Automatic Correction and Extension of Morphological Annotations

Ramy Eskander, Nizar Habash
Center for Computational Learning Systems, Columbia University
{reskander,habash}@ccls.columbia.edu

Ann Bies, Seth Kulick, Mohamed Maamouri
Linguistic Data Consortium, University of Pennsylvania
{bies,skulick,maamouri}@ldc.upenn.edu

Abstract

For languages with complex morphologies, limited resources and tools, and/or lack of standard grammars, developing annotated resources can be a challenging task. Annotated resources developed under time/money constraints for such languages tend to tradeoff depth of representation with degree of noise. We present two methods for automatic correction and extension of morphological annotations, and demonstrate their success on three divergent Egyptian Arabic corpora.

1 Introduction

Annotated corpora are essential for most research in natural language processing (NLP). For example, the development of treebanks, such as the Penn Treebank and the Penn Arabic Treebank, has been essential in pushing research on part-of-speech (POS) tagging and parsing of English and Arabic (Marcus et al., 1993; Maamouri et al., 2004). The creation of such resources tends to be quite expensive and time consuming: guidelines need to be developed, annotators hired, trained, and regularly evaluated for quality control. For languages with complex morphologies, limited resources and tools, and/or lack of standard grammars, such as any of the Dialectal Arabic (DA) varieties, developing annotated resources can be a challenging task. As a result, annotated resources developed under time/money constraints for such languages tend to tradeoff depth of representation with degree of noise. In the extremes, we find rich morphological representations that may be noisy and inconsistent or simple by highly consistent and reliable annotations that have limited usability. Furthermore, such resources are often developed by different research groups leading to many inconstancies that make pooling these resources not a very easy task.

In this paper, we describe two general techniques to address the limitations of the two types of annotations: corrections of rich noisy annotations and extensions of clean but shallow ones. We present our work on Egyptian Arabic, an important Arabic dialect with limited resources, and rich and ambiguous morphology. Resulting from this effort is the largest Egyptian Arabic corpus annotated in one common representation by pooling resources from three very different sources: a non-final, pre-release version of the ARZ[1] corpora from the Linguistic Data Consortium (LDC) (Maamouri et al., 2012g), the LDC's CallHome Egypt transcripts (Gadalla et al., 1997) and CMU's Egyptian Arabic corpus (CMUEAC) (Mohamed et al., 2012).

Although the paper focuses on Arabic, the basic problem is relevant to other languages, especially spontaneously written colloquial language forms such as those used in social media. The general solutions we propose are language independent given availability of specific language resources.

Next we discuss some related work and relevant linguistic facts (Sections 2 and 3, respectively). Section 4 presents our annotation correction technique; and Section 5 presents out annotation extension technique. Finally, Section 6 presents some statistics on the Egyptian Arabic corpus annotated in one unified representation resulting from our correction and extension work.

[1] ARZ is the language code for Egyptian Arabic, http://www-01.sil.org/iso639-3/documentation.asp?id=arz

2 Related Work

Much work has been done on automatic spelling correction. Both supervised and unsupervised approaches have been used employing a variety of tools, resources, and heuristics, e.g., morphological analyzers, language models, annotated data and edit-distance measures, respectively (Kukich, 1992; Oflazer, 1996; Shaalan et al., 2003; Hassan et al., 2008; Kolak and Resnik, 2002; Magdy and Darwish, 2006). Our work is different from these approaches in that it extends beyond spelling of word forms to deeper annotations. However, we use some of these techniques to correct not just the words, but also malformed POS tags.

A number of efforts exist on treebank enrichment for many languages including Arabic (Palmer et al., 2008; Hovy et al., 2006; Alkuhlani and Habash, 2011; Alkuhlani et al., 2013). Our morphological extension effort is similar to Alkuhlani et al. (2013)'s work except that they start with tokenizations, reduced POS tags and dependency trees and extend them to full morphological information.

There has been a lot of work on Arabic POS tagging and morphological disambiguation (Habash and Rambow, 2005; Smith et al., 2005; Hajič et al., 2005; Habash, 2010; Habash et al., 2013). The work by Habash et al. (2013) uses one of the resources we improve on in this paper. In their work, they simply attempt to "synchronize" unknown/malformed annotations with the morphological analyzer they use, thus forcing a reading on the word to make the unknown/malformed annotation usable. In our work, we address the cleaning issue directly. We intend to make these automatic corrections and extensions available in the future so that they can be used in future disambiguation tools.

