Using Ambiguity Detection to Streamline Linguistic Annotation

Wajdi Zaghouani, Abdelati Hawwari‡, Sawsan Alqahtani‡, Houda Bouamor, Mahmoud Ghoneim‡, Mona Diab‡ and Kemal Oflazer

Carnegie Mellon University in Qatar
{wajdiz,hbouamor}@qatar.cmu.edu, ko@cs.cmu.edu

‡George Washington University
{abhawwari,sawsanq,ghoneim,diab}@gwu.edu

Abstract

Arabic writing is typically underspecified for short vowels and other markups, referred to as diacritics. In addition to the lexical ambiguity exhibited in most languages, the lack of diacritics in written Arabic adds another layer of ambiguity which is an artifact of the orthography. In this paper, we present the details of three annotation experimental conditions designed to study the impact of automatic ambiguity detection, on annotation speed and quality in a large scale annotation project.

1 Introduction

Written Modern Standard Arabic (MSA) poses many challenges for natural language processing (NLP). Most written Arabic text lacks short vowels and diacritics rendering a mostly consonantal orthography (Schulz, 2004). Arabic diacritization is an orthographic way to describe Arabic word pronunciation, and avoid word reading ambiguity. In Arabic, diacritics are marks that reflect the phonological, morphological and grammatical rules. The lack of diacritics leads usually to considerable lexical and morphological ambiguity. Full diacritization has been shown to improve state-of-the-art Arabic automatic systems such as automatic speech recognition (ASR) systems (Kirchhoff and Vergyri, 2005) and statistical machine translation (SMT) (Diab et al., 2007). Hence, diacritization has been receiving increased attention in several Arabic NLP applications (Zitouni et al., 2006; Shahrour et al., 2015; Abandah et al., 2015; Belinkov and Glass, 2015). Building models to assign diacritics to each letter in a word requires a large amount of annotated training corpora covering different topics and domains to overcome the sparseness problem. The currently available MSA diacritized corpora are generally limited to religious texts such as the Holy Quran, educational texts or newswire stories distributed by the Linguistic Data Consortium.

This paper presents a work carried out within a project to create an optimal diacritization scheme for Arabic orthographic representation (OptDiac) project (Zaghouani et al., 2016a; Bouamor et al., 2015). The overreaching goal of our project is to manually create a large-scale annotated corpus with the diacritics for a variety of Arabic texts. The creation of manually annotated corpora presents many challenges and issues related to the linguistic complexity of the Arabic language. In order to streamline the annotation process, we designed various annotation experimental conditions in order to answer the following questions: Can we automatically detect linguistic difficulties such as linguistic ambiguity? To what extent is there agreement between machines and human annotators when it comes to detecting ambiguity? Can the automatic detection of the ambiguity speed up the annotation process?

In the next two sections we discuss related work (Section2) and the annotation framework (Section 3). Afterwards, we present the experimental setup in Section 4. In Section 5, we present the results of the evaluation experiment and in Section 6, we analyze the annotation disagreement errors found during the evaluation.

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/

Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity,
pages 127–136, Osaka, Japan, December 11-17 2016.

2 Background and Related Work

2.1 Arabic Diacritics

The Arabic script consists of two classes of symbols: letters and diacritics. Letters comprise long vowels such as 'A', 'y', 'w' as well as consonants.[1] Diacritics on the other hand comprise short vowels, gemination markers, nunation markers, as well as other markers (such as hamza, the glottal stop which appears in conjunction with a small number of letters, elongation, dots on letters, and emphatic markers). If present, these diacritics marks help to render a more precise reading of a given word in context as observed in the ARET project (Maamouri et al., 2012). In this experiment, we are mostly addressing three types of diacritical marks: short vowels, nunation (marker for indefiniteness), and shadda (gemination).

The available Arabic text content has some percentage of these diacritics present depending on domain and genre. For instance, religious text such as the Quran is fully diacritized to minimize chances of reciting it incorrectly as discussed in (Atwell et al., 2010). The same finding applies in most children educational texts and classical poetry. However, the majority of news text and variety of other genres are sparsely diacritized: For example, around 1.5% of tokens in the United Nations Arabic corpus bear at least one diacritic (Diab et al., 2007).

