# MADARi: A Web Interface for Joint Arabic Morphological Annotation and Spelling Correction

**Ossama Obeid, Salam Khalifa, Nizar Habash,**
**Houda Bouamor,† Wajdi Zaghouani,* Kemal Oflazer†**
Computational Approaches to Modeling Language Lab, New York University Abu Dhabi, UAE
† Carnegie Mellon University in Qatar, Qatar
* Hamad Bin Khalifa University, Qatar
{oobeid, salamkhalifa, nizar.habash}@nyu.edu,
hbouamor@qatar.cmu.edu, wzaghouani@hbku.edu.qa, ko@cs.cmu.edu

**Abstract**

In this paper, we introduce MADARi, a joint morphological annotation and spelling correction system for texts in Standard and Dialectal Arabic. The MADARi framework provides intuitive interfaces for annotating text and managing the annotation process of a large number of sizable documents. Morphological annotation includes indicating, for a word, in context, its baseword, clitics, part-of-speech, lemma, gloss, and dialect identification. MADARi has a suite of utilities to help with annotator productivity. For example, annotators are provided with pre-computed analyses to assist them in their task and reduce the amount of work needed to complete it. MADARi also allows annotators to query a morphological analyzer for a list of possible analyses in multiple dialects or look up previously submitted analyses. The MADARi management interface enables a lead annotator to easily manage and organize the whole annotation process remotely and concurrently. We describe the motivation, design and implementation of this interface; and we present details from a user study working with this system.

**Keywords:** Arabic, Morphology, Spelling Correction, Annotation

## 1. Introduction

Annotated corpora are vital for research in natural language processing (NLP). These resources provide the necessary training and evaluation data to build automatic annotation systems, and benchmark them. The task of human manual annotation, however, is rather difficult and tedious and several annotation interface tools have been created to assist in such effort. These tools tend to be specialized to optimize for specific tasks such as spelling correction, part-of-speech (POS) tagging, named-entity tagging, syntactic annotation, etc. Certain languages bring additional challenges to the annotation task. Compared with English, Arabic annotation introduces a need for diacritization of the diacritic-optional orthography, frequent clitic segmentation, and a richer POS tagset.

In this paper, we focus on designing and implementing a tool targeting Arabic dialect morphological annotation. Standard Arabic morphology is quite rich (Habash, 2010), but Arabic dialects introduce more complexity than Standard Arabic in that the input text has noisy orthography. For example, the word ويابوهاالخليج *wyAbwhAAlxlyj*[1] involves two spelling errors[2] (a word merge and character replacement) which can be corrected as وجابوها الخليج *wjAbwhA Alxlyj* 'and they brought it to the Gulf'. Furthermore, the first of the two corrected words includes two clitics that when segmented produce the form: و+ جابوا +ها *w+ jAbwA +hA* 'and+ they-brought +it'.

[1]Transliterations are in the Habash-Soudi-Buckwalter scheme (Habash et al., 2007).

[2]Since Arabic dialects do not have a standard orthography, *spelling correction* here means to conventionalize as per the CODA standard (Habash et al., 2018).

Previous work on Arabic morphology annotation interfaces focused either on the problem of manual annotations for POS tagging (Maamouri et al., 2014), or diacritization (Obeid et al., 2016), or spelling correction (Obeid et al., 2013). In this paper we present a tool that allows doing all of these tasks together, eliminating the possibility of error propagation from one annotation level to another. Our tool is named MADARi[3] after the project under which it was created: Multi-Arabic Dialect Annotations and Resources (Bouamor et al., 2018).

The remainder of this paper is structured as follows: we present work related to this effort in Section 2. In Section 3., we discuss the design and architecture of our annotation framework. In Section 4. and Section 5., we discuss the annotation and management interfaces, respectively. We finally describe a user study of working with MADARi in Section 6.

## 2. Related Work

Several annotation tools and interfaces were proposed for many languages and to achieve various annotation tasks. Some are general purpose annotation tools, such as BRAT (Stenetorp et al., 2012) and WebAnno (Yimam et al., 2013). Task-specific annotation tools for post-editing and error correction include the work of Aziz et al. (2012), Stymne (2011), Llitjós and Carbonell (2004), and Dickinson and Ledbetter (2012).

