Arabic Morphological Analyzer with Agglutinative Affix Morphemes and Fusional Concatenation Rules

Fadi Zaraket1  JadMakhlouta1
(1) American University of Beirut, Lebanon
{fz11, jem04}@aub.edu.lb

Abstract
Current concatenative morphological analyzers consider prefix, suffix and stem morphemes based on lexicons of morphemes, and morpheme concatenation rules that determine whether prefix-stem, stem-suffix, and prefix-suffix concatenations are allowed. Existing affix lexicons contain extensive redundancy, suffer from inconsistencies, and require significant manual work to augment with clitics and partial affixes if needed. Unlike traditional work, our method considers Arabic affixes as fusional and agglutinative, i.e. composed of one or more morphemes, introduces new compatibility rules for affix-affix concatenations, and refines the lexicons of the SAMA and BAMA analyzers to be smaller, less redundant, and more consistent. It also automatically and perfectly solves the correspondence problem between the segments of a word and the corresponding tags, e.g. part of speech and gloss tags.

Keywords: morphology; lexicons; computational linguistics; Arabization; affix.

Keywords in L2: علم الألسنة الحسابي; التعريب; الملحق الإضافي.
1 Short Summary in Arabic

ملخص باللغة العربية

تتطلب تقنيات معالجة اللغات الطبيعية لتحليل الص.serviceي لlausص السويسية العربية. وذلك بسبب الغياب الصوني للغة العربية إلى جانب مصادر أخرى للغموض، منها غياب الحركات في أكثر النصوص: المجالات الصفرية المتواجدة حاليا لللغة العربية تأخذ كمية معروفة وتحسب مكوناتها الصورية على شكل عدة حلول لكل منها تعيينات معينة وموحية مرتبطة بتشكيل كطل للكلمة الأصل. تعاني هذه المجالات من مشاكل عدة: أولها عدم قدرتها على عزل الكلمة، وثانيها عدم قدرتها على حل مشكلة الكلمات المتشابهة بدون فراق بينها، وثالثها عدم قدرتها على المطابقة الدقيقة بين أجزاء الكلمة ومواقع الخروف في أصل الكلمة، ورابعها خلل في الدقة في ربط التسويقات المختلفة لأصول الكلمات.

المجالات الصورية العاصرة (2010a) (Buckwalter, 2002; Kullick et al., 2010a) تعتمد على معامج المتصلة بالأدية، ولأصول الكلمات، ولالعاب اللاحقة وعلى قواعد اتصال تتحك صحة أصال متعلقة بادية بواض، واتصال أصل متعلق لاحق، وإتصال متعلق بادية متعلق لاحق. كما يبدو في الجدول 1 كل خانة في المعجم تحتوي على الكلمة الصورية، وعلى تشكيله بحركات كلمة، وعلى فتحنها التي تتحكم اتصال السكونات أخرى، وعلى تعليق يوجد موضع في الكلام، وعلى تعليق يوجد معناها. تحتوي المجالات على متعلقات ثنائية مكونة من متعلقات جزئية مثل: الكونان "ف" و "ب".
متعلقات مستقلة قابلة للاتصال مباشرة بالأصل "اللعب" لتكوين كلمتي "اللعب" و "لعب" اضافة الى ذلك، يمكن للمكون "س" أن ينتمي ل "اللعب" لتكوين "سيليعب" بدوره المكون "ف" يمكن أن ينتمي ل "سيليعب" تحتوي معاجم SAMA و BAMA الممكن تكونها ما يؤدي إلى المشاكل التالية:

أولاً، يؤدي تكرار الحنانات إلى صعوبة الحفاظ على تناسقها وفصائها.
ثانياً، يؤدي التكرار إلى تضخم المعجم خاصة عندما ننظر إلى إضافة مكون جديد إليه كومة الاستخدام.
ثالثاً، تحتوي كل حبجة على تعليمات معنوية مرتبطة بالكلمة المرتبطة. تكيب Diese التعليمات اتصالًا يؤدي إلى فقدان المتابعة الدقيقة بين مواقع الحروف في الكلمة الأم والتعليمات، وهو أمر هام جداً للتعلم الآلي.

