Developing an Egyptian Arabic Treebank:
Impact of Dialectal Morphology on Annotation and Tool Development

Mohamed Maamouri, Ann Bies, Seth Kulick, Michael Ciul,
[*]Nizar Habash and [*]Ramy Eskander

Linguistic Data Consortium
University of Pennsylvania
3600 Market Street, Suite 810
Philadelphia, PA 19104 USA
E-mail: {maamouri,bies,skulick,mciul}@ldc.upenn.edu

[*]Center for Computational Learning Systems
Columbia University
475 Riverside Drive, Suite 850, MC7717
New York, NY 10115 USA
E-mail: {habash,reskander}@ccls.columbia.edu

Abstract

This paper describes the parallel development of an Egyptian Arabic Treebank and a morphological analyzer for Egyptian Arabic (CALIMA). By the very nature of Egyptian Arabic, the data collected is informal, for example Discussion Forum text, which we use for the treebank discussed here. In addition, Egyptian Arabic, like other Arabic dialects, is sufficiently different from Modern Standard Arabic (MSA) that tools and techniques developed for MSA cannot be simply transferred over to work on Egyptian Arabic work. In particular, a morphological analyzer for Egyptian Arabic is needed to mediate between the written text and the segmented, vocalized form used for the syntactic trees. This led to the necessity of a feedback loop between the treebank team and the analyzer team, as improvements in each area were fed to the other. Therefore, by necessity, there needed to be close cooperation between the annotation team and the tool development team, which was to their mutual benefit. Collaboration on this type of challenge, where tools and resources are limited, proved to be remarkably synergistic and opens the way to further fruitful work on Arabic dialects.

Keywords: Egyptian Arabic, Dialectal Treebank, Dialectal Morphological Analyzer

1. Introduction

Egyptian Arabic, like other Arabic dialects, is sufficiently different from Modern Standard Arabic (MSA) that tools and techniques developed for MSA cannot be simply transferred over to work on Egyptian Arabic work (Maamouri, et al., 2006; Habash, et al., 2013). It is therefore important to develop annotated language resources for Egyptian Arabic and to develop such tools directly on Egyptian Arabic. In this paper we describe work on building a treebank for Egyptian Arabic. There are two immediate challenges in creating such a resource. First, by the very nature of Egyptian Arabic the data collected is more informal than MSA data, for example discussion forum text, which we use for the treebank discussed here. Second, a morphological analyzer is needed to mediate between the written text and the segmented, vocalized form used for the syntactic trees. The earlier MSA-based work benefited from the existence of a wide-coverage MSA morphological analyzer (SAMA) (Maamouri, et al., 2010). An analogous analyzer was required for the creation of the Egyptian Arabic treebank.

These two issues were exacerbated by tight project timelines, which did not allow for the annotation to follow the creation of the morphological analyzer for Egyptian Arabic (CALIMA-ARZ, or CALIMA-Egyptian, henceforth CALIMA) (Habash, et al., 2012b). Instead, the development of the analyzer and the treebank had to proceed in parallel. This led to the necessity of a feedback loop between the treebank team and the analyzer team, as improvements in each area were fed to the other. Therefore, by necessity, there needed to be close cooperation between the annotation team and the tool development team, which was to their mutual benefit.

We first give some background on the linguistic characteristics of Egyptian Arabic and previous work in this area. We then describe the project pipeline, with a focus on how the different steps reinforce each other.

2. Linguistic Facts

2.1 Arabic and its Dialects

The linguistic reality of what is commonly called ‘the Arabic Language’ is characterized by the coexistence of a standard written language, ‘Modern Standard Arabic’ (MSA), and a multitude of regional dialects with significant geographic and social variation (Holes, 2004; Habash, 2010). These dialects, which have their own sub-dialects, show important phonological, morphological, syntactic and lexical intra- and inter-linguistic differences.

Since Arabic dialects are not officially written or standardized, they are deprived of the vast resources that researchers enjoy when working with MSA, their written counterpart. It is difficult to find officially written text resources for most Arabic dialects. The scarcity or non-availability of officially written dialectal Arabic resources is somewhat alleviated by the growing presence of unofficial written electronic media: such as web logs, discussion forums (DF), bulletin boards, and SMS-chats. Internet-based non-formal communication seems to increasingly favor the use of dialectal Arabic (DA) over MSA. However, since there is no standardized orthography for any Arabic dialect, ad hoc transcriptions have often been used with a significant degree of noise and a high level of inconsistency in the orthographic forms of the data whether it be in Arabic script or in a Romanized representation of it. Recent efforts by Habash, et al. (2012a) introduced a computationally oriented conventional orthography for DA (henceforth, CODA), to address some of these issues.