For Arabic, there are several existing annotation tools, however, they are designed to handle specific NLP tasks and are not easy to adapt to our project. Examples include tools for semantic annotation such as the work of Saleh and Al-Khalifa (2009) and El-ghobashy et al. (2014),

[3]مداري *madAriy* means 'my orbit' in Arabic.

and the work on dialect annotation by Benajiba and Diab (2010) and Diab et al. (2010). Attia et al. (2009) built a morphological annotation tool. Recently, Al-Twairesh et al. (2016) introduced MADAD, a general-purpose online collaborative annotation tool for readability assessments project in Arabic. In the COLABA initiative (Diab et al., 2010), the authors built tools and resources to process Arabic social media data such as blogs, discussion forums, and chats. Javed et al. (2018) presented an online interface for joint syntactic annotation and morphological tokenization for Arabic.

In general, many of these existing tools are not designed to handle the peculiarities of dialectal Arabic. They neither provide facilities for managing thousands of documents nor permit the distribution of tasks to tens of annotators, including managing inter-annotator agreement (IAA) tasks. Our interface borrows ideas from three other existing annotation tools: DIWAN, QAWI, and MANDIAC. Here we describe each of these tools and how they have influenced the design of our system.

**DIWAN** is an annotation tool for Arabic dialectal texts (Al-Shargi and Rambow, 2015). It provides annotators with a set of tools for reducing duplicate effort including the use of morphological analyzers to pre-compute analyses, and the ability to apply analyses to multiple occurrences simultaneously. However, it requires installation on a Windows machine and the user interface is not very friendly to newcomers.

**QAWI** (the QALB Annotation Web Interface) was used for token-based text editing to create raw and and text corrected parallel data for automatic text correction tasks (Obeid et al., 2013; Zaghouani et al., 2014; Zaghouani et al., 2015; Zaghouani et al., 2016). It supported the exact recording of all modifications performed by the annotator which previous tools did not. We utilize this token-based editing system for minor text corrections that transform text of a given dialect into the appropriate CODA orthography (Habash et al., 2018).

**MANDIAC** utilized the token-based editor used in QAWI to perform text diacritization tasks (Obeid et al., 2016). More importantly, it introduced a flexible hybrid data storage system that allows for adding new features to the annotation front-end with little to no modifications to the back-end. MADARi utilizes this design to provide the same utility.

## 3. MADARi Design

The MADARi interface is designed to be used by human annotators to create a morphologically annotated corpus of Arabic text. The text we work with comes from social media and is highly dialectal (Bouamor et al., 2018; Khalifa et al., 2018) and has numerous spelling errors. The annotators will carefully correct the spelling of the words and also annotate their morphology. The in-context morphology annotation includes tokenization, POS tagging, lemmatization and English glossing.

### 3.1. Desiderata

In order to manage and process the annotation of the large scale dialectal Arabic corpus, we needed to create a tool to streamline the annotation process. The desiderata for developing the MADARi annotation tool include the following:

- The tool must have very minimal requirements on the annotators.
- The tool must allow off-site data management of documents to allow annotation leaders to assign and grade documents from anywhere in the world and to allow hiring annotators anywhere in the world.
- The tool must allow easily customizable POS tag sets by annotation leads.
- The tool must allow easy access to other user annotations of similar texts.
- The tool must allow for easy navigation between spelling changes and morphological disambiguation.

### 3.2. Design and Architecture

The design of our interface borrows heavily from the design of MANDIAC (Obeid et al., 2016). In particular, we utilized the client-server architecture, as well as the flexible hybrid SQL/JSON storage system used by MANDIAC. This allows us to easily extend our annotation interface with minor changes, if any, to the back-end. Our system stores documents one sentence per row, unlike MANDIAC which stores one document per row. This modification allows the annotation interface to handle larger file sizes without affecting its performance by only overwriting the JSON of the modified sentences and not that of the entire document. Like, DIWAN and MANDIAC, we also utilize MADAMIRA (Pasha et al., 2014), a morphological analyzer and disambiguator for Arabic to pre-compute analyses.