في هذا البحث، نذكر المساهمات التالية:

أولاً، نبني مجالاً صريحاً حيث يعتمد المكونات الصوفية الاساسية ويصل بينهما، كما يحدد قواعد لديها مستقلة من كتب الصرف العربية.

ثانياً، نحل مشاكل الاختلاف بين المكونات في SAMA و BAMA وندرس أثر تصحيحاتها.

ثالثاً، نحل مشكلة المتابعة بين الكلمة الأم والتعليمات المعنوية، والتعليمات التي تحدد موقع الكلمة في النص.

النهج الذي نعتمده في التحليل الصوفي يمكن أن يعبر عن نفس الحنانات في الجدول 3، يتعامل 3 مكونات أساسية وكون واحد جزئي و 3 قواعد الاتصال جزئي، حيث أن الاتصال بين الكلمات يمكن أن يكون مع قواعد الاتصال خاصة بكل خانة، بينما لا يحتاج الأمر إلا إلى خانة واحدة باستخدام منهجنا.

في ما يلي نشرح تطبيق منهجنا في برنامج صرف آل حديث لللغة العربية ونعرض تقييمنا للنتائج:

استطاع البرنامج تطبيق المعاجم التي تحتاجها للتحليل الصوفي للغة العربية، واستطاع تصحيح اخطاء التناسق الموجودة في أداة التحليل الآلي الحالية، كما استطاع أن يحل مشكلة المتابعة الفورية بين الكلمة الأم ومكوناتها الصوفية في مقابل التعليمات وأجزاءها.

شكر:

شكر المجلس الوطني اللبناني للبحوث لدعمه هذا العمل.


2 Introduction

Natural language processing (NLP) applications require the use of morphological analyzers to preprocess Arabic text (Benajiba et al., 2007; Habash and Sadat, 2006). Given a white space and punctuation delimited Arabic word, Arabic morphological analyzers return the internal structure of the word composed of several morphemes including affixes (prefixes and suffixes) and stems (Al-Sughaiyer and Al-Kharashi, 2004). They also return part of speech (POS) and other tags associated with the word and its constituent morphemes. For example, for the word فسيعبون (fsylbwn (and/so they will play), the analyzer may return قسم f sy as a prefix morpheme with the POS tag fa/CONJ+sa/FUT+ya/IV3MD and with gloss tag and/so + will + they (people), لعب l b as a stem with POS tag loEab/VERB IMPERFECT and with gloss tag play, and ون wn as a suffix with POS tag uwna/IVSUFF_SUBJ:MP_MOOD:I and with gloss tag [MASC.PL.]. The alignment and correspondence between the original word and the several parts and tags of the morphological solution are essential to the success of NLP tasks such as machine translation and information extraction (Lee et al., 2011; Semmar et al., 2008).

Current concatenative morphological analyzers such as BAMA (Buckwalter, 2002) and SAMA (Kulick et al., 2010a) are based on lexicons of prefixes $L_p$, stems $L_s$, and suffixes $L_x$. As shown in Table 1, each entry in a lexicon includes the morpheme, its vocalized form with diacritics, a concatenation compatibility category tag, a part of speech tag (POS), and the gloss tag. Separate compatibility rules specify the compatibility of prefix-stem $R_{ps}$, stem-suffix $R_{sx}$, and prefix-suffix $R_{px}$ concatenations. The affixes in $L_p$ and $L_x$ contain final forms of generative affixes. For example, the affixes $\bar{f}$ (and/so), and $\check{y}$ (he/it) in the above example are valid standalone prefixes, and can be concatenated to the stem لعب l b to form قلعب f l b and ولعب y l b , respectively. In addition, the morpheme س s (will) can connect to لعب y l b to form فيلعب syl b . In turn, the morpheme $\bar{f}$ (and/so) can form فيلعب fyl b and فيلعب fsyl b . The BAMA and SAMA $L_p$ lexicons contain all the prefixes that can be generated from the three morphemes $\check{y}$, $\bar{f}$, and $\check{s}$, as shown in Table 1. Several problems arise.