2.2 Annotation Ambiguity

In general, there are several reasons that may cause disagreement in annotation decisions including human errors, lack of precision in the guidelines, and the lack of expertise and training of the annotators. This disagreement rate further increases due to the inherent natural ambiguity in the human language itself where various interpretations for a word are possible. Such linguistic ambiguity has been reported in many annotation projects involving various linguistic phenomenon, such as the coreference relations, the predicate-argument structure, the semantic roles and the L2 language errors (Versley and Tbingen, 2006; Iida et al., 2007), prosodic breaks (Jung and Kwon, 2011; Ruppenhofer et al., 2013; Rosen et al., 2013), as well as the various Arabic PropBank projects (Diab et al., 2008; Zaghouani et al., 2010; Zaghouani et al., 2012) and the Arabic TreeBank (Maamouri et al., 2010).

Poesio and Artstein (2005) classify ambiguity into explicit and implicit types. The explicit ambiguity refers to the individuals' understanding of the annotation task. On the other hand, implicit ambiguity refers to those revealed after observing and contrasting the annotation done in the same task by other annotators. Annotators are generally asked to detect and resolve ambiguous cases, which can be a difficult task to accomplish. This leads to a lower inter-annotator agreement in such tasks.

2.3 Annotation Complexity

There are many studies that evaluate the language complexity in addition to the quality of manual annotation and also allow the identification of many factors causing lower inter-annotator agreements. For example, Bayerl and Paul (2011) showed that there is a correlation between the inter-annotator agreement and the complexity of the annotation task; for instance, the larger the number of categories is, the lower the inter-annotator agreement is. Moreover, the categories prone to confusions are generally limited. This brings out two complexity issues related to the number of categories and to the existence of ambiguity between some the categories as explained in (Popescu-Belis, 2007). Furthermore, there are some annotation tasks for which the choice of a label is entirely left to the annotator, which can lead to even more complexity and lower agreement. In our project, the annotators frequently encounter complex linguistic issues such as ambiguity and the multiple possible and acceptable solutions including the free edit mode. In the next sections, we present these issues in detail.

3 Annotation Framework

The annotation pipeline in large annotation projects requires the involvement of many dedicated parties. In our project, the annotation is led by a lead annotator with a team of four native Arabic-speaking annotators from three Arab countries (Egypt, Palestine, and Tunisia) and a programmer. All the annotators

[1] Arabic transliteration is presented in the Buckwalter scheme (Buckwalter, 2002)

hold at least a university-level degree and they have a good knowledge of the Arabic language. The lead annotator is responsible for the entire annotation pipeline including the corpus compilation, the annotation of the gold-standard evaluation files, the guidelines, the ongoing training of the annotators, and the evaluation of the annotation quality throughout the lifespan of the project.

3.1 Guidelines

Before starting the task, we provided the annotators with detailed guidelines, describing our diacritization scheme and specifying when and where to add the diacritics required. We describe the annotation procedure and explain how to deal with borderline cases. We also include several annotated examples to illustrate the specified rules. We provide some examples of each case including the diacritization exceptions and some specific rules for: the Shadda gemination mark, the Soukoun (absence of a vowel) and the Nunation marks at the end of a word. Moreover, in some cases, the letters followed by a long Alif letter ا, should not be diacritized as it is considered a deterministic diacritization as in مِيثاقُ /miy**v**Aqu/ 'Treaty' and not مِيثَاقُ /miy**va**Aqu/.[2] A summary of the most common Arabic diacritization rules is also added as a reference in the guidelines.

3.2 Annotation Tool

We designed and implemented MANDIAC, a web-based annotation tool and a work-flow management interface (Obeid et al., 2016), the tool is based on QAWI (Obeid et al., 2013) a token-based editor, used to annotate and correct spelling errors in Arabic text for the Qatar Arabic Language Bank (QALB) project.[3] The basic interface of the annotation tool is shown in Figure 1, apart from the surface controls, the interface allows annotators to select from an automatically generated diacritized words list and/or edit words manually as shown. The annotation interface allows users to undo/redo actions, and the history is kept over multiple sessions. The interface includes a timer to keep track of how long each sentence annotation has taken. We used the timer feature to measure the annotation speed later on during the evaluation experiments.