## 4. Annotation Interface

The annotation interface (illustrated in Figures 1 to 4) is where annotators perform the annotation tasks assigned to them. Here we describe the different components and utilities provided this interface.

### 4.1. Task Overview

When starting an annotation session, annotators are first shown the "Task Overview" screen (Figure 1). Here annotators can see information on the size of the current task and their progress so far (Figure 1a). The sentence list can be filtered to contain sentences matching a desired search term using the filter bar (Figure 1b). The list of sentences in the current task is also displayed with validated tokens color-coded green (Figure 1c). Clicking on any word in the sentence list will open the annotation interface (Figure 2) at that word.

### 4.2. Word Analysis

The essential component of our interface is the morphological analysis screen (Figure 2). The original text is provided for reference at the top of the panel (Figure 2a). Figure 2b displays the updated form of the words, and allows selecting a word to annotate. The currently selected word is colored blue; and validated words are colored green.

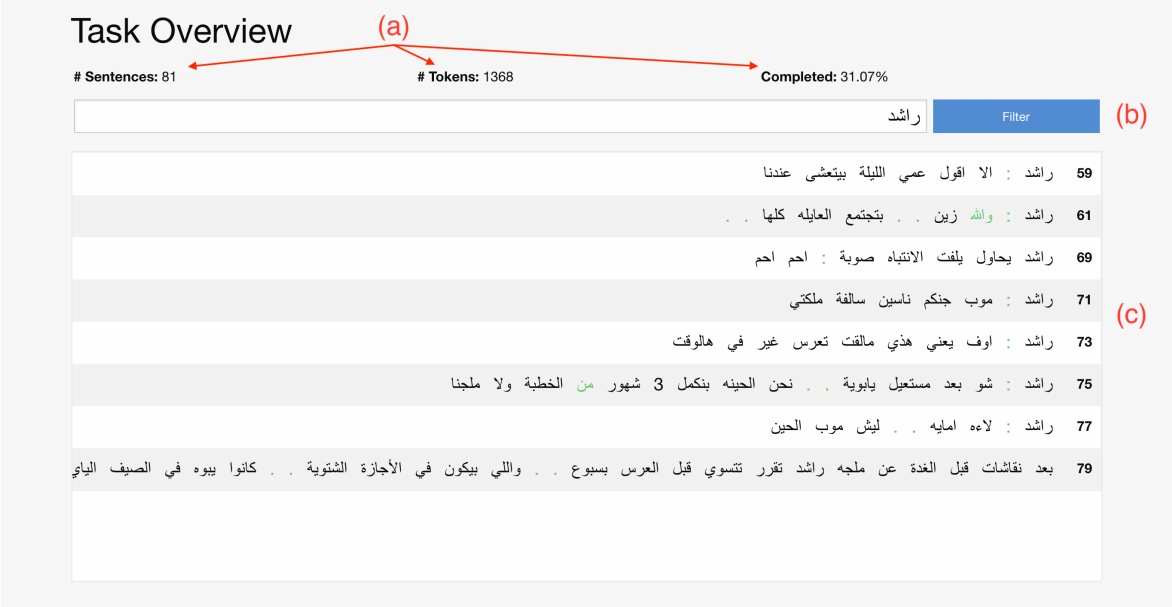



Figure 1: MADARi Task Overview screen

Figure 2c is the heart of the annotation interface, where annotators manually disambiguate the morphological analysis. Disambiguation includes morphologically tokenizing each word into proclitics, baseword, and enclitics (Figure 2c, first row from right to left, respectively). Each of these are assigned a POS tag as well as a morphological feature[4] where applicable (Figure 2c, second row). Annotators also assign the lemma, gloss, and dialect for each word (Figure 2c, third row, second to fourth fields from the left respectively). For the convenience of the annotators, we provide pre-computed values for each field using MADAMIRA's morphological analysis. Each word has a validated field (Figure 2c, third row, right-most field) to indicate that the annotator has fully analyzed it and is confident with their analysis.

