DIWAN: A Dialectal Word Annotation Tool for Arabic

Faisal Al-Shargi
Universität Leipzig
Leipzig
Germany
alshargi@informatik.uni-leipzig.de

Owen Rambow
CCLS, Columbia University
New York, NY
USA
rambow@ccls.columbia.edu

Abstract

This paper presents DIWAN, an annotation interface for Arabic dialectal texts. While the Arabic dialects differ in many respects from each other and from Modern Standard Arabic, they also have much in common. To facilitate annotation and to make it as efficient as possible, it is therefore not advisable to treat each Arabic dialect as a separate language, unrelated to the other variants of Arabic. Instead, we make analyses from other variants available to the annotator, who then can choose to use them or not.

1. Introduction

Arabic is a Central Semitic language, closely related to Aramaic, Hebrew, Ugaritic and Phoenician. It is spoken by 420 million speakers (native and non-native) in the Arab World. Arabic also is a liturgical language of 1.6 billion Muslims around the world.

Modern Standard Arabic (MSA) is the official *Arabic* language. It is the educational language and official language used in news and official communication across the Arabic-speaking world. When Arabs communicate spontaneously in informal settings, they use dialectal Arabic (DA). There are divisions of many dialects of the Arabic language that occur between the spoken languages of different regions. Some varieties of Arabic in North Africa, for example, are incomprehensible to an Arabic speaker from the Levant or the Arabian Peninsula.[1]

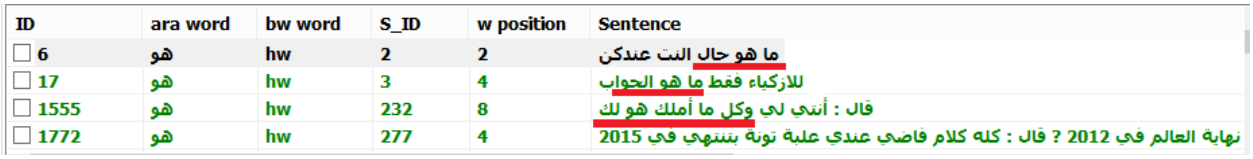
Within these broad regions, further and considerable geographic distinctions exist – within countries, across country borders, and between cities and villages. Some examples include Gulf Arabic, Bahraini Arabic, Najdi Arabic, Hijazi Arabic, Yemeni Arabic, Yemeni Hadhrami Arabic, Yemeni Sanaani Arabic, Yemeni Ta'izzi-Adeni Arabic, Dhofari Arabic, Omani Arabic, Shihhi Arabic, and the Peninsular Arabic dialects.

Despite this diversity, all Arabic dialects share certain properties: much of their phonology, templatic morphology augmented by affixes and a large set of clitics, large parts of their syntax, and important (though unpredictable) parts of the lexicon.

Current natural language processing (NLP) tools work well with MSA because they were designed specifically for the processing of MSA, and because of the abundance of MSA resources. Applying the NLP tools designed for MSA directly to DA yields significantly lower performance (Chiang et al., 2006; Habash and Rambow, 2006; Benajiba et al., 2010; Habash et al., 2012). This makes it imperative to direct research to building resources and tools for DA processing.

[1] When Arabic speakers of different dialects meet, they tend to navigate towards a *middle* Arabic that encapsulates the shared aspects they are aware of in order to maximize communication. A better and harder test of comprehension is to eavesdrop on a conversation in another dialect.

Proceedings of the Second Workshop on Arabic Natural Language Processing, pages 49–58,
Beijing, China, July 26-31, 2015.



ID	ara word	bw word	S_ID	w position	Sentence
6	هو	hw	2	2	ما هو حال البت عندكن
17	هو	hw	3	4	للأزكياء فقط ما هو الجواب
1555	هو	hw	232	8	قال : أنبي لي وكل ما أملك هو لك
1772	هو	hw	277	4	نهاية العالم في 2012 ؟ قال : كله كلام فاضي عندي علبة تونة بتنتهي في 2015

Figure 1: An example of MSA and DA code switching

Arabic dialects lack large amounts of consistent data due to two main factors: the lack of orthographic standards for the dialects, and the lack of overall Arabic content on the web (Benajiba et al., 2010). While the rise of the internet has increased the amount of DA being written, sometimes Arabic dialects come mixed with the MSA in various forms of text (see Figure 1, which shows the code switching in our DIWAN tool). Furthermore, language used in social media poses a challenge for NLP tools in general in any language due to the difference in genre. Therefore, in order to create tools for dialectal Arabic, annotated DA corpora are needed in a variety of dialects.