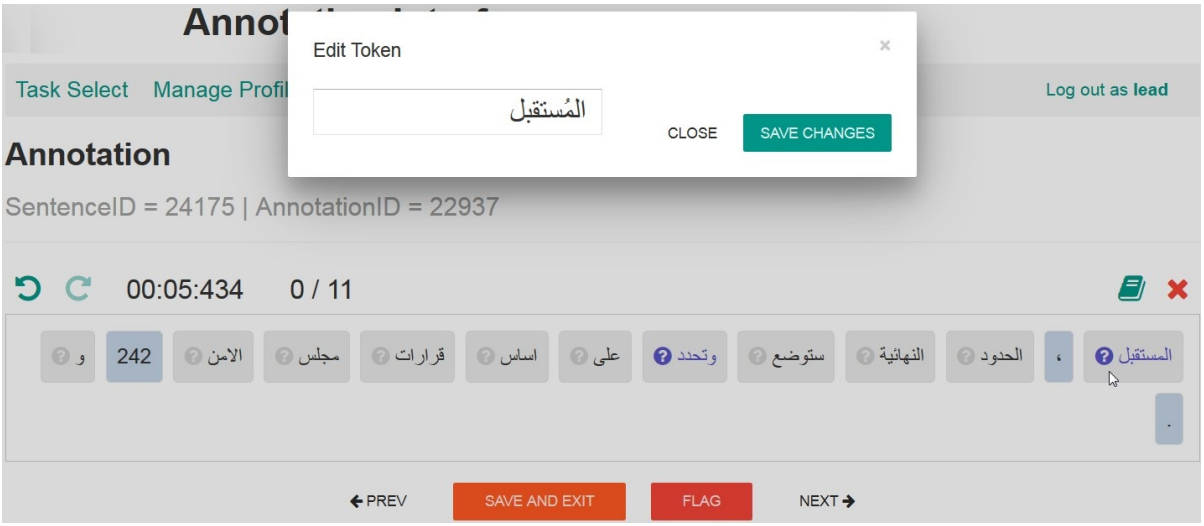

Figure 1: Editing a word marked as possibly ambiguous

[2]In this case the short vowel /a/ following the letter ث /v/ should not be added as specified in the Arabic diacritization guidelines.

[3]The Qatar Arabic Language Bank (QALB) project is large-scale manually annotated Arabic text correction project (Zaghouani et al., 2014; Zaghouani et al., 2015; Zaghouani et al., 2016b; Mohit et al., 2014; Rozovskaya et al., 2015).

4 Experimental setup

4.1 Evaluation sets

We use the corpus of contemporary Arabic (CCA) compiled by Al-Sulaiti and Atwell (2006). It is a balanced corpus divided into the following genres: autobiography, short stories, children's stories, economics, education, health and medicine, interviews, politics, recipes, religion, sociology, science, sports, tourism and travel. The CCA corpus text genres were carefully selected by its compilers since the target users of the corpus were mostly language teachers and teachers of Arabic as a foreign language. Various metadata information are included in the corpus such as the information about the text, the author and the source. In order to use the CCA corpus, a normalization effort was done to produce a consistent XML mark-up format to be used in our annotation tool. Furthermore, we split paragraphs and sentences by period and remove repeated sentences after the initial segmentation in order to start the annotation process.

4.2 Annotation Process

The annotation consists of a single annotation pass as commonly done in many annotation projects due to time and budget constraints (Rozovskaya and Roth, 2010; Nagata et al., 2006; Izumi et al., 2004; Gamon et al.,). While performing the annotation task, the annotators do not need to add the diacritics for each word, instead, we use MADAMIRA (Pasha et al., 2014), a system for morphological analysis and disambiguation of Arabic, to provide automatically diacritized candidates. Therefore, the annotators are asked to choose the correct choice from the top three candidates suggested by MADAMIRA, when possible, if it appears in the list. MADAMIRA is able to achieve a lemmatization accuracy of 96.0% and a diacritization accuracy of 86.3%. Otherwise, if they are not satisfied with the given candidates, they can manually edit the word and add the correct diacritics. We hypothesize that such integration of an automatic analyzer in the annotation process will lead to a much faster annotation than purely manual annotation, provided that the preassigned tags are sufficiently accurate.