Maamouri et al. (2009) described a set of manual and automatic techniques used to improve on the quality of the Penn Arabic Treebank. Their work is most similar to ours except in the following aspects: we work only on morphology and for dialectal Arabic, whereas their work is primarily on syntax and standard Arabic. Furthermore, the challenge of malformed tags is not a major problem for them, while it is a core problem for us. Furthermore, we work with data that has partial annotations that we extend, while their work was for very rich syntax/morphology annotations.

3 Linguistic Facts

The Arabic language is a collection of variants, most prominent amongst which is Modern Standard Arabic (MSA), the official language of the media and education. The other variants, the Arabic dialects, are the day-to-day native vernaculars spoken in the Arab World. While MSA is the official language, it is not the native language of any modern day Arabic speakers. Their differences from MSA are comparable to the differences between Romance languages and Latin.[2]

Egyptian Arabic poses many challenges for NLP. Arabic in general is a morphologically complex language which includes rich inflectional morphology, expressed both templatically and affixationally, and several classes of attachable clitics. For example, the Egyptian Arabic word وهيكتبوها *wi+ha+yi-ktib-uw+hA*[3] 'and they will write it' has two proclitics (+و *wi+* 'and' and +ه *ha+* 'will'), one prefix -ي *yi-* '3rd person', one suffix -و *-uw* 'masculine plural' and one pronominal enclitic +ها *+hA* 'it/her'. The word is considered an inflected form of the lemma *katab* 'write [lit. he wrote]'. An important challenge for NLP work on dialectal Arabic in general is the lack of an orthographic standard. Egyptian Arabic writers are often inconsistent even in their own writing (Habash et al., 2012a), e.g., the future particle ح *Ha* appears as a separate word or as a proclitic +ه/+ح *Ha+*/*ha+*, reflecting different pronunciations. Arabic orthography in general drops diacritical marks that mark short vowels and gemination. However in analyses, we want these diacritics to be indicated. Moreover, some letters in Arabic (in general) are often spelled inconsistently which leads to an increase in both sparsity (multiple forms of the same word) and ambiguity (same form corresponding to multiple words), e.g., variants of Hamzated Alif, أ Â or إ Ă, are often written without their Hamza (ء '): ا *A*; and the Alif-Maqsura (or dotless Ya) ى ý and the regular dotted Ya ي *y* are often used interchangeably in word final position (El Kholy and Habash, 2010). For the

[2] Habash and Rambow (2006) reported that a state-of-the-art MSA morphological analyzer has only 60% coverage of Levantine Arabic verb forms.

[3] Arabic orthographic transliteration is presented in the Habash-Soudi-Buckwalter scheme (Habash et al., 2007):
ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي
A b t θ j H x d ð r z s š S D T Ď ς γ f q k l m n h w y
in addition to ' ء, Â أ, Ă إ, Ā آ, ŵ ؤ, ŷ ئ, ħ ة, ý ى.

purposes of normalizing the representations used in computational models, we follow the work of Habash et al. (2012a) who devised a *conventional orthography for dialectal Arabic* (CODA) for use in computational processing of Arabic dialects..

An analysis of an Egyptian word for our work consists of a surface form that may not be in CODA (henceforth, RAW), a fully diacritized CODA form (henceforth, DIAC), a morpheme split form (henceforth, MORPH), which may slightly differ from the allomorphic DIAC surface forms, a POS tag for each morpheme and stem, and a lemma (henceforth LEM). For instance, the Egyptian Arabic example used above has the following analysis:

RAW	*whyktbuwhA*
DIAC	*wiHayiktibuwhA*
MORPH	*wi+Ha+yi+ktib+uwA+hA*
POS	CONJ+FUT_PART+IV3P+IV +IVSUFF_SUBJ:3P+IVSUFF_DO:3FS
LEM	*katab*

The morphological analyzers we use in the paper, CALIMA (Habash et al., 2012b) and SAMA (Graff et al., 2009), both generate the different levels of representation discussed above.

4 Automatic Morphological Correction

In this section, we present the effort on automatic morphological correction of rich noisy annotations. We next describe the data set we work with and the problems it has. This is followed by a discussion of our approach and results including an error analysis.