- The $L_p$ and $L_x$ lexicons contain redundant entries and that results in complex maintenance and consistency issues (Maamouri et al., 2008; Kulick et al., 2010b).
- Augmenting $L_p$ and $L_x$ with additional morphemes, such as $\check{y}$ (the question glottal hamza), may result in a quadratic explosion in the size of the lexicons (Hunspell, 2012).
- The concatenated forms in $L_p$ and $L_x$ contain concatenated POS and other tags. The segmentation correspondence between the prefix concatenated from several morphemes and the tags associated with it is lost. In several cases, this leads to missing correspondence between the tokens of the morphological solution and the segmentation of the original word.

In this paper we make the following contributions. More details about this paper and the supporting tools are available online 1.

- We build a novel Arabic morphological analyzer with agglutinative affixes and fusional affix concatenation rules ($R_{pp}$ and $R_{xx}$) using textbook based Arabic morphological rules as well as the concatenation rules of existing analyzers. Agglutinative affix morphemes can be concatenated to form an affix. Fusional affix concatenation rules state whether two affixes can

\footnote{1http://webfea.fea.aub.edu.lb/fadi/dkwk/doku.php?id=sarf}
Table 2: Example rules from $R_{pp}$

| Category 1 | Category 2 | Resulting Category |
|------------|------------|--------------------|
| NPref-Li   | NPref-Al   | NPref-LiAl         |
| substitute: $r /| | | \| |
| Pref-Wa    | {NOT “Pref-0” AND NOT “NPref-La” AND NOT “PVPref-La”} {$2}$ |
| IVPref-li- | {“IVPref-@-@*”} {“IVPref-(@1)-liy(@2)”} |
| substitute: $d /| | | \| | $d / | | | \| | $d / | | | \| | $(+2) | | | \| | $to $ |

be concatenated and contain a regular expression that forms the resulting orthographic and semantic tags from the tags of the original morphemes (Spencer, 1991; Vajda).

- We solve 197 and 208 inconsistencies in the existing affix lexicons of BAMA and SAMA, respectively. We evaluate our approach using the ATBv3.2 Part 3 data set (Maamouri et al., 2010) and report on the effect of our corrections on the annotations.

- We solve the correspondence between the morphological solution and the morphological segmentation of the original text problem where we report perfect results, while a SAMA post-processing technique (Maamouri et al., 2008) reports 3.7% and MADA+TOKAN (Habash et al., 2009) reports 9.6% disagreement using the ATBv3.2 Part 3 data set (Maamouri et al., 2010).

3 Our method

Our method considers three types of affixes:

- **Atomic** affix morphemes such as $_y$ (he/it) can be affixes on their own and can directly connect to stems using the $R_{ps}$ and $R_{sx}$ rules.

- **Partial** affix morphemes such as $_s$ (will) can not be affixes on their own and need to connect to other affixes before they connect to a stem.

- **Compound** affixes are concatenations of atomic and partial affix morphemes as well as other smaller compound affixes. They can connect to stems according to the $R_{ps}$ and $R_{sx}$ rules.

We form compound affixes from atomic and partial affix morphemes using newly introduced prefix-prefix $R_{pp}$ and suffix-suffix $R_{sx}$ concatenation rules.

Our method, unlike conventional analyzers, considers $L_p$ and $L_s$ to be lexicons of atomic and partial affix morphemes only associated with several tags such as the vocalized form, the part of speech (POS), and the gloss tags. Agglutinative affixes are defined as prefix-prefix $R_{pp}$ and suffix-suffix $R_{sx}$ concatenation or agglutination rules. An agglutination rule $r \in R_{pp} \cup R_{sx}$ takes the compatibility category tags of affixes $a_1$ and $a_2$ and checks whether they can be concatenated. If so, the rule takes the tags of $a_1$ and $a_2$ and generates the affix $a = r(a_1,a_2)$ with its associated tags.