Annotated and non-annotated corpora are both essential for Arabic NLP research. The lack of DA resources has important negative consequences for most tasks. The problem faced by the NLP research community is that developing such resources is quite expensive and time consuming: guidelines need to be developed, annotators must be hired and trained, there is a need for regular evaluation and quality control (QC), and finally also a need for new tools for DA complex morphologies. The reason for this last specific need is that most of the differences between dialectal Arabic and MSA will be in the morphological structures of the words, and this will extend to their written forms. Unfortunately, the resulting data will inevitably show noise and inconsistencies. These are partly due to the time/cost constraints but also to the nature of the domain itself and the lack of stability of the orthography and sometimes of the linguistic structures themselves.

The challenges experienced in a jumpstart of the morphological annotation for a 25K pilot Levantine Arabic Treebank (using SAMA for the morphological output because of the existence of significant similarities between MSA and Arabic dialects) showed that Arabic dialects have to be treated as new and separate languages (Maamouri, et al., 2006). Habash and Rambow (2006) and Habash, et al. (2012b) showed that using MSA analyzers for both Levantine Arabic and Egyptian Arabic achieved a coverage in the low 60%.

2.2 Egyptian Arabic

Egyptian Arabic has the advantage over all other dialects of Arabic of being the language of the largest linguistic community in the Arab region, and also of having a rich level of internet communication. This connected well with the informal genres required by the project, such as the discussion forum genre.

Because of the difficulty of recruiting Egyptian Arabic native speakers in the US, contacts were established with the University of Alexandria in Egypt, and a dozen annotators were recruited for the Egyptian Arabic morphological annotation task. They were trained and took part in the drafting of the Egyptian Arabic Morphological Annotation Guidelines (Maamouri, et al., 2012i). These guidelines deal specifically with the morphological features of Egyptian Arabic, and focus on function words and other particular characteristics that are distinct from MSA.

2.3 Linguistic Features of Egyptian Arabic

Phonologically, Egyptian Arabic is characterized by the following features:

> (a) the loss of the interdentals /ð/ and /θ/ which are replaced by /d or z/ and /t or s/ respectively, thus giving those two original consonants a heavier load. Examples include ذكر /zakar/ *to mention*, ذبح /dabaħ/ *to slaughter*, ثلج /talg/ *ice*, ثمن /taman/ *price*, and ثبت /sibit/ *to stay in place, become immobile*.
>
> (b) the exclusion of /q/ and /ǰ/ from the consonantal system, being replaced by the /ʔ/ and /g/, e.g., قطن /ʔuṭn/ *cotton*, and جمل /gamal/ *camel*.

At the level of morphology and syntax, the structures of Egyptian Arabic closely resemble the overall structures of Modern Standard Arabic with relatively minor differences to speak of. Finally, the Egyptian Arabic lexicon shows some significant elements of semantic differentiation.

The most important morphological difference between Egyptian Arabic and Modern Standard Arabic is in the use of some Egyptian clitics and affixes that do not exist in Modern Standard Arabic. For instance, Egyptian Arabic has the future proclitics h+ and ħ+ as opposed to the standard equivalent s+.

Lexically, there is a lexical difference between Egyptian Arabic and MSA where no etymological connection or no cognate spelling is available. For example, the Egyptian Arabic بص /buSS/ *look* is أنظر /'unZur/ in MSA.

For a more extended discussion of the differences between MSA and Egyptian Arabic, see Habash, et al. (2012b).

3. Approach to Simultaneous Annotation and Morphological Analyzer Development

A key aspect of this work was ensuring that the annotation in the treebank and the analyzer would agree on parts-of-speech (POS), lemmas, and vocalizations.

3.1 Bootstrapping CALIMA

CALIMA-ARZ (or CALIMA Egyptian) refers to the Columbia Arabic Language and dIalect Morphological Analyzer (CALIMA) for Egyptian Arabic (ARZ[1]) (Habash, et al., 2012b). We will refer to this tool here as CALIMA.

[1] ARZ is the ISO designation for the Egyptian Arabic dialect.