4.1 Data

We use a non-final, pre-release version of six manually annotated Egyptian Arabic corpora developed by the LDC, and labeled as "ARZ", parts one through six. The published versions of these corpora (Maamouri et al., 2012a-f) do not include the annotation errors discussed in this paper. Rather, in the official releases of the data from the LDC, such problematic cases with an unknown POS tag sequence (as in the example at the end of Section 4.2) were caught and given a NO_FUNC POS tag instead, in order to allow syntactic annotation of the data to proceed, and in order to meet data publication deadlines. The combined corpus consists of about 274K words. The annotations are very detailed contextually selected morphological analyses that include for each RAW word its LEM, POS, MORPH and DIAC as described earlier. The LDC used the CALIMA[4] Egyptian Arabic morphological analyzer (Habash et al., 2012b) to provide the annotators with sets of analyses to select from.[5] CALIMA's non-lexical morphological coverage (i.e. model of affixes and stem POS combinations) is almost complete; and its lexical entries are of high precision. However, CALIMA lacks some lexical items, i.e., its lexical recall is not perfect – Habash et al. (2012b) report coverage of 84% for basic CALIMA and 92% for CALIMA extended with SAMA (Graff et al., 2009) (henceforth, CALIMA+SAMA or simply the analyzer).[6] Many missing entries are a result of spelling variants that are not modeled in CALIMA. In cases when CALIMA fails to provide analyses or the annotators disagree with all the provided analyses, the annotators enter the information manually or copy and modify CALIMA provided analyses, which sometimes introduces errors.

For the purpose of this work, we consider all analyses in the corpus that are in the CALIMA+SAMA morphological analyzer to be correct. We will not attempt to modify them. Almost 30% of the corpus analyses are *not* in the analyzer, i.e. analyzer out-of-vocabulary (OOV). We discuss next the general patterns of these analyses. We refer to the original corpus analyses as the "Baseline" analyses.

4.2 Patterns of OOV Analyses in Baseline

About 3.3% of all OOV analyses (and 1% of all corpus words) are tagged as TYPOs.[7] We do not address these cases in this paper.

Over half of the POS OOVs (56%) in the pre-release data involve a different category of a nominal (NOUN/NOUN_PROP/ADJ). This is a well known issue even in MSA. The rest of the cases involve incorrect feature combinations such as giving the unaccusative verb اتنفذ *Aitnaf∼iđ* 'be performed' the POS PV_PASS (passive perfective).[8] Another example is assigning the feminine singular pronoun دي *diy* the

[4]Columbia Arabic Language and dIalect Morphological Analyzer

[5]SAMA, the Standard Arabic Morphological Analyzer (Graff et al., 2009), was used to provide the annotators with analyses for the MSA tokens.

[6]In our work, we distinguish between morphological analysis, which refers to producing the various readings of a word out of context, and morphological tagging (or disambiguation), which identifies the appropriate analysis in context.

[7]The rate of TYPO words in the ARZ data is almost 18 times the rate in the MSA PATB data sets.

[8]The inflected verb *Aitnaf∼iđ* is the passive voice of the verb with the lemma *naf∼ađ* or the active voice of the verb with the lemma *Aitnaf∼iđ*.

POS DEM_PRON instead of DEM_PRON_FS. Or the imperative verb الغوا *AilguwA* 'cancel [you plural]' the POS CV+CVSUFF_SUBJ:2MS (for 'you masculine singular') instead of the correct CV+CVSUFF_SUBJ:2MP. A tiny percentage of all POS tags in the corpus (2%) include case-related variation (e.g. CONJ vs Conj); these add to type sparsity, but are trivial to handle.

Among LEMs and DIACs, there is considerable variation in the Arabic spelling, particularly involving the spelling of Alif/Hamza forms, the Egyptian long vowels /e:/ and /o:/ and often requiring adjustment to conform to CODA guidelines.[9] The following are some examples. Specific CODA cases include spelling كده *kidah* 'as such' as كدا *kdA* or spelling قوي *qawiy* [pronounced /awi/] 'very' as اوي *Awy*. The preposition فيه *fiyh* 'in it' is incorrectly spelled as *fiyuh* (allomorphic form is incorrect). The word بيت *bayt* 'house' is spelled *biyt* (long vowel spelling error). And finally the interjection لأ *lÂ* 'no!' is spelled as (the implausible form) لء *la'*.

Among LEMs, over 63% of the errors is due to inconsistency in assigning lemmas of punctuation and digit, a trivial challenge. 29% of the cases are spelling errors such as those discussed above. The remaining 10% are due to not following the specific format guidelines of lemmas (e.g., must be singular, uncliticized, and with a sense id number). Among DIACs, almost all of the mismatches are non-CODA-compliant spelling variations. One third is Alif/Hamza forms, and another quarter is long vowel spelling. One eighth involves diacritic choice.