Generally, the final form of a word is a concatenation of the proclitics, baseword, and enclitics. However, there are certain cases where that is not true because some orthographic rewrite rules must apply (Habash et al., 2018). Using the example in the introduction, وجابوها *wjAbwhA* should be tokenized by annotators into و+ جابوا +ها *w+ jAbwA +hA*. However, when displaying the detokenized token, the system should show وجابوها *wjAbwhA* and not وجابواها *wjAbwAhA*. MADARi has built-in rewrite rules for trivial detokenization cases but we also allow annotators to manually edit the detokenized form manually as needed (Figure 2c, third row, left-most field).

### 4.3. Text Editing

Annotators can freely alternate between morphological analysis and spelling modification of the words in the sentence. This gives them the freedom to make joint decisions on spelling and morphology and avoid error propagation. Sentence edits can be made by going to the "Edit Sentence" view (Figure 3). In the "Edit Sentence" view, only the word tokens of the sentence are shown, each with a left and right arrow button surrounding them (Figure 3a). Clicking on one of these arrows merges that token with the one on the left or right respectively. Double clicking on a token displays the "Edit Token" pop-up (Figure 3a). In this pop-up, an annotator can edit a word or split it into multiple tokens by inserting spaces between the letters.

### 4.4. Utilities

We have added a number of utility features to make the annotation process easier and more efficient for annotators. Basic utilities include undo and redo buttons (Figure 2h), switching between English and Arabic POS tags (Figure 2f). Annotators can jump to the next or previous sentence, go to the "Task Overview" screen, or exit the task in the navigation bar (Figure 2e). All functions in the navigation bar automatically save any changes made by the annotator. Furthermore, annotators can see what document and sentence they are currently annotating as well as the whether there are any unsaved changes in the task status bar (Figure 2g).

We also allow annotators to update multiple instances of a word with the same orthography together. In the "Contexts" panel, annotators are shown a list of all occurrences of a word within the current document in context (Figure 4a). They can then select each context they would like to update by clicking on the check box on the left of each instance. Finally, annotators click on the "Apply to Selected" button (Figure 4a) to apply the analysis of the current word to all the selected instances.

Additionally, we provide annotators with a search utility to look up previously submitted analyses as well as query MADAMIRA for out-of-context analyses in different dialects and apply a chosen analysis in real-time using the "Analysis Search" panel (Figure 2d, Figure 4b). Annotators type in a word to query in the search bar. Clicking

[4]We use the CAMEL POS tag set and features defined by Khalifa et al. (2018).