The goal of our **Di**alectal **W**ord **An**notation tool (DIWAN) is to address these gaps on the resource creation level. In designing DIWAN, we have determined several important design goals:

1. We want to exploit the similarity between dialects as much as possible to facilitate annotation, which in general is costly and slow.
2. We want to use a convention for orthography (which the input text does not necessarily follow).
3. We want to create data which can be used both for creating morphological analyzers (which produce all morphological analyses for a word outside of any context) and morphological taggers (which determine the correct morphological analysis -- including the POS tag -- for a word in context).

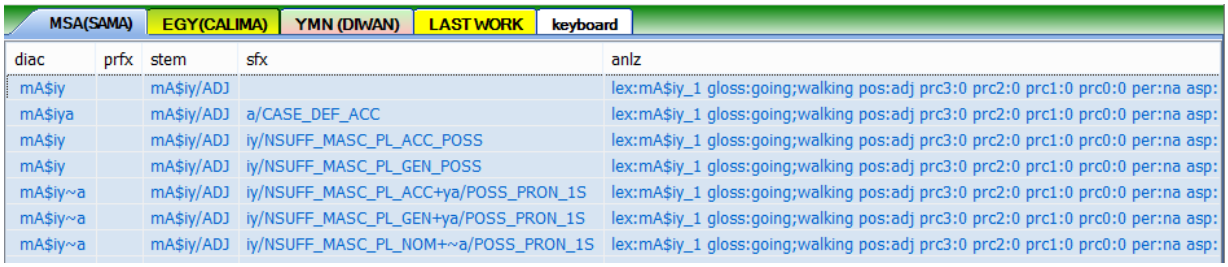
This paper explains the design decisions we have made in order to meet these goals. DIWAN is fully implemented for use on Microsoft Windows and is currently in use for the annotation of Palestinian, Yemeni, and Moroccan Arabic.

This paper is structured as follows. In Section 2, we review the NLP components we use in DIWAN. In Section 3, we describe the workflow when using DIWAN. In Section 4, we describe the specific annotation tasks the annotator performs. Section 5 gives some technical detail about the implementation. Section 6 discusses related work. We conclude in Section 7 with a discussion of future work.

2. NLP Resources used in DIWAN

In order to make the annotation task easier, DIWAN uses three main existing NLP resources: the MSA morphological analyzer SAMA, the Egyptian morphological analyzer CALIMA-EGY, and the morphological tagger MADAMIRA which works for both MSA and Egyptian. We describe them in turn.

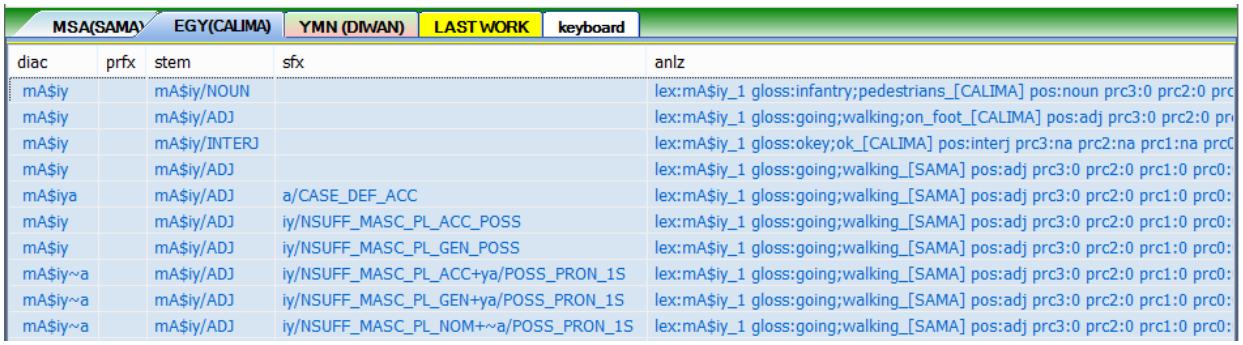
The first resource is the Standard Arabic Morphological Analyzer, SAMA 3.1 (Graff et al. 2009), which is based on the BAMA analyzer (Buckwalter 2004). This system uses lexical databases, divided into prefixes, stems, and suffixes, to assign words all possible MSA analyses. A sample output is shown in Figure 2 (in Buckwalter transliteration), for the input word ماشي *mA$i*, which is ambiguous between various inflected forms of a verb meaning `walk'.