The tags of $r(a_1,a_2)$ are generated from the corresponding tags of $a_1$ and $a_2$ via applying substitution rules. Our rules are fusional in the sense that they modify the orthography and the semantic tags of the resulting affixes by more than simple concatenation.

We illustrate this with the example rules in Table 2. Row 1 presents a rule that takes prefixes with category **NPref-Li** such as $l$ ($li$- (for) and prefixes with category **NPref-Al** such as $l$ (the).
The substitution rule replaces the Ë with Ê resulting in ÊË li-. The compound prefix ÊË corresponds to the fusion of two atomic prefixes and the fusion is one character shorter than the concatenation.

Row 2 states that prefixes of category Pref-Wa can be concatenated with prefixes with categories that are neither of Pref-0, NPref-La, and PVPref-La categories as denoted by the Boolean expression. The resulting category is denoted with \{S2\} which means the category of the second prefix. For example, و w (and) which has a category Pref-Wa, can be combined with ال al (the) with the category NPref-Al, and the resulting compound prefix و al has the category of the second NPref-Al. This category determines concatenation with stems and suffixes.

The third rule uses a wild card character \('\star'\) to capture substrings of zero or more characters in the second category. In the resulting category, it refers to the \(i^{th}\) substring captured by the wild cards using the \(\theta\) operator followed by a number \(i\). Substitution rules for gloss and POS tags start with the letters \(d\) and \(p\), respectively. The \(+2\) pattern in the substitution rule means that the partial gloss \(to\) should be appended after the gloss of the second affix.

Our method is in line with native Arabic morphology and syntax textbooks (Mosaad, 2009; AlRajehi, 2000b,a) which introduce only atomic and partial affixes and discuss rules to concatenate the affixes, and the syntax, semantic, and phonological forms of the resulting affixes. For example, Row 3 in Table 2 translates the textbook rule: IVPref-li- prefixes connect to imperfect verb prefixes and transform the subject pronoun (in the gloss) to an object pronoun. We built our rules in four steps.

1. We encoded textbook morphological rules into patterns.
2. We extracted atomic and partial affixes from the BAMA and SAMA lexicons.
3. We grouped the rest of the BAMA and SAMA affixes into rules we collected from textbooks.
4. We refined the rules wherever necessary, and we grouped rules that shared the same patterns.

We validated our work by generating all possible affixes and compared them against the BAMA and SAMA affix lexicons. This helped us learn inconsistencies in the BAMA and SAMA lexicons.

**Morpheme level segmentation.** (Habash et al., 2009) lists 13 different valid segmentation schemes. In 10 of those schemes, a word may be segmented in the middle of a compound affix. According to the latest ATB standards, the word وسيلة وفاة (and they will play it) should be segmented into و س يلة + و + و فاة which separates the compound prefix وس يلة into two morphemes. Our method is based on atomic and partial affix morphemes and enables all valid segmentations.

(Maamouri et al., 2008) reports that 3.7% of more than 300 thousand ATB entries exhibit discrepancy between the unvocalized input string and the corresponding unvocalized form of the segmented morphological solution. The analysis of the example لقضاء لقادح, li/PREP + Al/DET + qaDA’/NOUN, (for the justice) is segmented into two tokens: 1i/PREP and Al/DET + qaDA’/NOUN. Consequently, the best approximation of the unvocalized entry of each token is ل and لقضاء, respectively, with an extra letter ل. This is not a faithful representation of the original text data and the segmentation does not correspond with that of the input text. Up until the release of ATB 3 v3.2, this correspondence problem between the unvocalized entries of segmented tokens and the input string resulted in “numerous errors” (Kulick et al., 2010b). Later work (Kulick et al., 2010b) provided an improved solution that is corpus specific as stated in further documentation notes (Maamouri et al., 2010) which also state that “it is possible that future releases either will not
include extensive checking on the creation of these INPUT STRING tree tokens, or will leave out completely such tokens.”