4.3 Ambiguity Detection

In order to classify each word as ambiguous or not, we apply several preprocessing and filtering steps on the datasets. We run MADAMIRA on the datasets to provide us with all possible morphological analyses associated with confidence scores for each word in context. MADAMIRA applies SVM and language models to derive predictions for the words morphological features and then scores each words analysis list based on how well each analysis agrees with the model predictions. The top scoring analysis is MADAMIRA's most probable reading of the word in context. We hypothesized that ambiguous words in context would have other competing high-scoring analyses within a threshold difference from the top scoring one. Based on a previous experience, we chose the threshold to be 15%, therefore, we keep the top scoring analysis and all other analyses that are within 15% difference from the top one. We further reduce this list to remove redundant and insignificant variants based on certain criteria. We remove case and mood diacritic marks, which encode inflectional properties. Additionally, we remove the diacritics of the third possessive pronouns because its diacritic marks are highly affected by the case and mood marks that we attempt to neutralize. Additionally, we filter out nouns that are exactly the same but differ only in the letter Alef normalization (ا أ إ آ |, >, <, { and A) (e.g الاِنْتِخابِيَة Al<inotixAbiy∼ap and الاِنْتِخابِيَة AlAinotixAbiy∼ap 'The electoral'); thus, if we have two instances differ only in Alef normalization we only keep one of them. We also remove the addition of gemination sign known as shaddah (∼) to the Sun letters to assimilate the letter Lam (ل l) of a preceding definite article 'Al' in nouns (e.g. النازِيّ AlnAziy∼ and النّازِيّ Aln∼Aziy∼ 'The Nazi' and also in الرَغْبَة Alragobap and الرَّغْبَة Alr∼agobap 'The desire'). The above filtering process is performed because it decreases the possible analyses but they do not have an impact in detecting the lexical ambiguity which is our goal. We finally make sure that the remaining analyses are unique because we may end up with repeated words after removing specific diacritics marks; additionally, words that are the same orthographically but differ in other features such as lemma and part of speech tags are also removed.

If the resulting list of possible analyses contain more than one possibility, the word is marked as ambiguous; otherwise, it is believed to be not ambiguous. Words that have no analysis generated using MADAMIRA are also considered ambiguous. For each sentence, we count the number of words that are marked as ambiguous using our approach, and then calculate the percentage of ambiguity. We sort the sentences according to their ambiguity percentages in descending order so that we give annotators ranked sentences for annotation. Because we are concerned with MSA dataset only, we further filter out dialectal sentences using AIDA (Elfardy and Diab, 2012), a tool that classifies words and sentences as MSA (formal Arabic) or DA (Dialectal Arabic).

5 Evaluation

For the evaluation, we used a sample of 10K-Words from the CCA corpus representing 4 domains with approximately 2.5K-words per domain (children stories, economics,sports and politics). We have three experimental conditions for three evaluations carried over a period of six weeks.

1. **The first condition (COND1):** In the first experimental condition (COND1), four annotators were given raw undiacritized sentences and were asked to add the missing diacritics as per the guidelines. They either select one of the top three diacritization choices computed by MADAMIRA or manually edit the word.

2. **The second condition (COND2):** In the second experimental condition (COND2), we provided the raw undiacritized sentences to a first group of two annotators (Group 1) and we asked them to mark and add the required diacritics only to the words they believe are ambiguous while ignoring the rest of the non ambiguous words in the sentence.

3. **The third condition (COND3):** For the third experimental condition (COND3), we gave, to a different group of two annotators (Group 2), the same sentences assigned to Group 1 while having the sentences explicitly marked as potentially ambiguous using the MADAMIRA as explained previously (again the top three MADAMIRA choices were provided). Furthermore, in COND3, the annotators were asked to tell whether they agree or not with the ambiguity class provided for each word using the tool and also by adding the missing diacritics in case they agree that the given word is ambiguous.