MSA(SAMA) | EGY(CALIMA) | YMN (DIWAN) | LAST WORK | keyboard

diac	prfx	stem	sfx	anlz
mA$iy		mA$iy/ADJ		lex:mA$iy_1 gloss:going;walking pos:adj prc3:0 prc2:0 prc1:0 prc0:0 per:na asp:
mA$iya		mA$iy/ADJ	a/CASE_DEF_ACC	lex:mA$iy_1 gloss:going;walking pos:adj prc3:0 prc2:0 prc1:0 prc0:0 per:na asp:
mA$iy		mA$iy/ADJ	iy/NSUFF_MASC_PL_ACC_POSS	lex:mA$iy_1 gloss:going;walking pos:adj prc3:0 prc2:0 prc1:0 prc0:0 per:na asp:
mA$iy		mA$iy/ADJ	iy/NSUFF_MASC_PL_GEN_POSS	lex:mA$iy_1 gloss:going;walking pos:adj prc3:0 prc2:0 prc1:0 prc0:0 per:na asp:
mA$iy~a		mA$iy/ADJ	iy/NSUFF_MASC_PL_ACC+ya/POSS_PRON_1S	lex:mA$iy_1 gloss:going;walking pos:adj prc3:0 prc2:0 prc1:0 prc0:0 per:na asp:
mA$iy~a		mA$iy/ADJ	iy/NSUFF_MASC_PL_GEN+ya/POSS_PRON_1S	lex:mA$iy_1 gloss:going;walking pos:adj prc3:0 prc2:0 prc1:0 prc0:0 per:na asp:
mA$iy~a		mA$iy/ADJ	iy/NSUFF_MASC_PL_NOM+~a/POSS_PRON_1S	lex:mA$iy_1 gloss:going;walking pos:adj prc3:0 prc2:0 prc1:0 prc0:0 per:na asp:

Figure 2: SAMA result for search on word ماشي *mA$y*



MSA(SAMA) | EGY(CALIMA) | YMN (DIWAN) | LAST WORK | keyboard

diac	prfx	stem	sfx	anlz
mA$iy		mA$iy/NOUN		lex:mA$iy_1 gloss:infantry;pedestrians_[CALIMA] pos:noun prc3:0 prc2:0 prc
mA$iy		mA$iy/ADJ		lex:mA$iy_1 gloss:going;walking;on_foot_[CALIMA] pos:adj prc3:0 prc2:0 pr
mA$iy		mA$iy/INTERJ		lex:mA$iy_1 gloss:okey;ok_[CALIMA] pos:interj prc3:na prc2:na prc1:na prc
mA$iy		mA$iy/ADJ		lex:mA$iy_1 gloss:going;walking_[SAMA] pos:adj prc3:0 prc2:0 prc1:0 prc0:
mA$iya		mA$iy/ADJ	a/CASE_DEF_ACC	lex:mA$iy_1 gloss:going;walking_[SAMA] pos:adj prc3:0 prc2:0 prc1:0 prc0:
mA$iy		mA$iy/ADJ	iy/NSUFF_MASC_PL_ACC_POSS	lex:mA$iy_1 gloss:going;walking_[SAMA] pos:adj prc3:0 prc2:0 prc1:0 prc0:
mA$iy		mA$iy/ADJ	iy/NSUFF_MASC_PL_GEN_POSS	lex:mA$iy_1 gloss:going;walking_[SAMA] pos:adj prc3:0 prc2:0 prc1:0 prc0:
mA$iy~a		mA$iy/ADJ	iy/NSUFF_MASC_PL_ACC+ya/POSS_PRON_1S	lex:mA$iy_1 gloss:going;walking_[SAMA] pos:adj prc3:0 prc2:0 prc1:0 prc0:
mA$iy~a		mA$iy/ADJ	iy/NSUFF_MASC_PL_GEN+ya/POSS_PRON_1S	lex:mA$iy_1 gloss:going;walking_[SAMA] pos:adj prc3:0 prc2:0 prc1:0 prc0:
mA$iy~a		mA$iy/ADJ	iy/NSUFF_MASC_PL_NOM+~a/POSS_PRON_1S	lex:mA$iy_1 gloss:going;walking_[SAMA] pos:adj prc3:0 prc2:0 prc1:0 prc0:

Figure 3: CALIMA-Egyptian result for search on word ماشي *mA$y*

The second resource is the Columbia Arabic Language and dialect Morphological Analyzer for Egyptian (CALIMA-EGY) (Habash et al. 2012b). It is an analyzer for Egyptian. A sample output is shown in Figure 3 for the input word ماشي *mA$y*. CALIMA returns the MSA readings shown in Figure 2, and in addition has Egyptian readings, in particular the interjection `OK'.