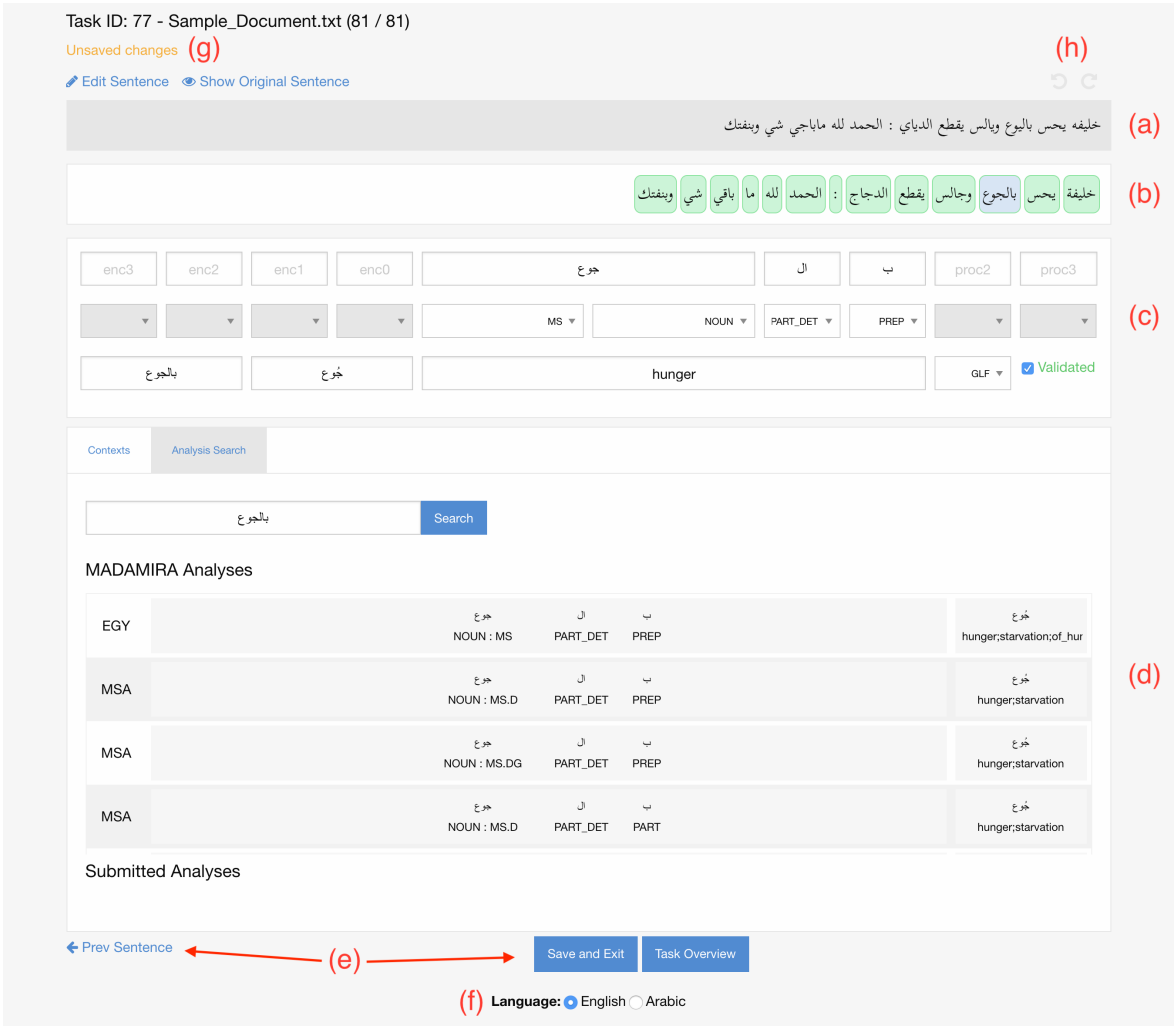
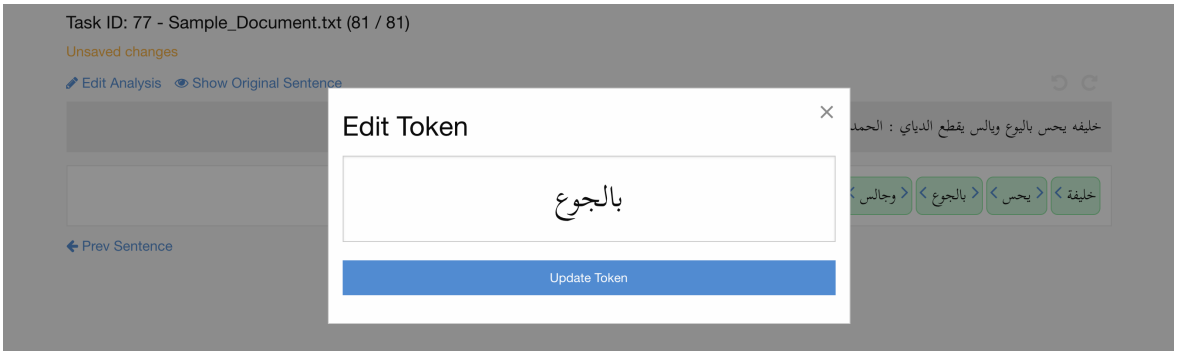