Our method provides a general solution for the segmentation correspondence problem since the valid compound affixes preserve the input text segmentation. In particular, a partial affix \( \text{Al/DET} \) connects to the atomic affix \( \text{li/PREP} \) and resolves the problem.

**Redundancy and Inconsistencies.** Consider the partial affix lexicon in Table 1. Our method replaces the first five rows with three atomic affix morphemes and one partial affix morpheme in \( L_p \) and three rules to generate compound morphemes in \( R_{pp} \). In the original representation, the addition of the prefix \( \text{ya-} \) (them/both) required the addition of four entries, three of them only differ in their dependency on the added \( \text{ya-} \). The addition of \( \text{w} \) required the addition of five entries. In our method, the equivalent addition of \( \text{ya-} \) requires only two rules in \( R_{pp} \) and the addition of \( \text{w} \) requires only one additional entry in \( L_p \). The difference in lexicon size is much larger when we consider the full lexicon.

We discovered a total of 197 and 208 inconsistencies in the affix lexicons of BAMA version 1.2 and SAMA version 3.2, respectively. We found a small number of these inconsistencies manually and we computed the full list via comparing \( L_p \) and \( L_x \) with their counterparts computed using our agglutinative affixes. Most of the inconsistencies are direct results of partially redundant entries with erroneous tags. We note that SAMA corrected several BAMA inconsistencies, but also introduced several new ones when modifying existing entries to meet new standards. SAMA also introduced fresh inconsistencies when introducing new entries. The full list of inconsistencies with description is available online 1.

### 4 Related work

Other morphological analyzers such as ElixirFM (Smrž, 2007), MAGEAD (Habash et al., 2005), and MADA+TOKAN (Habash et al., 2009) are based on BAMA and SAMA and use functional and statistical techniques to address the segmentation problem. (Lee et al., 2011) uses syntactic information to resolve the same problem. A significant amount of the literature on Arabic NLP uses the Arabic Tree Bank (ATB) (Maamouri and Bies, 2004) with tags from BAMA and SAMA for learning and evaluation (Shaalan et al., 2010; Benajiba et al., 2007; Al-Jumaily et al., 2011).

Several researchers stress the importance of correspondence between the input string and the tokens of the morphological solutions. Recent work uses POS tags and a syntactic morphological agreement hypothesis to refine syntactic boundaries within words (Lee et al., 2011). The work in (Grefenstette et al., 2005; Semmar et al., 2008) uses an extensive lexicon with 3,164,000 stems, stem rewrite rules (Darwish, 2002), syntax analysis, proclitics, and enclitics to address the same problem. We differ from partial solutions in (Maamouri et al., 2008; Kulick et al., 2010b) in that our segmentation is an output of the morphological analysis and not a reverse engineering of the multi-tag affixes.

TOKAN in the MADA+TOKAN (Habash et al., 2009) toolkit works as a post morphological disambiguation tokenizer. TOKAN tries to match the output of MADA, an SVM morphological disambiguation tool based on BAMA and the ATB, with a segmentation scheme selected by the user. We differ in that the segmentation is part of the morphological analysis and the segmentation can help in the disambiguation task performed later by the NLP task. We perform morpheme based segmentation, which subsumes all possible higher level segmentation schemes.

The morphological analyzer (Attia, 2006) divides morphemes into proclitics, prefixes, stems, suffixes and enclitics and supports inflections using alteration rules. We differ in that we support vocalization and provide glosses for individual morphemes.
Table 3: Lexicon size comparison.

|        | $|L_p|$ | $|R_{pp}|$ | $|L_x|$ | $|R_{xx}|$ | $\Delta_{hmz}^L$ | $\Delta_{hmz}^R$ |
|--------|------|--------|------|--------|-------------|-------------|
| BAMA   | 299  | –      | 618  | –      | 295         | –           |
| Agglutinative | 70  | 89      | 181  | 123    | 1           | 32          |
| With fusional   | 43  | 89      | 146  | 128    | 1           | 32          |
| With grouping   | 41  | 7       | 146  | 32     | 1           | 1           |
| SAMÁ   | 1325 | –      | 945  | –      | 1,296       | –           |
| Agglutinative | 107 | 129     | 221  | 188    | 1           | 38          |
| With fusional   | 56  | 129     | 188  | 194    | 1           | 38          |
| With grouping   | 53  | 18      | 188  | 64     | 1           | 1           |