The third resource is MADAMIRA (Pasha et al. 2014). MADAMIRA is a system for morphological analysis and disambiguation of Arabic that combines some of the best aspects of two previously commonly used systems for Arabic processing, MADA (Habash and Rambow, 2005; Habash et al., 2009; Habash et al., 2013) and AMIRA (Diab et al., 2007). MADAMIRA improves upon the two systems with a more streamlined Java implementation that is more robust, portable, extensible, and is faster than its ancestors by more than an order of magnitude. Contrary to SAMA and CALIMA-EGY, which provide all morphological analyses for a word regardless of context, MADAMIRA chooses a single analysis given the context of the word in a sentence. For example, in the sentence ماشي كده ؟ *mA$i kdh?* `Is that OK?', the interjection meaning will be chosen.

3. DIWAN Workflow

We designed and built DIWAN as a desktop application which can work locally (offline) or online. As an annotation tool, we have designed DIWAN with two types of users: administrators and annotators. The administrator's responsibility is to create the DIWAN database, specify its settings, and track the annotator's work.

The administrator has several roles:

1. She can create, edit and delete tables in the database.
2. She can create, edit and delete annotator accounts.
3. She can check the status of the annotation tasks for each annotator.
4. She can trace the annotator progress, work time, errors, etc.
5. She can generate reports and statistics on the underlying database (created by the annotators).
6. She can of course also annotate the data.

The annotators can only annotate data. The administrator assigns tasks to each annotator, and the annotations are added to the DIWAN database. As the annotator creates annotations, he

can reuse the resulting lexical entries in the DIWAN tool as a new resource for himself or for other annotators.

To work with DIWAN, the administrator first prepares the data. We assume that the data is DA written in Arabic script. There are two ways of preparing the data:

1. The administrator can either simply use DIWAN itself to identify sentences and words in the corpus. DIWAN extracts sentences and words from the prepared file and builds a DIWAN database.
2. Or the administrator can send the corpus to MADAMIRA. MADAMIRA not only identifies sentences and words, it also performs morphological analysis (using MSA and Egyptian resources) and tagging, making a single analysis available for each input word in context. After getting the resulting data from MADAMIRA, DIWAN will present the analysis for each word to the annotator as a default annotation option. As in the previous case, DIWAN extracts sentences and words from the prepared file and builds a DIWAN database (which now includes the MADAMIRA analysis).

These two options are shown in Figure 4.

The annotator makes the dialect annotations by using the DIWAN GUI. We describe this process in detail in Section 4.

4. Annotation Tasks

We describe the workflow of the annotator.

4.1. Initialization of the Annotation GUI

The annotator starts out by choosing if he wants to work locally, i.e., offline, or connected to the database. The offline option is useful when an internet connection is not reliable. In that case, the work is uploaded in batch at the end of the session.

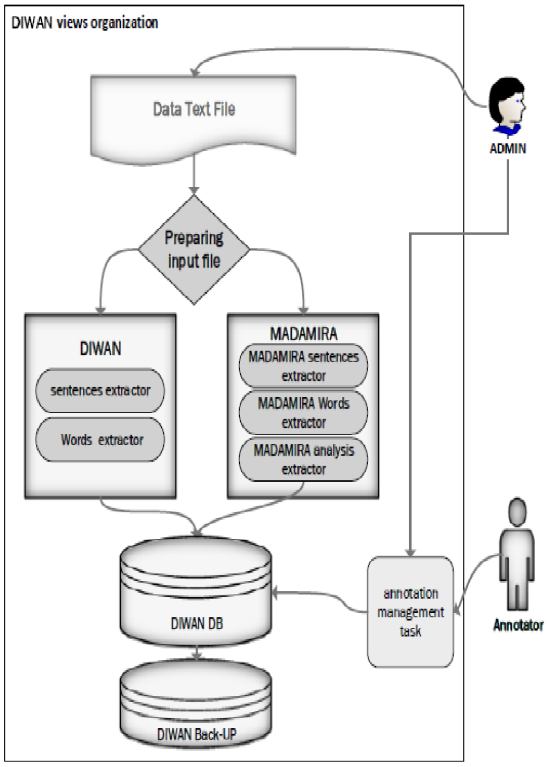


Figure 4: DIWAN setup workflow

When online, several annotators can work at once, sharing their work immediately through the centralized database.