The first version of CALIMA was built using the Egyptian Colloquial Arabic Lexicon (ECAL) (Kilany, et al., 2002), which was developed as part of the CALLHOME Egyptian Arabic (CHE) corpus (Gadalla, et al., 1997). This process consisted of two steps. First, ECAL entries had to be converted to a form usable by the LDC. ECAL included 66K entries, which we converted into diacritized Arabic script words and lemmas. In ECAL, only phonological form and undiacritized orthography are available. We used a technique described in more detail in (Habash, et al., 2012b) to combine the phonological form and undiacritized Arabic script into diacritized Arabic script that is useable by the LDC for the annotation process. A finite-state transducer (FST) was implemented to map a phonological form to multiple possible diacritized Arabic script forms. Then the form that is the same as the undiacritized orthography (except for diacritics) is used as the diacritized orthography.

Most of this conversion effort was accomplished with manual linguistic mapping rules followed by manual checking and correction (Habash, et al., 2012b). Second, after the converted ECAL examples are used to construct the databases of the morphological analyzer which specify all prefixes, suffixes and stems, in addition to encoding all allowable pairings among them, accomplishing some degree of paradigm completion automatically. Furthermore, a set of manually specified orthographic variants of the prefixes and suffixes were used to add entries automatically. For further details on the development of CALIMA, see (Habash et al., 2012b).

3.2 First Stages of Annotation

The treebank annotation goes through two stages. The first is POS/morphological annotation, and the second is the syntactic tree construction. The first stage is the one in which the morphological analyzer is needed, as each word is given as input, and the annotator (ideally) chooses one of the available solutions.

We started this process using the first version of CALIMA, the one just described. There were, naturally, many “holes” in the analyzer, for which it either did not provide a solution for a given input word, or did not provide the desired solution. However, we did not want to simply leave such words unannotated. The first solution taken was simply to allow the annotator to enter “proposed” solutions. As could be expected, such manual entries were prone to error. For example, for the word قتلوه `qtlwh` *they killed him*, annotators originally supplied the following annotation, which includes incorrect vocalizations for the Egyptian Arabic word:

```
qatal/PV+uw/PVSUFF_SUBJ:3P+hu/PVSUFF_DO:3MS
  قَتَلُ   +   وُ   +   ه
```

The correct annotation should be

```
qatal/PV+uwA/PVSFF_SUBJ:3P+uh/PVSUFF_DO:3MS
  قَتَل   +   واُ   +   هُ
```

To overcome this problem, we modified the CALIMA tables to allow the generation of “wildcard” solutions, in which the analyzer’s output also included solutions, in which the stem for an open-class word (noun, etc.) would be unvocalized, but the prefixes and suffixes exactly matched the possibilities elsewhere in CALIMA. The idea was that while we expected open-class items to have missing solutions, the closed-class items and morphemes (pronouns, etc.) should not, so we could at least restrict what annotators could enter for those aspects of new solution.

The CALIMA analyzer is based, like SAMA before it, on tables which takes as input the input string, and output a morphological description with both a (possibly) complex POS tag and vocalization of the different segments in the solution (e.g., prefix, stem, suffix). We modified these tables to translate CALIMA into a finite-state transducer, so that it could run bi-directionally, and given a POS tag, it would output all possible vocalizations for that POS tag. The process of reorganizing and analyzing the CALIMA tables to produce the finite state version also allowed us to extend those tables with the “wildcard” solutions, by associating each (prefix, suffix) combination that appeared with some POS tag as a possible solution for any input word that matched that (prefix, suffix), with the stem required to be filled in by the annotator.

3.3 CALIMA Revision

At this point, the annotated solutions that did not match CALIMA solutions were sent to the CALIMA team. A solution is considered to match CALIMA if it matches in all its component parts: diacritized form, lemma, POS tag and morpheme segmentation. The non-matching solutions included (1) wildcard solutions, with a stem that was missing in CALIMA, and (2) fully manual solutions.

The CALIMA team used these cases as potential additional entries for the analyzer. Of course, they could not be simply entered into the CALIMA lexicon automatically. They went through a process of arbitration (sometimes requiring further joint discussion by the treebank and analyzer teams) and normalization, before being entered into the CALIMA tables.

Alongside this process, monthly meetings between the LDC and Columbia were held to revise and update the guidelines and choices made in some CALIMA-ARZ entries based on feedback from the annotation team.