The Inter-Annotator Agreement (IAA) is measured by using pairwise percent agreement averaged over all pairs of annotations (APP). The pairwise percent agreement (also called observed agreement) is computed as the percentage of times two annotators assign the same label to a unit. If a single letter in a given word has one diacritization mismatch, then the whole word is considered as disagreement. A high APP score denotes that at least two annotators agree on the annotation and therefore, the probability that the annotation is erroneous is very small.

	CCA Corpus
APP$_{COND1}$	83.10%
APP$_{COND2}$	69.09%
APP$_{COND3}$	**88.31%**

Table 1: Inter-Annotator Agreement (IAA) in terms of Average Pairwise Percent agreement (APP) recorded during the evaluation of 10K-words from the CCA dataset in three experimental conditions; higher is better.

Furthermore, in order to measure the impact on the annotation speed, we measured the mean annotation time by computing the average time required to annotate a word for a sentence and then average it over all sentences for a given experimental condition by all the annotator. The Average annotation speeds are shown in Table 2.

	Annotation Speed
Words / Minute$_{COND1}$	8.22
Words / Minute$_{COND2}$	6.59
Words / Minute$_{COND3}$	**10.09**

Table 2: Average annotated words per minute recorded during the evaluation of 10K-words from the CCA dataset in three experimental conditions

The results obtained in Table 1 and Table 2 show that in COND1 the annotators obtained a fairly good agreement of 83.10% and average speed of 8.22 words/minute ranking in the second place in terms of performance overall. COND2 obtained surprisingly has the lowest agreement of only 69.09% and also lowest time performance of only 6.59 words / minute. A follow up with the annotators revealed that the results of COND2 are due to the fact that annotators spent a lot of time thinking whether a given word is ambiguous or not so they can add the required diacritics. This leads to spending more time due to the hesitation in addition to the difficulty of the task as we will show in the next section. Finally, COND3 reveals the best overall performance with a high agreement of 88.31% and the highest rate of words per minute of 10.09. The results of COND3 can be explained by the automatic ambiguity analysis provided to the annotators which substantially reduced the hesitation in deciding if a given word is ambiguous and therefore it reduced the annotation possibilities by assisting the annotators in their decisions.

6 Error analysis

We found that a large number of the agreement errors are due to the inherent linguistic complexity of the Arabic language leading to some annotation hesitations and inconsistency between the annotators when there is an obvious ambiguity in the context. For instance, in many cases the annotators did not agree on whether to add the diacritics or not, while in other cases, the annotators disagreed on the interpretation of the word. We compiled below the list of the most important cases of disagreement observed during the error analysis.

1. **Lexical Ambiguity:** This means that a word could carry more than one acceptable reading (homonymy) such as in the case of the word (قبل qbl which has the following two lexical readings a. قَبْلَ qabola ‘before’ and b. قِبَلَ /qibala/ ‘capability’.

2. **Morphological Ambiguity:** For this category, we observed two types of annotation disagreement: word-structure ambiguity and inflection ambiguity. The diacritization of word structure can be interpreted as a morphological task. As in the diacritization of the second letter of trilateral verbs such as in حَسِبَ يَحْسَبُ Hasiba/yaHosabu ‘To think’ versus حَسَبَ يَحْسِبُ /Hasaba/yaHosibu/ ‘To count’. Since the Arabic language is a morphologically rich language, each inflected word could have a different way to be diacritized, especially in cases where some pronouns are attached to the verbs or the nouns as in أَحْسَنا /AaHosanA/ ‘they help/do good’ (3rd,Dual,Masculine) versus in أَحْسَنّا /AaHosan∼A/ ‘we help/do good’ (1st,Plural). In another disagreement case, we found some cases of verbal voice inflection confusion between the active voice and the passive voice such as in تَعُدُّ /taEud∼u ‘she counts’ versus تُعَدُّ /tuEid∼u/ ‘It is considered’.