4.2. Choice of Word to Annotate and Using the Resources

The annotator has three options of how to order the words he wants to annotate: by frequency, by text order, or by coverage. The frequency-based approach has the advantage that a large number of tokens can be annotated at once, while the text order provides a natural right-to-left annotation order through the text. Ordering by coverage moves those words to the top of the list which the MADAMIRA system cannot analyze as MSA or Egyptian. This typically (but not always, of course) means that the word is specific to the dialect in question (for example, كتب *ktb* `wrote (3ms)' is common to all dialects, while مشبوجة *m$bwjp* `puffy (fs)' is specific to Yemeni). Therefore, ordering by coverage will move

words specific to the dialect to the top. Of course, ordering by coverage also misses dialectal words which look like a word in MSA or Egyptian, but mean something else than the MSA or Egyptian counterpart ("faux amis").

In our experience, the frequency-based ordering is useful at the beginning, when the annotator can quickly annotate high-frequency words which typically don't change in form or meaning. Often, these words are dialect-specific. However, once a sufficient number of high-frequency words have been annotated, the annotator can choose to switch to the text-order view. He then continues to annotate lower-frequency words in their textual order. The color coding shows him which words are already annotated. The annotator can also hide the annotated words by clicking on a button. If an annotation effort is interested in creating a dialect-specific lexicon quickly, the ordering-by-coverage approach may be the most appropriate.

Whatever ordering criterion the annotator chooses, he sees an ordered list of words, with the already annotated words in green and the words to be annotated in red. The annotator then clicks on a word in the word panel on the left, and sees in a panel at the top of the GUI a scrollable list of all occurrences of this word in the corpus (one per line), shown in context. The annotator chooses which instances of the word he wants to annotate (i.e., which instances have the same analysis) by clicking a checkbox corresponding to that instance. Typically, he would survey all occurrences and judge which ones have the same analysis. He then chooses a representative example, clicks on it, and proceeds to the main annotation panel. When the annotator clicks on the word's checkbox, by default he will get the MADAMIRA result in the annotation panel (assuming the administrator has chosen to include the MADMIRA analyses).

The annotator performs the annotation tasks in the main annotation panel. There are several input boxes which the annotator needs to fill in as part of the annotation; we will explain them in Section 4.3. As mentioned, DIWAN retrieves an proposed analysis in context for the chosen word token from MADAMIRA and populates all text input boxes and checkboxes automatically with the analysis MADAMIRA finds, which may be based on an MSA analysis or an Egyptian analysis. In some cases, MADAMIRA does not find an analysis, in which case this is clearly shown. The annotator now has several choices as to how to enter the annotation.

1. He can accept the MADAMIRA analysis as correct in this dialect as well.
2. He can modify the MADAMIRA analysis and save the changes.
3. He can look at the list of SAMA analyses for the word (interpreting the word as MSA), and choose one. This analysis then populates all input boxes. He can then choose to accept this analysis, or modify and save it.
4. He can look at the list of CALIMA analyses for the word (interpreting the word as Egyptian), and choose one. This analysis then populates all input boxes. He can then choose to accept this analysis, or modify and save it.
5. He can do word substitution: if the word does not produce the correct (or any) analysis in SAMA or CALIMA, but he knows a word that does and that has the same morphological analysis, then he can enter that word, search in SAMA or (more likely) CALIMA, and then edit the analysis, but only to modify the word itself. For example, assume the annotator is working on Yemeni and the input word is يقطب *yqTb `speeds up (3ms)'*. This does not produce an analysis in SAMA or CALIMA. So instead, the annotator searches for يكتب yktb `writes (3ms)', which has exactly the same morphological analysis, and then edits the stem by replacing كتب with قطب , and updates the gloss to `hasten, speed up'.
6. Finally, he can create an analysis from scratch. This would normally be the most time consuming option.