3.4 Treebank Revision and Further Annotation

After a new CALIMA version was created, it was then integrated into the POS/morphological annotation stage of treebank annotation work. Since it had fewer “holes”, using it improved the annotation process, since it was more frequently the case that the desired solution was available for the annotator, lessening the need for wildcard or manual solutions.

However, while analyzer coverage was improved, it was not perfect of course, and "holes" still remained in the analyzer, and the annotators still needed to use manual and wildcard solutions. And so the cycle described in Section 3.3 and this section repeats, as the remaining new solutions are sent to the analyzer team, which creates a new version of the analyzer, which is sent back to the treebanking team, and so on.

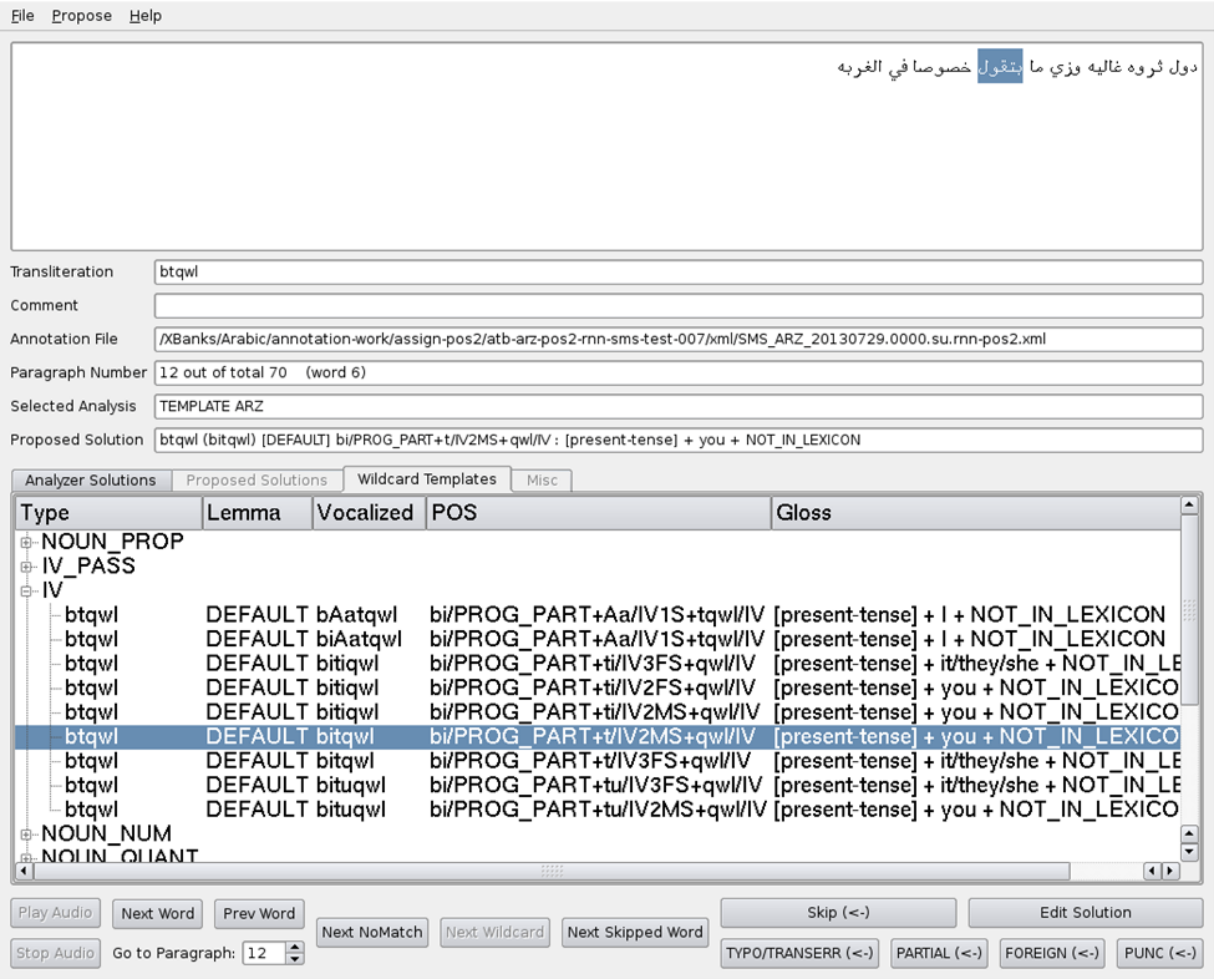
Proposed solutions (manual or wildcard), are given to the analyzer team, and integrated into the analyzer, but they do not necessarily exactly match the solutions as existing in the treebank. However, we have the goal of making the analyzer and treebanking in sync as much as possible, so that the morphological solutions in the analyzer exactly match a solution in CALIMA.