5 Results

The $|L_p|$, $|L_x|$, $|R_{pp}|$, and $|R_{xx}|$ entries in Table 3 report the number of rules and the sizes of the affix lexicons needed to represent the affixes of BAMA and SAMA. The entries also report the effect of agglutinative affixes, fusional rules, and grouping of rules with similar patterns using wildcards on the size. Using our method, we only require 226 and 323 entries to represent the 917 and the 2,270 entries of BAMA and SAMA affixes with inconsistencies corrected, respectively. We observe that we only need 12 more entries in $L_p$, 42 in $L_x$, 18 rules in $R_{pp}$, and 64 in $R_{xx}$ for a total of 136 entries to accommodate for the transition from BAMA to SAMA. This is one order of magnitude less than 1,353 additional entries to SAMA. We also note that we detect most of the inconsistencies automatically and only needed to validate our corrections in textbooks and corpora.

Segmentation. We evaluate our segmentation under the guidelines of the ATBv3.2 Part 3, compared to a SAMA post processing technique (Maamouri et al., 2008), and to MADA+TOKAN (Habash et al., 2009). Our automatically generated segmentation agrees with 99.991% of the entries. We investigated the 25 entries for which our solution disagreed with the LDC annotation of the ATB, and we found out that both solutions were valid. SAMÁ+ (Maamouri et al., 2008) reports at least a 3.7% discrepancy after accounting for normalizations of several segmentation options. TOKAN disagrees with 9.6% of the words. It disregards input diacritics and performs segmentation based on the POS entries of the morphological solutions in a similar approach to (Maamouri et al., 2008). Since TOKAN is not concerned with the correspondence problem, it serves as a baseline.

Augmentation. The question clitic, denoted by the glottal sign (hamza $^{\text{a}}$), is missing in BAMA and SAMA (Attia, 2006). $\Delta_{hmz}^L$ and $\Delta_{hmz}^R$ columns show that our method only requires one more atomic affix and one more fusional rule to accommodate for the addition of the question clitic whereas BAMA and SAMA need 295 and 1,296 additional entries, respectively, with more chances of inducing inconsistencies.

Lexicon inconsistencies. To evaluate how much lexical inconsistencies are significant we evaluated the presence of the detected inconsistencies in the ATBv3.2 Part 3 and found that 0.76% of the entries that adopted the SAMA solution were affected by the gloss inconsistencies. The rest of the entries have manually entered solutions. In total 8.774% of the words and 3.264% of the morphological solutions are affected by inconsistencies in gloss and POS tags. Finally, our analyzer automatically solves the 7 ATB occurrences of the question clitic.

Acknowledgement. We thank the Lebanese National Council for Scientific Research (LNCSR) for funding this research.
References

Al-Jumaily, H., Martínez, P., Martínez-Fernández, J., and Van der Goot, E. (2011). A real time named entity recognition system for Arabic text mining. *Language Resources and Evaluation*, pages 1–21.

Al-Sughaiyer, I. A. and Al-Kharashi, I. A. (2004). Arabic morphological analysis techniques: a comprehensive survey. *American Society for Information Science and Technology*, 55(3):189–213.

AlRajehi, A. (2000a). التطبيق النحويُّ `altibyq anlahwy (The syntactical practice). Renaissance (nahda), first edition.

AlRajehi, A. (2000b). التطبيق العَرْبِيُّ `altibyq alṣarfy (The morphological practice). Renaissance (An-nahda), first edition.

Aoe, J.-i. (1989). An efficient digital search algorithm by using a double-array structure. *IEEE Transactions on Software Engineering*, 15(9):1066–1077.

Attia, M. A. (2006). An ambiguity-controlled morphological analyzer for modern standard arabic modelling finite state networks. In *The Challenge of Arabic for NLP/MT Conference*. The British Computer Society.