Combinations of these error types occur, of course. One extreme case is the progressive particle prefix *bi*, which should be tagged as bi/PROG_PART, but appears additionally as b/PROG_PART, ba/PROG_PART, bi/PART_PROG, bi/PRO_PART, and bi/FUT_PART.

Example For the rest of this section, we consider the example word حيأجلوا *HyÂjlwA* 'and they will postpone'. Figure 1 contrasts an erroneous analysis in the pre-release data with a corrected version of it. There are multiple problems in this example. First, the POS tag is both internally inconsistent and is inconsistent with the MORPH choice. The POS has a singular subject prefix (IV3MS) and a plural subject suffix (IV-SUFF_SUBJ:P); and the plural subject suffix is written using the morpheme (*+uh*), which corresponds to a direct object enclitic. The two morphemes, *+uh* and *+uwA*, are homophonous, which is the most likely cause for this error. Second, the future marker (*Ha+*) is written in a non-CODA-compliant way (*ha+*) in the analysis. And finally, the lemma is malformed, containing multiple extra sense id digits. It is important to point out that there are multiple ways to correct the analysis. For example, it can be *Ha+yi+Âaj∼il+uh* FUT_PART+IV3MS+IV+IVSUFF_DO:3MS 'he will postpone it'.[10]

4.3 Approach

Our target is to provide correct morphological analyses for the OOV annotations in the pre-release version of the ARZ corpus. Since not all of the OOV annotations are wrong in principle, we do not force map them all to CALIMA+SAMA in-vocabulary variants, especially for open class categories, where we know CALIMA+SAMA may be deficient. As such, our general solution focuses on correcting closed classes (some stems and all of the affixes) by mapping them to in-vocabulary variants. We also use a set of language-specific preprocessing corrections for common orthographic variations (for all open and closed classes). An important tool we use throughout to rank choices and break ties is modified Levenshtein edit distance.[11]

Next, we present the four steps of our correction process: annotation preprocessing, morpheme-POS correction, lemma correction and surface DIAC generation.

Annotation Preprocessing When first reading the pre-release annotations, we perform a preprocessing step that includes a set of deterministic

[9] LDC annotators were not asked to comply with CODA guidelines during the annotation task. Therefore, multiple spelling variants for OOV Egyptian Arabic words were to be expected.

[10] Since our approach currently considers words out of context, such a correction is not preferred because it requires more character edits (see Figure 2). We acknowledge this to be a limitation and plan to address it in the future.

[11] The Levenshtein edit distance is defined as the minimum number of single-character edits (insertion, deletion and substitution) required to change one string into the other. For Arabic words and morphemes, we modify the cost of substitutions involving two phonologically or orthographically similar letters to count as half edits. We acquire the list of such letter substitutions from Eskander et al. (2013), who report them as the most frequent source of errors in Egyptian Arabic orthography. We map all diacritic-only morphemes to empty morphemes in both ways at a cost of half edit also. For POS tag edit distance, we use the standard definition of Levenshtein edit distance. Edit cost is an area where a lot of tuning could be done and we plan to explore it in the future.

RAW	هياجلو *hyAjlw*	
Analysis	**Incorrect Annotation**	**Correct Annotation**
DIAC	*hayiÂaj∼iluh*	*HayiÂaj∼iluwA*
MORPH	*ha+yi+Âaj∼il+uh*	*Ha+yi+Âaj∼il+uwA*
POS	FUT_PART+IV3MS+IV+IVSUFF_SUBJ:P	FUT_PART+IV3P+IV+IVSUFF_SUBJ:P
LEM	*Âaj∼ill1*	*Âaj∼il_1*

Figure 1: An incorrect annotation example with a possible correction.

corrections for common non-CODA-compliant orthographic variations and errors, and POS tagging typos. The corrections apply to the POS tags, lemmas, morphemes and surface forms. Examples of these corrections include the following: reordering diacritics, e.g., *saji∼l* → *saj∼il*; removing duplicate diacritics, e.g., *saj∼iil* → *saj∼il*; adjusting Alif-Hamza forms to match the diacritics that follow them, e.g., *ĂaSl* → *ÂaSl*; and POS tag capitalization, e.g., Fut_Part → FUT_PART.

Morpheme-POS Correction For morpheme correction purposes, we define an abstract representation that combines all the closed-class morphemes and POS tags. For open-class stems, we simply use the POS tag. For example, the abstract morpheme representation for the correct version of the word in Figure 1 is *Ha/FUT_PART+yi/IV3P+IV+uwA/IVSUFF_SUBJ:P*. We will refer to this representation as the inflectional morph-tag (IMT).