3.5 Adaptation of Morphological Annotation Tool

We have updated the POS annotation tool to enable faster and more accurate annotation. Annotators have three levels of detail that they may employ to select the best solution for a token: analyzer solutions, wildcard solutions, and proposed solutions.

The best option, whenever possible, is for the annotator to have the correct solution available without having to enter anything manually. A morphological analyzer engine provides solutions from databases of Standard Arabic (SAMA) and Egyptian Arabic dialect (CALIMA).

Figure 1 shows a screenshot of the POS annotation tool, demonstrating annotation using a wildcard solution.



دول ثروه غاليه وزي ما بتقول خصوصا في الغربه

Transliteration: btqwl

Comment:

Annotation File: /XBanks/Arabic/annotation-work/assign-pos2/atb-arz-pos2-rnn-sms-test-007/xml/SMS_ARZ_20130729.0000.su.rnn-pos2.xml

Paragraph Number: 12 out of total 70 (word 6)

Selected Analysis: TEMPLATE ARZ

Proposed Solution: btqwl (bitqwl) [DEFAULT] bi/PROG_PART+t/IV2MS+qwl/IV : [present-tense] + you + NOT_IN_LEXICON

Type	Lemma	Vocalized	POS	Gloss
NOUN_PROP				
IV_PASS				
IV				
btqwl	DEFAULT	bAatqwl	bi/PROG_PART+Aa/IV1S+tqwl/IV	[present-tense] + I + NOT_IN_LEXICON
btqwl	DEFAULT	biAatqwl	bi/PROG_PART+Aa/IV1S+tqwl/IV	[present-tense] + I + NOT_IN_LEXICON
btqwl	DEFAULT	bitiqwl	bi/PROG_PART+ti/IV3FS+qwl/IV	[present-tense] + it/they/she + NOT_IN_LE
btqwl	DEFAULT	bitiqwl	bi/PROG_PART+ti/IV2FS+qwl/IV	[present-tense] + you + NOT_IN_LEXICO
btqwl	DEFAULT	bitiqwl	bi/PROG_PART+ti/IV2MS+qwl/IV	[present-tense] + you + NOT_IN_LEXICO
btqwl	DEFAULT	bitqwl	bi/PROG_PART+t/IV2MS+qwl/IV	[present-tense] + you + NOT_IN_LEXICO
btqwl	DEFAULT	bitqwl	bi/PROG_PART+t/IV3FS+qwl/IV	[present-tense] + it/they/she + NOT_IN_LE
btqwl	DEFAULT	bituqwl	bi/PROG_PART+tu/IV3FS+qwl/IV	[present-tense] + it/they/she + NOT_IN_LE
btqwl	DEFAULT	bituqwl	bi/PROG_PART+tu/IV2MS+qwl/IV	[present-tense] + you + NOT_IN_LEXICO
NOUN_NUM				
NOUN_QUANT				

Go to Paragraph: 12

Figure 1. Wildcard annotation in the Egyptian Arabic morphological annotation tool

4. Improved Synchronization of CALIMA and Treebank Annotation

The graph in Figure 2 shows the increase in synchronization between the treebank (Maamouri, et al., 2012a-h) and CALIMA over the CALIMA versions 0.2, 0.3, and 0.4.2. We measure the synchronization in four ways, represented by the four lines. In all cases we are comparing the annotation in the treebank for the source token string to the possible solutions for that source token string available in the CALIMA version.

1) pos – The POS tag in the treebank is the same as the POS tag for at least one solution in CALIMA (for this source token string).

2) pos-lemma – Both the POS tag and lemma in the treebank match the POS tag and lemma for at least one solution in CALIMA.

3) pos-voc – Both the POS tag and vocalization in the treebank match the POS tag and vocalization for at least one solution in CALIMA.

4) pos-voc-lemma – The POS tag, vocalization, and lemma in the treebank match the POS tag, vocalization, and lemma for at least one solution in CALIMA.