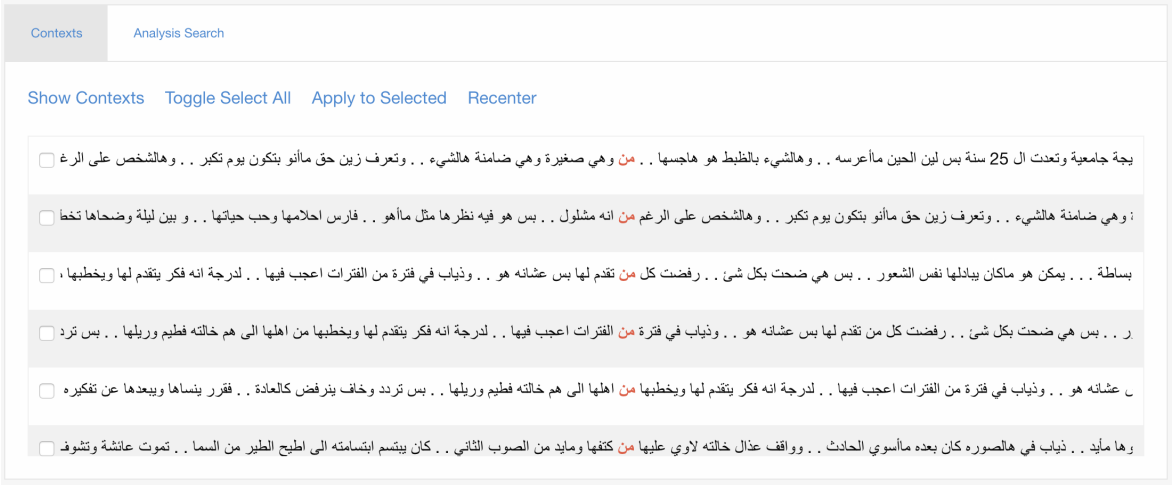

Figure 2: Full view of the MADARi annotation interface


(a) Token merge and split view

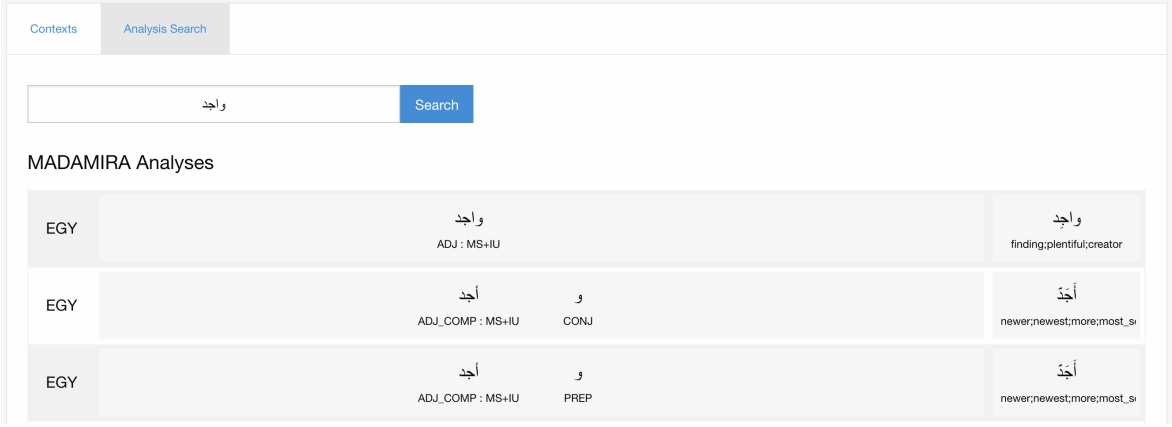



(b) Token edit pop-up

Figure 3: Edit sentence view




(a) Contexts panel view




(b) Analysis search panel.

Figure 4: Contexts and analysis search view

"Search" retrieves a list of out-of-context analyses from MADAMIRA and a list of previously submitted analyses of the search term. Double clicking on a listed analysis applies it to the current word if the current word can be tokenized in such a way that its clitics match those of the selected analysis. For example بالليل *bAllyl* 'during the night' and بالنهار *bAlnhAr* 'during the day' have the same proclitics ب+ال *b+Al+* and no enclitics, thus they have matching clitics.