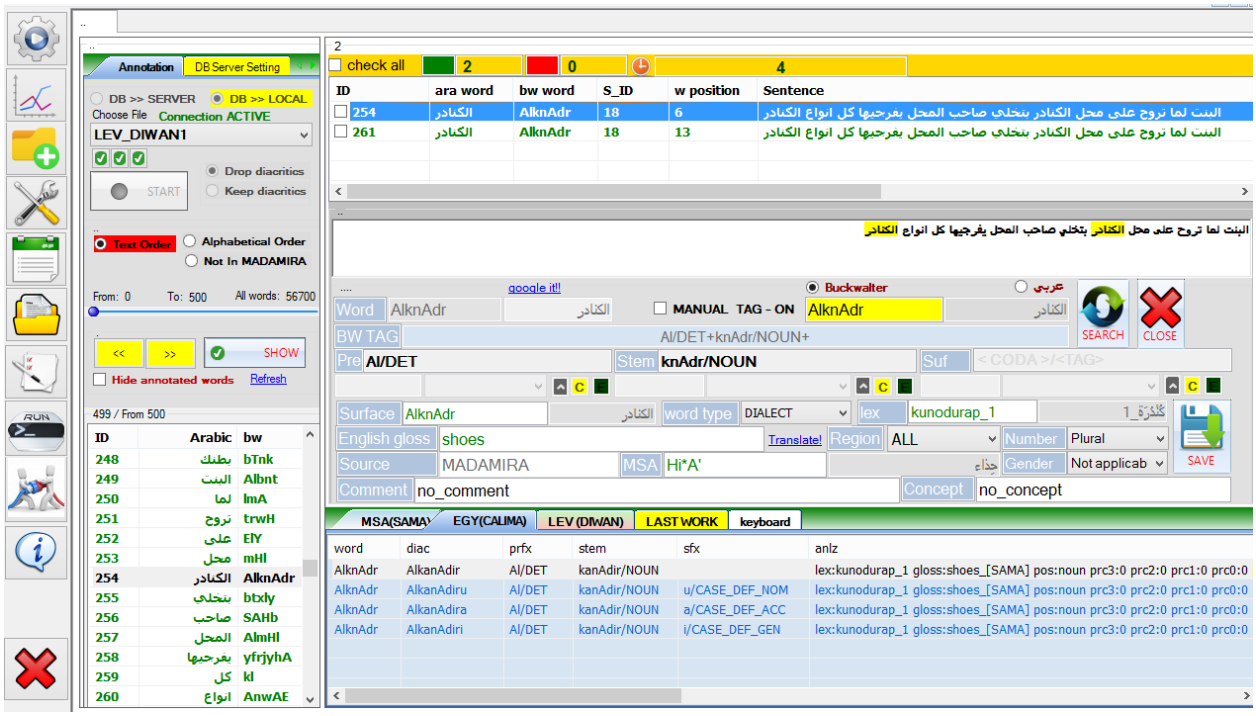


Figure 5: The main DIWAN annotation interface

4.3. Specific Annotation Tasks

Each annotation task corresponds to a specific input device. The interface is shown in Figure 5. Note that here we describe the annotation task as if it is performed from scratch (option 6 in Section 4.2 above).

1. Rewriting the word in the conventionalized orthography (CODA) defined for that dialect (Habash et al. 2012a). Note that CODA may include diacritics or not; in the examples we show in this paper, it does not.
2. Breaking into prefix, stem and suffix. These two tasks are performed jointly in three text input boxes.
3. Adding morph-specific features (in the style of the Linguistic Data Consortium Arabic resources). This task is performed using drop-down menus separately for the prefix, stem, and suffix.
4. Adding the English gloss and MSA equivalent. This task is performed in two dedicated text input boxes.
5. Adding functional morphology. The morpheme-based annotation performed using drop-down menus adds morphological information to morphs. For example, Egyptian باصات `busses' is annotated at the morph level as bAS/NOUN+At/NSUFF_FEM_PL, since +ات is the suffix for regular feminine plural nouns. However, the form is in fact a masculine plural form (and thus a type of broken plural), so that the annotator would mark باصات as functionally masculine and plural. This task is performed using two drop-down menu boxes (one for number, one for gender). Note that none of the existing resources (MADAMIRA, SAMA, or CALIMA) mark functional number and gender, so that this task needs to be performed manually for each word in any case.
6. Marking Arabic variant. The annotator can choose to mark whether a word is in fact MSA rather than dialect (the default assumption). This is useful when code switching occurs, and the annotator does not want to add an MSA word to the dialectal vocabulary. Furthermore, the annotator can choose a specific region within the dialect. This is useful when a dialectal form is not typical for the region that the text is from. For example a Lebanese word may occur in a Palestinian Arabic text, such as بيي وعيلتي كلها بالضفه *byy wEylty*

klhA bAlDfh `my father and all my family in West Bank', where all words are Palestinian, except بيي byy 'my father', which is more used in Lebanese Arabic.[2]

When everything is correct, the annotator can save his annotation directly to the database; as a result, the color of the analyzed word token or tokens will change from red to green.

In addition, we have added some functionalities in order to help the annotators in their annotation, like Google search on a word (which the annotator can use to verify the meaning; often image search is useful for this purpose), and Google translation for finding the English gloss.

5. The Database and Output Files

In this section, we briefly summarize the databases and file formats used by DIWAN. Only the administrator has the ability to directly access these databases, the annotators can only access it through the DIWAN interface. This ensures the integrity of the DIWAN data. The database has three main tables, the D_sentences table, the D_madamira table, and the D_result table. The D_sentences table includes all the words organized into sentences from the input. The D_madamira table contains the result of the MADAMIRA analysis on the input text. The D_result table is the table that contains all work by the annotators.

