4.2%, with a machine learning based system outperforming a system using hand-written rules. A careful error analysis suggests that when we account for annotation errors in the gold standard, the error rate drops to 0.8%, with the hand-written rules outperforming the machine learning-based system.

9.2 Lessons Learned

We can draw several general conclusions from our experiments.

- The features relevant for the prediction of complex linguistic phenomena cannot necessarily be easily read off from the given representation of the data. Sometimes, due to data sparseness and/or limitations in the machine learning paradigm used, we need to extract features from the available representation in a manner that profoundly changes the representation (as is done in bilexical parsing (Collins, 1997)). Such transformations require a deep understanding of the linguistic phenomena on the part of the researchers.
- Researchers developing hand-written rules may follow an empirical methodology in natural language processing if they use data sets to develop and test the rules — the only true methodological difference between machine learning and this kind of hand-writing of rules

ERRORS	COUNT	SYSTEM	GOLD POS	GOLD TREE
All Statistical	45	14	11	20
All Rule-based	70	10	24	36
Statistical only	13	11	0	2
Rule-based only	38	7	13	18
Statistical $\bigcap$ Rule-based	32	3	11	18

Table 2: Results of Error Analysis

is the type of learning (human or machine). For certain phenomena, machine learning may result in only a small or no improvement in performance over hand-written rules.

- Error analysis remains a crucial part of any empirical work in natural language processing. Not only does it contribute insight into how the system can be improved, it also reveals problems with the underlying data. Sometimes the problems are just part of the noise in the data, but sometimes the problems can be fixed. Annotations on data are not themselves naturally occurring data and thus may be subject to critique. Note that an error analysis requires a good understanding of the linguistic phenomena and of the data.

9.3 Outlook

Our work was motivated in two ways: to help treebanking, and to develop tools for automatic case determination from unannotated text. For the first goal, our error analysis has shown that 86% of the errors found by our hand-written rules are in fact treebank errors. Furthermore, we suspect that the hand-written rules have very few false positives (i.e., cases in which the treebank has been annotated in error but our rules predict exactly that error). Thus we believe that our tool can serve an important function in improving the treebank annotation.

For our second motivation, the next step will be to adapt our feature extraction to work on the output of parsers, which typically exclude dash-tags. We note that for many contexts, we do not currently rely on dash-tags but rather identify the relevant structures on our own (such as idafa, tamyiz, and so on). We suspect that the machine learning-based approach will outperform the hand-written rules, as it can learn typical errors the parser makes. As the treebank will soon be revised and hand-checked, we will postpone this work until the new release of the treebank, which will allow us to train better parsers as the data will be more consistent.

Acknowledgements

The research presented here was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract Nos. HR0011-06-C-0023, HR0011-06-C-0022 and HR0011-06-1-0003. Any opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of DARPA.

References

Ron Buckley. 2004. *Modern Literary Arabic: A Reference Grammar*. Librairie du Liban.

Tim Buckwalter. 2002. Buckwalter Arabic morphological analyzer version 1.0.

Michael Collins. 1997. Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the ACL (jointly with the 8th Conference of the EACL)*, pages 16–23, Madrid, Spain.

Nizar Habash and Owen Rambow. 2007. Arabic Diacritization through Full Morphological Tagging. In *Proceedings of the 8th Meeting of the North American Chapter of the Association for Computational Linguistics/Human Language Technologies Conference (HLT-NAACL07)*.

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

Clive Holes. 2004. *Modern Arabic: Structures, Functions, and Varieties*. Georgetown University Press. Revised Edition.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank :

Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Rani Nelken and Stuart Shieber. 2005. Arabic Diacritization Using Weighted Finite-State Transducers. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages at 43rd Meeting of the Association for Computational Linguistics (ACL'05)*, pages 79–86, Ann Arbor, Michigan.

Otakar Smrž and Jan Hajič. 2006. The Other Arabic Treebank: Prague Dependencies and Functions. In Ali Farghaly, editor, *Arabic Computational Linguistics: Current Implementations*. CSLI Publications.

Dimitra Vergyri and Katrin Kirchhoff. 2004. Automatic Diacritization of Arabic for Acoustic Modeling in Speech Recognition. In Ali Farghaly and Karine Megerdoomian, editors, *COLING 2004 Workshop on Computational Approaches to Arabic Script-based Languages*, pages 66–73, Geneva, Switzerland.

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