## 5. Management Interface

The Annotation Management Interface enables the lead annotator to easily manage and organize the whole annotation process remotely and concurrently. The management interface contains: (a) a user management tool for creating new annotator accounts and viewing annotator progress; (b) a document management tool for uploading new documents, assigning them for annotation, and viewing submitted annotations; (c) a monitoring tool for viewing overall annotation progress; (d) a data repository and annotation export feature; and (e) a utility for importing pre-annotated documents, overriding MADAMIRA's analyses.

## 6. User Study

Our tool is being used as part of an ongoing annotation project on Gulf Arabic (Khalifa et al., 2018). In this paper, we describe the experience of one annotator who has done annotations in different settings previously. The annotator morphologically disambiguated 80 sentences totaling in 1,355 raw tokens of Gulf Arabic text.

The annotator preferred, based on her experience, to convert the orthography of the text to CODA first, which made the disambiguation task more efficient.

It took about 52 minutes to complete this task (corresponding to a rate of 1,563 words/hour). The annotator made a few minor fixes later on, which is an advantage of our tool to minimize error propagation. The total number of words that were changed from the raw tokens to CODA was 288 (21%). Changes were mostly spelling adjustments and the rest is word splitting (44 cases or 15% of all changes) but no merges. The final word count is 1,398 words.

Following the CODA conversion, the annotator worked on tokenization, POS tagging, lemmatization and English glossing. This more complex task took around 6 hours (at a rate of 277 words/hour). This makes the cumulative time

spent to finish the spelling adjustment and the full disambiguation tasks for this set of data about 7 hours (at a rate of 200 words/hour).

Since the tool provides initial guesses for all the annotation components, the annotator was able to use many of the valid decisions as is, and modify them in other cases. In the event of a word split, the tool currently removes the raw word predictions, but the analysis search utility allows fast access to alternatives to select from.

We compared the final tokenization, POS tag and lemma choices to the ones suggested by the tool on the CODA version of the text. We found that the tool gave correct suggestions 74% of the time on tokenization, 69% of the time on baseword POS tags and 70% of the time on lemmas.

The annotator indicated that their favorite utilities were the ability to annotate multiple tokens of the same type in different contexts simultaneously, and the ability to use the *Analysis Search* box to annotate multiple fields simultaneously.

## 7. Conclusion and Future Work

We presented an overview of our web-based annotation framework for joint morphological annotation and spelling correction of Arabic. We plan to release the tool and make it freely available to the research community so it can be used in other related annotation tasks. In the future, we will continue extending the tool to support different dialects and genres of Arabic.

## Acknowledgments

This publication was made possible by grant NPRP7-290-1-047 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

## Bibliographical References

Al-Shargi, F. and Rambow, O. (2015). DIWAN: A Dialectal Word Annotation Tool for Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, page 49.

Al-Twairesh, N., Al-Dayel, A., Al-Khalifa, H., Al-Yahya, M., Alageel, S., Abanmy, N., and Al-Shenaifi, N. (2016). Madad: A readability annotation tool for arabic text. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Attia, M., Rashwan, M. A., and Al-Badrashiny, M. A. (2009). Fassieh, a Semi-automatic Visual Interactive Tool for Morphological, PoS-Tags, Phonetic, and Semantic Annotation of Arabic Text Corpora. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5):916–925.

Aziz, W., de Sousa, S. C. M., and Specia, L. (2012). PET: a tool for post-editing and assessing machine translation. In *Proceedings of the LREC'2012*.

Benajiba, Y. and Diab, M. (2010). A Web Application for Dialectal Arabic Text Annotation. In *Proceedings of the LREC Workshop for Language Resources (LRS) and Human Language Technologies (HLT) for Semitic Languages: Status, Updates, and Prospects*.

Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., Obeid, O., Khalifa, S., Eryani, F., Erdmann, A., and Oflazer, K. (2018). The MADAR Arabic Dialect Corpus and Lexicon. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Diab, M., Habash, N., Rambow, O., Altantawy, M., and Benajiba, Y. (2010). Colaba: Arabic dialect annotation and processing. In *LREC Workshop on Semitic Language Processing*.