3. **Part of Speech Ambiguity (POS):** This is one of the most frequent disagreement cases found during the error analysis, in fact, it is common to have many possible POS for a given word in Arabic depending on the personal interpretation of the sentence as in the case of the verb نُجِيبَ /nujiybu/ ‘we+answer’ versus the adjective نَجِيبَ /najiybN/ ‘outstanding’.

4. **Case Endings Ambiguity:**

In Arabic, the case endings are those attached to the ends of words to indicate the words' grammatical function. Using the case endings correctly, requires a solid knowledge of grammar. With no surprise, we found many annotation disagreement in this category. For example, the genitive ثَلاثٍ /valAvK/ 'three' was confused with the nominative ثَلاثٌ /valAvN/ 'three'.

5. **The indeclinable nouns (Diptote):** Indeclinable nouns are a type of nouns that have special case endings rules and they only have two possible case endings. When the noun is indefinite, the possible case endings are /-a/ for the genitive and /-u/ for the nominative, while the accusative has no nunation. We located several cases of Diptote errors when the noun is indefinite such as in: أَشِقَّاءَ /¿a$iq∼A'a/ 'brothers' (genitive) versus أَشِقّاءٍ 'brothers' with a wrong genitive nunation marker.

6. **Phonology ambiguity:** As the diacritization is considered an orthographic representation of the phonological phenomena, some phonological cases depend on the phonological context and some changes could happen as a result of an assimilation phenomena. For example, we noticed several cases of disagreement related to the definite article Al as it could be pronounced in two ways: the first way is known as the sun letter /Al/ where the letter /l/ is silent and a gemination diacritic sign is marked on the following letter. The second case is the moon letter /Al/, where the letter /l/ is pronounced as in the example of الدَّهِشَةُ /Al**l**d∼h$apu/ 'The+surprise' versus الدَهْشَةُ /Al**d**ah$apu/ 'The+surprise'.

7. **Pragmatic variations:** In this type of disagreement, the annotators were confused between two possible and acceptable ways to pronounce a given word and the difference is only dictated by the regional usage as the case of the word دولي /dwly/ 'international' which could be diacriticized as /d**aw**oliy∼/ دَوْلي or as /d**uw**aliy∼/ دُوَلِيّ.

8. **Level of Diacritization:** We observed that frequently, the annotators did not agree on the level of diacritization to be added despite the existence of guidelines. Cases of disagreement like the following are frequently observed: لِلمُلْتَزَماتِ liAlmul**o**taz**i**mAt 'for+the+committed'(PL+Fem.) versus لِلمُلتَزماتِ /liAlmu**lta**zmAt/ 'for+the+committed'(PL+Fem.).

9. **Diacritization Typos:** While not frequent, several cases of extra diacritics marks were added accidentally by the annotators as in قَصَّرَنا /q**a**S∼aronA/ 'We+abridged' versus the wrong extra diacritic a in قَصَّرْنا /qa**a**S∼aronA/.

Conclusion

In this paper, we present our method to detect the ambiguous annotation cases within a Diacritization annotation project. We discussed the complex linguistic challenges inherent in Arabic linguistic annotation. The results obtained in the evaluation suggest that the automatic ambiguity detection could effectively reduce the annotation time and also increase the Inter-annotator agreement. Moreover, we believe that the higher the accuracy of MADAMIRA choices, the faster the annotation could be as as manual edits will be reduced. However, we believe that the nature of the ambiguity of the Arabic language as attested by many disagreement cases, has strongly impacted the overall agreement results. On the other hand, we believe that a better agreement could be achieved if the annotators followed the annotation guidelines consistently.

Acknowledgments

This publication is made possible by grant NPRP-6-1020-1-199 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

Gheith A Abandah, Alex Graves, Balkees Al-Shagoor, Alaa Arabiyat, Fuad Jamour, and Majid Al-Taee. 2015. Automatic diacritization of arabic text using recurrent neural networks. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):183–197.

Latifa Al-Sulaiti and Eric Steven Atwell. 2006. The design of a corpus of contemporary arabic. *International Journal of Corpus Linguistics*, 11(2):135–171.