We build two models for this task. First, we build an IMT language model from the CALIMA+SAMA databases. This models all possible inflections in the analyzer without the open class stems. This model includes 304K sequences. Second, we construct a map from all the seen IMTs in the ARZ corpus to all the in-vocabulary IMTs in the IMT language model. The mapping includes a cost that is based on the edit distance discussed earlier. Figure 2 shows the top mappings for the IMTs in our example. Both models are implemented as finite state machines using the ATT FSM toolkit (Mohri et al., 1998).

The input, possibly incorrect, IMT is converted into an FSM that is then composed with the mapping transducer and the language model automaton to generate a cost-ranked list of mappings. The output for our example is listed in Figure 3. We then replace the input POS and MORPH with the top ranked correction: *Ha/FUT_PART+yi/IV3MS+IV+uh/IVSUFF_SUBJ:P* at a cost of 4.0. The open class stem is not modified.

Base IMT Morpheme	**Mapped IMT Morphemes**	**Cost**
ha/FUT_PART	Ha/FUT_PART	0.5
	sa/FUT_PART	1.0
yi/IV3MS	yi/IV3MS	0.0
	ya/IV3MS	1.0
	y/IV3MS	1.0
	yu/IV3MS	1.0
	yi/IV3P	2.0
IV	IV	0.0
	PV	1.0
	CV	1.0
uh/IVSUFF_SUBJ:P	uwA/IVSUFF_SUBJ:P	1.5
	na/IVSUFF_SUBJ:FP	3.0

Figure 2: Top mappings for the IMT morphemes ha/FUT_PART, yi/IV3P, IV and uh/IVSUFF_SUBJ:P

Input: *ha/FUT_PART+yi/IV3P+IV+uh/IVSUFF_SUBJ:P*	
FSM Output	**Cost**
Ha/FUT_PART+yi/IV3P+IV+uwA/IVSUFF_SUBJ:P	4.0
Ha/FUT_PART+y/IV3P+IV+uwA/IVSUFF_SUBJ:P	5.0
Ha/FUT_PART+ti/IV2P+IV+uwA/IVSUFF_SUBJ:P	6.0
Ha/FUT_PART+yi/IV3MS+IV+uh/IVSUFF_DO:3MS	6.5
Ha/FUT_PART+yi/IV3MS+IV+kuw/IVSUFF_DO:2P	7.0
Ha/FUT_PART+yi/IV3MS+IV+nA/IVSUFF_DO:1P	7.0
Ha/FUT_PART+tu/IV2P+IV+uwA/IVSUFF_SUBJ:P	7.0
sa/FUT_PART+ya/IV3FP+IV+na/IVSUFF_SUBJ:FP	7.0
sa/FUT_PART+yu/IV3FP+IV+na/IVSUFF_SUBJ:FP	7.0
Ha/FUT_PART+yi/IV3MS+IV+kum/IVSUFF_DO:2P	7.5

Figure 3: Top corrections for the input ha/FUT_PART+yi/IV3P+IV+uh/IVSUFF_SUBJ:P

Lemma Correction We generate a map that includes all the possible lemmas for every possible stem morpheme in CALIMA+SAMA. For a given ARZ word analysis, if the stem morpheme is in CALIMA+SAMA, then we pick the lemma from its corresponding lemma set. When there is more than one possible lemma, we pick the lemma that is closest to the provided pre-release ARZ lemma, based on their string edit distance as defined earlier. If the stem morpheme is not in CALIMA+SAMA (e.g., open class), then we keep the ARZ lemma as it is.

In our example, the stem morpheme *Âaj∼il/IV* is paired in CALIMA+SAMA with the lemma *Âaj∼il_1*. Accordingly, *Âaj∼il_1* replaces the in-

put pre-release ARZ lemma.

Surface DIAC Generation After correcting the morphemes and POS tags in the input word, we use them to generate a new surface DIAC form. For all the closed-class morphemes and in-vocabulary open-class stems, we use CALIMA+SAMA to identify all the MORPH+POS to DIAC mappings. For open-class stems that are OOVs, we use their corresponding DIAC form in the input word.[12] This may lead to many possible sequences. We rank them by their edit distance (defined above) to the surface DIAC of the input word.