Beesley, K. R. (2001). Finite-state morphological analysis and generation of Arabic at xerox research: Status and plans. In *Workshop Proceedings on Arabic Language Processing: Status and Prospects*, pages 1–8, Toulouse, France.

Beesley, K. R. and Karttunen, L. (2003). *Finite-State Morphology: Xerox Tools and Techniques*. CSLI, Stanford.

Benajiba, Y., Rosso, P., and Benedíruiz, J. (2007). ANERsys: An Arabic named entity recognition system based on maximum entropy. pages 143–153.

Buckwalter, T. (2002). Buckwalter Arabic morphological analyzer version 1.0. Technical report, LDC catalog number LDC2002L49.

Darwish, K. (2002). Building a shallow Arabic morphological analyzer in one day. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*.

Grefenstette, G., Semmar, N., and Elkateb-Gara, F. (2005). Modifying a natural language processing system for european languages to treat arabic in information processing and information retrieval applications. In *ACL Workshop on Computational Approaches to Semitic Languages*, pages 31–37.

Habash, N., Rambow, O., and Kiraz, G. (2005). Morphological analysis and generation for Arabic dialects. In *Semitic ’05: Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 17–24, Morristown, NJ, USA.

Habash, N., Rambow, O., and Roth, R. (2009). Mada+tokan: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In Choukri, K. and Maegaard, B., editors, *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt. The MEDAR Consortium.
Habash, N. and Sadat, F. (2006). Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 49–52.

Hajič, J. and Zemánek, P. (2004). Prague arabic dependency treebank: Development in data and tools. In *NEMLAR International Conference on Arabic Language Resources and Tools*, pages 110–117.

Hunspell (2012). Hunspell manual page.

Kulick, S., Bies, A., and Maamouri, M. (2010a). Consistent and flexible integration of morphological annotation in the Arabic treebank. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10)*, Valletta, Malta.

Kulick, S., Bies, A., and Maamouri, M. (2010b). Consistent and flexible integration of morphological annotation in the arabic treebank. In *International Conference on Language Resources and Evaluation*. European Language Resources Association.

Lee, Y. K., Haghighi, A., and Barzila, R. (2011). Modeling Syntactic Context Improves Morphological Segmentation. In *Conference on Computational Natural Language Learning (CoNLL)*.

Lee, Y.-S., Papineni, K., Roukos, S., Emam, O., and Hassan, H. (2003). Language model based arabic word segmentation. In *Association for Computational Linguistics*, pages 399–406.

Maamouri, M. and Bies, A. (2004). Developing an Arabic treebank: methods, guidelines, procedures, and tools. In *Semitic ’04: Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, pages 2–9.

Maamouri, M., Bies, A., Kulick, S., Krouna, S., Gaddeche, F., and Zaghrouani, W. (2010). Arabic treebank: Part 3 version 3.2. In *Linguistic Data Consortium, LDC2010T08*.

Maamouri, M., Kulick, S., and Bies, A. (2008). Diacritic annotation in the arabic treebank and its impact on parser evaluation. In *International Conference on Language Resources and Evaluation*.

Mosaad, Z. (2009). الوجيز في الصرف *alwağyz fy alṣarf* (*The Briefing of Morphology*). As-Sahwa, first edition.

Semmar, N., Meriama, L., and Fluhr, C. (2008). Evaluating a natural language processing approach in arabic information retrieval. In *ELRA Workshop on Evaluation*.

Shaalan, K. F., Magdy, M., and Fahmy, A. (2010). Morphological analysis of ill-formed arabic verbs in intelligent language tutoring framework. In *Applied Natural Language Processing, Florida Artificial Intelligence Research Society Conference*. AAAI Press.

Smrž, O. (2007). Elixirfm: implementation of functional Arabic morphology. In *Semitic ’07: Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages*, pages 1–8, Prague, Czech Republic.

Spencer, A. (1991). Blackwell Textbooks in Linguistics.

Vajda, E. J. *Typology*. 