Eric Atwell, Nizar Habash, Bill Louw, Bayan Abu Shawar, Tony McEnery, Wajdi Zaghouani, and Mahmoud El-Haj. 2010. Understanding the quran: A new grand challenge for computer science and artificial intelligence. *ACM-BCS Visions of Computer Science 2010*.

Petra Saskia Bayerl and Karsten Ingmar Paul. 2011. What determines inter-coder agreement in manual annotations? a meta-analytic investigation. *Computational Linguistics*, 37(4):699–725.

Yonatan Belinkov and James Glass. 2015. Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285, Lisbon, Portugal.

Houda Bouamor, Wajdi Zaghouani, Mona Diab, Ossama Obeid, Kemal Oflazer, Mahmoud Ghoneim, and Abdelati Hawwari. 2015. A pilot study on arabic multi-genre corpus diacritization. In *Proceedings of the Association for Computational Linguistics Second Workshop on Arabic Natural Language Processing*, pages 80–88, Beijing, China.

Tim Buckwalter. 2002. Buckwalter Arabic morphological analyzer version 1.0. Technical Report LDC2002L49, Linguistic Data Consortium.

Mona Diab, Mahmoud Ghoneim, and Nizar Habash. 2007. Arabic diacritization in the context of statistical machine translation. In *Proceedings of Machine Translation Summit (MT-Summit)*, Copenhagen, Denmark.

Mona Diab, Aous Mansouri, Martha Palmer, Olga Babko-Malaya, Wajdi Zaghouani, Ann Bies, and Mohammed Maamouri. 2008. A pilot arabic propbank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*.

Heba Elfardy and Mona Diab. 2012. Aida: Automatic identification and glossing of dialectal arabic. In *Proceedings of the 16th eamt conference (project papers)*, pages 83–83.

Michael Gamon, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. In *Third International Joint Conference on Natural Language Processing, IJCNLP, Address=Hyderabad, India, Year=2008, Pages = 449–456, Title = Using Contextual Speller Techniques and Language Modeling for ESL Error Correction*.

Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. 2007. Annotating a japanese text corpus with predicate-argument and coreference relations. In *Proceedings of the Linguistic Annotation Workshop*, LAW '07, pages 132–139, Stroudsburg, PA, USA. Association for Computational Linguistics.

Emi Izumi, Kiyotaka Uchimoto, and Hitoshi Isahara. 2004. The NICT JLE corpus exploiting the language learners' speech database for research and education. *International Journal of The Computer, the Internet and Management*, 12(2):119–125, May.

Youngim Jung and Hyuk-Chul Kwon. 2011. Consistency maintenance in prosodic labeling for reliable prediction of prosodic breaks. In *Proceedings of the 5th Linguistic Annotation Workshop*, LAW V '11, pages 38–46, Stroudsburg, PA, USA. Association for Computational Linguistics.

Katrin Kirchhoff and Dimitra Vergyri. 2005. Cross-dialectal data sharing for acoustic modeling in arabic speech recognition. *Speech Communication*, 46(1):37–51.

Mohamed Maamouri, Ann Bies, Seth Kulick, Wajdi Zaghouani, David Graff, and Michael Ciul. 2010. From speech to trees: Applying treebank annotation to arabic broadcast news. In *Proceedings of International Conference on Language Resources and Evaluation (LREC 2010)*.

Mohamed Maamouri, Wajdi Zaghouani, Violetta Cavalli-Sforza, Dave Graff, and Mike Ciul. 2012. Developing aret: an nlp-based educational tool set for arabic reading enhancement. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 127–135. Association for Computational Linguistics.

Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghouani, and Ossama Obeid. 2014. The first qalb shared task on automatic text correction for arabic. In *Proceedings of the EMNLP Workshop on Arabic Natural Language Processing*, page 39.

Ryo Nagata, Atsuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. A feedback-augmented method for detecting errors in the writing of learners of english. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 241–248, Sydney, Australia.

Ossama Obeid, Wajdi Zaghouani, Behrang Mohit, Nizar Habash, Kemal Oflazer, and Nadi Tomeh. 2013. A web-based annotation framework for large-scale text correction. In *Sixth International Joint Conference on Natural Language Processing*, page 1.