Dickinson, M. and Ledbetter, S. (2012). Annotating Errors in a Hungarian Learner Corpus. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2012)*.

El-ghobashy, A. N., Attiya, G. M., and Kelash, H. M. (2014). A Proposed Framework for Arabic Semantic Annotation Tool. 3(1).

Habash, N., Soudi, A., and Buckwalter, T. (2007). On Arabic transliteration. In Abdelhadi Soudi, et al., editors, *Arabic Computational Morphology*, volume 38 of *Text, Speech and Language Technology*, chapter 2, pages 15–22. Springer.

Habash, N., Khalifa, S., Eryani, F., Rambow, O., Abdulrahim, D., Erdmann, A., Faraj, R., Zaghouani, W., Bouamor, H., Zalmout, N., Hassan, S., shargi, F. A., Alkhereyf, S., Abdulkareem, B., Eskander, R., Salameh, M., and Saddiki, H. (2018). Unified Guidelines and Resources for Arabic Dialect Orthography. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Habash, N. (2010). *Introduction to Arabic Natural Language Processing*. Morgan & Claypool Publishers.

Javed, T., Habash, N., and Taji, D. (2018). Palmyra: A platform independent dependency annotation tool for morphologically rich languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, may.

Khalifa, S., Habash, N., Eryani, F., Obeid, O., Abdulrahim, D., and Kaabi, M. A. (2018). A Morphologically Annotated Corpus of Emirati Arabic. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Llitjós, A. F. and Carbonell, J. G. (2004). The Translation Correction Tool: English-Spanish User Studies. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04)*.

Maamouri, M., Bies, A., Kulick, S., Ciul, M., Habash, N., and Eskander, R. (2014). Developing an Egyptian Arabic Treebank: Impact of Dialectal Morphology on Annotation and Tool Development. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Obeid, O., Zaghouani, W., Mohit, B., Habash, N., Oflazer,

K., and Tomeh, N. (2013). A Web-based Annotation Framework For Large-Scale Text Correction. In *The Companion Volume of the Proceedings of IJCNLP 2013: System Demonstrations*, pages 1–4, Nagoya, Japan.

Obeid, O., Bouamor, H., Zaghouani, W., Ghoneim, M., Hawwari, A., Alqahtani, S., Diab, M., and Oflazer, K. (2016). MANDIAC: A Web-based Annotation System For Manual Arabic Diacritization. In *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media*, page 16.

Pasha, A., Al-Badrashiny, M., Kholy, A. E., Eskander, R., Diab, M., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *In Proceedings of LREC*, Reykjavik, Iceland.

Saleh, L. M. B. and Al-Khalifa, H. S. (2009). AraTation: an Arabic Semantic Annotation Tool. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, pages 447–451. ACM.

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). BRAT: A Web-based Tool for NLP-assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Stroudsburg, PA, USA.

Stymne, S. (2011). Blast: a Tool for Error Analysis of Machine Translation Output. In *Proceedings of the ACL'2011: Systems Demonstrations*, pages 56–61.

Yimam, S. M., Gurevych, I., de Castilho, R. E., and Biemann, C. (2013). WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations. In *ACL (Conference System Demonstrations)*, pages 1–6. The Association for Computer Linguistics.

Zaghouani, W., Mohit, B., Habash, N., Obeid, O., Tomeh, N., Rozovskaya, A., Farra, N., Alkuhlani, S., and Oflazer, K. (2014). Large Scale Arabic Error Annotation: Guidelines and Framework. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland.

Zaghouani, W., Habash, N., Bouamor, H., Rozovskaya, A., Mohit, B., Heider, A., and Oflazer, K. (2015). Correction annotation for non-native Arabic texts: Guidelines and corpus. *Proceedings of The 9th Linguistic Annotation Workshop*, pages 129–139.

Zaghouani, W., Habash, N., Obeid, O., Mohit, B., and Oflazer, K. (2016). Building an Arabic Machine Translation Post-Edited Corpus: Guidelines and Annotation. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Portorož, Slovenia.