The administrator can at any time produce a file output from DIWAN which reflects the annotation. The file includes the results of MADAMIRA if no manual annotation has been done on it. A sample output is shown in Figure 6. We briefly summarize this format:

ثلاثة زعران قاعدين بمزرعه - vlAvh zErAn qAEdyn bmzrEh

924 ثلاثة vlAvp diac:vlAvp lex:valAv_1 bw:+vlAv/NOUN_NUM+p/NSUFF_FEM_SG msa:valAv_1 gloss:three pos:noun_num gen:f num:s region:ALL diwan_source:MADAMIRA source_mod:no source_search:vlAvp anno:diwan_approved

925 زعران zErAn diac:zErAn lex:>azoEar_2 bw:+zErAn/NOUN+ msa:>azoEar_2 gloss:brigands;scoundrels pos:noun gen:m num:p region:ALL diwan_source:MADAMIRA source_mod:yes source_search:zErAn anno:diwan_approved

926 قاعدين qAEdyn diac:qAEdyn lex:qAEid_1 bw:+qAEd/ADJ+yn/NSUFF_MASC_PL msa:jAls_1 gloss:sitting;seated;lazy;inactive;evaders_(draft_dodgers) pos:adj gen:m num:p region:ALL diwan_source:MADAMIRA source_mod:no source_search:qAEdyn anno:diwan_approved

927 بمزرعه bmzrEh diac:bmzrEp lex:mazoraE_1 bw:b/PART+mzrE/NOUN+p/NSUFF_FEM_SG msa:mazoraE_1 gloss:farm;plantation pos:noun gen:f num:s region:ALL diwan_source:EGY source_mod:no source_search:bmzrEp anno:diwan_approved

Figure 6: Extract of an output file generated by DIWAN of an annotated text in context

924: the word number in the text
ثلاثة **, vlAvp:** the word in Arabic script and Buckwalter transliteration
diac:vlAvp: The CODA spelling (which, recall, may or may not be diacritized)
lex:valAv_1: the lexeme
bw:+vlAv/NOUN_NUM+p/NSUFF_FEM_SG The Buckwalter part-of-speech and morpheme split; this is a the morpheme-based morphological annotation; the plusses indicate the boundaries between prefix, stem, and suffix.
msa:valAv_1: MSA equivalent
gloss:three: English gloss
pos:noun_num: The core part-of-speech tag
gen:f: functional gender

[2] Palestinian Arabic is particularly challenging due the common dialect mixing in different sub-varieties of it resulting from the particular situation of Palestinian refugees in different countries.

num:s: functional number
region:ALL: applicable dialectal subregion
diwan_source:MADAMIRA: which resource did the annotator use in DIWAN
source_mod:no: did the annotator modify the source?
source_search:vlAvp: what keyword did the annotator use to search the resource? (In this case, since the resource is MADAMIRA, the search keyword is necessarily the word itself.)
anno:diwan_approved: did an annotator work on this word?

6. Related Work

There are two related interfaces that have been used for annotating dialectal Arabic that we are familiar with.

The annotation tool used at the Linguistic Data Consortium for annotating the Egyptian Treebank (Maamouri et al. 2014) is based on previous interfaces used at the LDC for treebanking, notably for MSA. The approach towards morphological annotation used at the LDC is a bootstrapping approach, which aims at developing an annotated corpus in conjunction with a morphological analyzer. The morphological analyzer developed in conjunction with the Egyptian Arabic Treebank is in fact, the same CALIMA-Egyptian system we use. In contrast to DIWAN, there is no attempt at incorporating resources from other dialects, which is also due to the fact that the Egyptian Treebank was a pioneer in the area of resources for dialectal Arabic. Furthermore, the LDC interface does not support annotation of functional number and gender, and concentrates on morpheme-based annotation (which DIWAN also supports, following the LDC approach).

The COLABA annotation tool (Diab et al. 2010a) is a web application, unlike DIWAN which is a desktop application. As a result, unlike DIWAN, the COLABA tool does not support offline work. The most important difference is that COLABA is oriented towards lexicon creation, not annotation in context. Thus, words in context are not assigned morphological features. For our work, it is crucial that we get an annotation of morphological features in context so that DIWAN can be used to create corpora to train taggers. Furthermore, COLABA does not use resources from other dialects, as does DIWAN.

7. Conclusion and Future Work

We have presented DIWAN, a tool designed for the morphological annotation of Arabic dialectal text. It incorporates resources from other dialects (and new resources can be included as they become available) in order to lighten the annotator burden. It uses a conventionalized spelling for Arabic dialects which is maintained in parallel with the naturally occurring spontaneous orthography. And it generates a file format which preserves the linear order of the input text, so that it can be used both for deriving morphological analyzers, and for training morphological taggers.