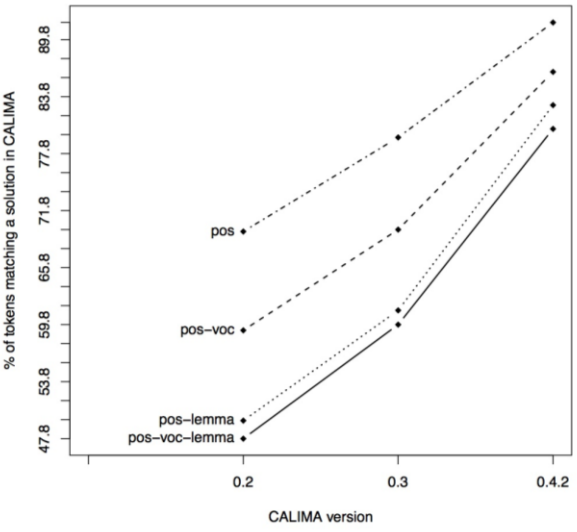


Figure 2. Improvement in synchronization between successive CALIMA versions and Egyptian Arabic morphological annotation

As the graph shows, there is an increasing coverage of the tokens in the treebank over the three CALIMA versions. Naturally, the least demanding metric, matching on POS tags only, has the highest percentage of coverage, while the most demanding, matching on pos-voc-lemma, has the least coverage, although the gap between them also narrows in version 0.4.2.[2]

These percentages are based over the numbers in the treebank which (1) are classified as Egyptian Arabic (ARZ), (2) are not punctuation or digits, which are irrelevant for CALIMA classification, and (3) have a solution, rather than the NO_FUNC placeholder for when a solution wasn't available at all.

1.5% of the tokens across the entire Egyptian Arabic corpus are NO_FUNC. This figure does however go down when considered by corpus section (Maamouri, et al., 2012a-h), as in Table 1.

Corpus section	%NO_FUNC
ARZ Part 1	1.8%
ARZ Part 2	1.6%
ARZ Part 3	1.7%
ARZ Part 4	1.5%
ARZ Part 5	1.5%
ARZ Part 6	1.7%
ARZ Part 7	1.3%
ARZ Part 8	1.0%

Table 1: Improvement in CALIMA coverage over successive Egyptian Arabic corpus segments.

The synchronization however decreases with the current version of CALIMA, 0.5, which was prepared after the annotation in these sections was completed. A main reason for this is that the 0.5 version consolidated and eliminated many alternate forms that were present in the earlier versions of CALIMA. For example, the first person pronoun was present as both أنا >anA and انا AnA in CALIMA 0.4.2, and was also present in both ways in the treebank. In CALIMA 0.5, only the انا AnA form is present, with the consequence that the existing أنا >anA forms in the treebank are not a match with CALIMA 0.5.

This merely points out that this is an on-going process, and another round of treebank/CALIMA synchronization is needed.

It is important to note that the CALIMA system discussed here is a restricted version of CALIMA, where only Egyptian Arabic is present. However, there are richer CALIMA versions where SAMA and CALIMA are combined together (CALIMA-SAMA-ADAM) to cover both Egyptian Arabic and Modern Standard Arabic (Habash et al., 2012b). The more extended version of CALIMA is used in the tools developed at Columbia University for Egyptian Arabic POS tagging and morphological disambiguation (Habash, et al., 2013;

[2] It is important to note that many mismatches in the synchronization process are due to the inconsistent insertion of sukun (no vowel) diacritic in the Egyptian Arabic (ARZ) corpora. Ignoring the differences in sukun diacritics should considerably improve the synchronization recall.

Pasha, et al., 2014). Since annotation errors and inconsistencies cannot be tolerated for training these tools, the Egyptian Arabic corpus was subjected to additional automatic processing to enforce consistency. Details of this effort are described in Eskander, et al. (2013).

5. Conclusions

This data has been treebanked and released as e-corpora (Maamouri, et al., 2012a-h), and will be published in the LDC Catalog in the near future.

In future work, we will be comparing parsing results using this Egyptian Arabic data to results obtained for MSA data such as Kulick, et al. (2006) and Green, et al. (2010).

Developing the morphological analyzer and the treebank annotation in parallel was successful, showing improvement from one segment to the next for both the analyzer and the annotation. Annotators relied increasingly on appropriate solutions provided by CALIMA, and CALIMA's coverage increased with each iteration. Throughout the project, contacts between the CALIMA team and the LDC Treebank team were crucial to solving nagging issues and meeting common goals. Collaboration on this type of challenge, where tools and resources are limited, proved to be remarkably synergistic, and opens the way to further fruitful work on Arabic dialects.

6. Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract Nos. HR0011-11-C-0145 and HR0011-12-C-0014. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

7. References

Ramy Eskander, Nizar Habash, Ann Bies, Seth Kulick, Mohamed Maamouri. (2013). Automatic Correction and Extension of Morphological Annotations. In *Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse*, pages 1–10, Sofia, Bulgaria, August 8-9, 2013.

Hassan Gadalla, Hanaa Kilany, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, Amr Shalaby, Krisjanis Karins, Everett Rowson, Robert MacIntyre, Paul Kingsbury, David Graff, and Cynthia McLemore. (1997). *CALLHOME Egyptian Arabic Transcripts*. Linguistic Data Consortium, Catalog No.: LDC97T19.

Spence Green and Christopher D. Manning. (2010). Better Arabic Parsing: Baselines, Evaluations, and Analysis. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, China.

Nizar Habash and Owen Rambow. (2006). MAGEAD: A Morphological Analyzer and Generator for the Arabic Dialects. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 681–688, Sydney, Australia.

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

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

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. (2012b). A Morphological Analyzer for Egyptian Arabic. In *NAACL-HLT 2012 Workshop on Computational Morphology and Phonology (SIGMORPHON2012),* pages 1–9, Montreal, Canada.

Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh (2013). Morphological analysis and disambiguation for dialectal Arabic. In *Proceedings of the conference of the North American Association for Computational Linguistics*, pp. 426-432, Atlanta, Georgia.

Clive Holes. (2004). *Modern Arabic: Structures, Functions, and Varieties*. Georgetown Classics in Arabic Language and Linguistics. Georgetown University Press.

Hanaa Kilany, Hassan Gadalla, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, and Cynthia McLemore. (2002). *Egyptian Colloquial Arabic Lexicon*. Linguistic Data Consortium, Catalog No.: LDC99L22.

Seth Kulick, Ryan Gabbard, and Mitchell Marcus. (2006). Parsing the Arabic Treebank: Analysis and Improvements. In *Proceedings of Treebanks and Linguistic Theories (TLT) 2006*.

Mohamed Maamouri, Ann Bies, Tim Buckwalter, Mona Diab, Nizar Habash, Owen Rambow, Dalila Tabessi. (2006). Developing and Using a Pilot Dialectal Arabic Treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006).*

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. (2012a). *Egyptian Arabic Treebank DF Part 1 V2.0*. Linguistic Data Consortium, Catalog No.: LDC2012E93.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. (2012b). *Egyptian Arabic Treebank DF Part 2 V2.0*. Linguistic Data Consortium, Catalog No.: LDC2012E98.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. (2012c). *Egyptian Arabic Treebank DF Part 3 V2.0*. Linguistic Data Consortium, Catalog No.: LDC2012E89.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. (2012d). *Egyptian Arabic Treebank DF Part 4 V2.0*. Linguistic Data Consortium, Catalog No.: LDC2012E99.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. (2012e). *Egyptian Arabic Treebank DF Part 5 V2.0*. Linguistic Data Consortium, Catalog No.: LDC2012E107.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos

Krouna, Dalila Tabassi, and Michael Ciul. (2012f). *Egyptian Arabic Treebank DF Part 6 V2.0*. Linguistic Data Consortium, Catalog No.: LDC2012E125.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. (2012g). *Egyptian Arabic Treebank DF Part 7 V2.0*. Linguistic Data Consortium, Catalog No.: LDC2013E12.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. (2012h). *Egyptian Arabic Treebank DF Part 8 V2.0*. Linguistic Data Consortium, Catalog No.: LDC2013E21.

Mohamed Maamouri, Sondos Krouna, Dalila Tabessi, Nadia Hamrouni, and Nizar Habash. (2012i). *Egyptian Arabic Morphological Annotation Guidelines*. Linguistic Data Consortium.

Mohamed Maamouri, David Graff, Basma Bouziri, Sondos Krouna, Ann Bies, Seth Kulick. (2010). *Standard Arabic Morphological Analyzer (SAMA) Version 3.1*. Linguistic Data Consortium, Catalog No.: LDC2010L01.

Arfath Pasha, Mohamed Al-Badrashiny, Ahmed El Kholy, Ramy Eskander, Mona Diab, 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 9th International Conference on Language Resources and Evaluation*, Reykjavik, Iceland.