Ossama Obeid, , Houda Bouamor, Wajdi Zaghouani, Mahmoud Ghoneim, Abdelati Hawwari, Mona Diab, and Kemal Oflazer. 2016. Mandiac: A web-based annotation system for manual arabic diacritization. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC-2016) Workshop on Free/Open-Source Arabic Corpora and Corpora Processing Tools (OSACT2)*.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Massimo Poesio and Ron Artstein. 2005. Annotating (anaphoric) ambiguity. In *In Proceedings of the Corpus Linguistics Conference*.

Popescu-Belis, 2007. *Le role des metriques d'evaluation dans le processus de recherche en TAL*, pages 67–91. Traitement Automatique de la Langue, vol. 48, n. 1.

Alexandr Rosen, Jirka Hana, Barbora Štindlová, and Anna Feldman. 2013. Evaluating and automating the annotation of a learner corpus. *Language Resources and Evaluation*, pages 1–28, April.

Alla Rozovskaya and Dan Roth. 2010. Annotating esl errors: Challenges and rewards. In *NAACL Workshop on Innovative Use of NLP for Building Educational Applications*, Los Angeles, CA.

Alla Rozovskaya, Houda Bouamor, Nizar Habash, Wajdi Zaghouani, Ossama Obeid, and Behrang Mohit. 2015. The second qalb shared task on automatic text correction for arabic. In *Proceedings of the ACL-IJCNLP Workshop on Arabic Natural Language Processing*, page 26.

Josef Ruppenhofer, Russell Lee-Goldman, Caroline Sporleder, and Roser Morante. 2013. Beyond sentence-level semantic role labeling: linking argument structures in discourse. *Language Resources and Evaluation*, 47(3):695–721.

Eckehard Schulz. 2004. *A Student Grammar of Modern Standard Arabic*. Cambridge University Press, Cambridge, United Kingdom.

Anas Shahrour, Salam Khalifa, and Nizar Habash. 2015. Improving arabic diacritization through syntactic analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1309–1315, Lisbon, Portugal.

Yannick Versley and Universitt Tbingen. 2006. Disagreement dissected: vagueness as a source of ambiguity in nominal (co-) reference. In *In: Ambiguity in Anaphora Workshop Proceedings*, pages 83–89.

Wajdi Zaghouani, Mona Diab, Aous Mansouri, Sameer Pradhan, and Martha Palmer. 2010. The revised arabic propbank. In *Proceedings of the Association for Computational Linguistics Fourth Linguistic Annotation Workshop*, pages 222–226. Association for Computational Linguistics.

Wajdi Zaghouani, Abdelati Hawwari, and Mona Diab. 2012. A pilot propbank annotation for quranic arabic. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature co-located with the North American Association Computational Linguistics conference (NAACL-HLT 2012*, page 78.

Wajdi Zaghouani, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2014. Large scale arabic error annotation: Guidelines and framework. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26-31, 2014.*, pages 2362–2369.

Wajdi Zaghouani, Nizar Habash, Houda Bouamor, Alla Rozovskaya, Behrang Mohit, Abeer Heider, and Kemal Oflazer. 2015. Correction annotation for non-native arabic texts: Guidelines and corpus. In *Proceedings of the Association for Computational Linguistics Fourth Linguistic Annotation Workshop*, pages 129–139.

Wajdi Zaghouani, Houda Bouamor, Abdelati Hawwari, Mona Diab, Ossama Obeid, Mahmoud Ghoneim, Sawsan Alqahtani, and Kemal Oflazer. 2016a. Guidelines and framework for a large scale arabic diacritized corpus. In *The Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3637–3643. European Language Resources Association (ELRA).

Wajdi Zaghouani, Nizar Habash, Ossama Obeid, Behrang Mohit, Houda Bouamor, and Kemal Oflazer. 2016b. Building an arabic machine translation post-edited corpus: Guidelines and annotation. In *International Conference on Language Resources and Evaluation (LREC 2016)*.

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.