DIWAN has been used to annotate Levantine (Palestinian) Arabic (Jarrar et al. 2014). The annotators for Levantine quickly became proficient with using the tool after annotating about 100 words. The Palestinian corpus includes 45,000 annotated words (tokens). We are currently using DIWAN to annotate Yemeni (Sana'ai) Arabic. The Yemeni corpus contains 32,325 words (tokens), and the annotator for Yemeni is the first author of the present paper. Finally, we have embarked on a small project for Moroccan Arabic. We have collected 64,171 words of Moroccan for annotation. In separate publications in the future, we will report on the Yemeni and Moroccan annotation efforts. We will also report on a general methodology about how to use such resources to create morphological analyzers and taggers.

One interesting question is how our tool compares to other annotation tools. We believe that the built-in access to morphological analyzers for other variants is unique, and provides a specific advantage in annotating Arabic dialects. How-

ever, we have not performed experiments to show this. While such experiments would be very useful, they would also be quite costly, since the same texts would need to be annotated twice by different annotators.

We will continue to improve the DIWAN tool. As more dialects are annotated, we intend to add the created resources to the interface to make them available to users working on new dialects (parallel to the SAMA and CALIMA-Egyptian resources).

Currently, DIWAN is available only for Microsoft Windows. We are investigating reimplementing it in a platform-independent manner. DIWAN is freely available; for information, please consult the following URL:

http://volta.ccls.columbia.edu/~rambow/diwan/home.html

Acknowledgments

This paper is based upon work supported by DARPA Contract No. HR0011-12-C-0014. 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. We thank Nizar Habash for many helpful suggestions, including the idea of providing the ability of searching using similar words. We thank three anonymous reviewers for helpful feedback which has improved the presentation of this paper. We thank Mustafa Jarrar for very useful discussions and feedback. We also thank the Levantine annotators, Faeq Alrimawi, Diyam Akra, and Linda Alamir, as well as Aidan Kaplan who is developing the Moroccan corpus, for their feedback which helped improve the tool.

References

Y. Benajiba and M. Diab. 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*.

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

D. Chiang, M. Diab, N. Habash, O. Rambow, and S. Shareef. 2006. Parsing Arabic Dialect. In Proceedings of the European chapter of the Association of Computational Linguistics (EACL).

M. Diab, N. Habash, O. Rambow, M. AlTantawy, and Y. Benajiba. 2010a. COLABA: Arabic Dialect Annotation and Processing. In Proceedings of the Language Resources (LRs) and Human Language Technologies (HLT) for Semitic Languages at LREC.

M. Diab, K. Hacioglu, and D. Jurafsky. 2007. Automated Methods for Processing Arabic Text: From Tokenization to Base Phrase Chunking. In van den Bosch, A. and Soudi, A., editors, Arabic Computational Morphology: Knowledge-based and Empirical Methods. Kluwer/Springer.

D. Graff, M. Maamouri, B. Bouziri, S. Krouna, S. Kulick, and T. Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

N. Habash and O. Rambow. 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 573–580, Ann Arbor, Michigan.

N. Habash and O. Rambow. 2006. MAGEAD: A Morphological Analyzer and Generator for the Arabic Dialects. In Proceedings of ACL, Sydney, Australia.

N. Habash, M. Diab, and O. Rabmow. 2012a. Conventional Orthography for Dialectal Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Istanbul.

N. Habash, R. Eskander, and A. Hawwari. 2012b. A Morphological Analyzer for Egyptian Arabic. In Proc. of the Special Interest Group on Computational Morphology and Phonology, Montréal, Canada.

N. Habash, O. Rambow, and R. Roth. 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. The MEDAR Consortium, April.

N. Habash, R. Roth, O. Rambow, R. Eskander, and N. Tomeh. 2013. Morphological Analysis and Disambiguation for Dialectal Arabic. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Atlanta, GA.

M. Jarrar, N. Habash, D. Akra, and N. Zalmout 2014. "Building a Corpus for Palestinian Arabic: a Preliminary Study." In *Proceedings of the Workshop on Arabic Natural Language Processing (ANLP 2014)*.

M. Maamouri, A. Bies, S. Kulick, M. Ciul, N. Habash and R. Eskander. 2014. Developing a dialectal Egyptian Arabic Treebank: Impact of Morphology and Syntax on Annotation and Tool Development. In Proceedings of the Lan- guage Resources and Evaluation Conference (LREC), Reykjavik, Iceland.

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