In our example, this process is rather trivial: every morpheme is paired with only one surface DIAC in the morphological analyzer. The surface DIACs corresponding to *Ha/FUT_PART*, *yi/IV3P*, *Âaj∼il/IV* and *uwA/IVSUFF_SUBJ:P* are *Ha*, *yi*, *Âaj∼il* and *uwA*, respectively. The final combined surface is *HayiÂaj∼iluwA*.

A more interesting example is the word علينا *ʕalay+nA* 'upon us' which has the analysis *ʕalaý*/PREP+*nA*/PRON_1P. The MORPH stem *ʕalaý* has two DIAC forms: *ʕalaý* and *ʕalay*. The second form is only used when an enclitic is present. It is selected in this example because it has a smaller edit distance to the full word input DIAC form than the surface stem *ʕalaý*. In the future, we plan to use more sophisticated generation and detokenization techniques (El Kholy and Habash, 2010).

4.4 Results and Error Analysis

Results We conducted a manual evaluation for 1,000 words from the internal, pre-release ARZ after applying the automatic correction process. This set is a blind test set, i.e., not used as part of the development. The results are listed in Table 1 for the lemmas, POS tags, diacritized morphemes and diacritized surface forms, in addition to the complete morphological analyses (token-based), where the correction output is compared to the pre-release ARZ annotations (the baseline).

The results are listed for different subsets of the data. The first row lists the results considering the complete 1,000 words, where all the in-vocabulary words are considered correct. This is only intended to give an overall estimate of the correctness of the set. The second row lists the results for CALIMA+SAMA OOV words only. The third row is the same as the second, but excluding punctuations, digits and typos. Focusing on the last row, we see that we achieve between 58% and 24% error reduction on different features, and reach almost 40% error reduction on all features combined.

Error Analysis For POS, 99.7% of all the correct cases in the Baseline were not changed. Only one case was changed and it was caused by an error in the input MORPH splits. Of the erroneous cases in the Baseline, 40% were not changed. Among the attempted changes, 71% successfully fixed the baseline problem. Almost all of the failed changes are due to implausible null pronouns in the Baseline that were not handled in the current implementation, which only considered correct null pronouns. We plan to address these in the future. Among the errors that were not addressed, the most common case involves nominal form (41%) followed by hard features to resolve and open class passive-voice inconsistency (each 27%).

Regarding lemmas, 93.9% of all correct baseline lemmas remained correct. In the rest, over-correction attempts resulting from matching the OOV lemma to the wrong in-vocabulary lemma backfired. Around 13.9% of the erroneous baseline lemmas were not modified and 1.5% were modified incorrectly. The rest, 84.6%, were successfully fixed. Almost all of the system errors resulting from changes involve over correction by mapping to incorrect INV lemma forms.

Finally, as for diacritized forms, 96.9% of the correct baseline DIACs remained correct; the rest fell victim to over-correction. Among incorrect baseline cases, 43% remained unchanged; and 45% were fixed; 4% were over-corrected and 8% only partially corrected. Remaining DIAC errors are mostly in open classes where the analyzer recall problems cannot help.

5 Automatic Morphological Extension

In this section, we present the general technique we use to extend shallow annotations. We discuss the data sets, the approach and evaluation results next.

5.1 Data

We conduct our experiments on two different Egyptian Arabic corpora: the CALLHOME Egypt (CHE) corpus (Gadalla et al., 1997) and Carnegie Mellon University Egyptian Arabic corpus (CMUEAC) (Mohamed et al., 2012).

[12] Since the surface DIAC splits are not provided, we determine the exact boundary of the surface DIAC stem by minimizing the edit distance between the prefixing/suffixing morphemes and the full input surface DIAC form.

		LEM	POS	MORPH	DIAC	POS +MORPH	**All**
All words	Baseline	79.8%	93.2%	92.2%	91.1%	87.3%	72.7%
	System	95.7%	95.5%	93.8%	93.6%	91.5%	90.0%
Analyzer OOV	Baseline	47.1%	82.4%	79.7%	76.8%	66.8%	28.4%
	System	88.9%	88.42%	83.9%	83.4%	77.9%	73.9%
Analyzer OOV, no Punc/Digit/Typos	Baseline	71.3%	82.5%	74.1%	69.7%	59.0%	43.0%
	System	88.0%	87.3%	80.5%	79.7%	71.3%	65.3%

Table 1: Accuracy of the automatic morphological correction of internal, pre-release ARZ data.

CHE The CHE corpus contains 140 telephone conversation transcripts of about 179K words. Each word is represented by its phonological form and undiacritized Arabic script orthography. The orthography used is quite similar to the CODA standard we use. Being a transcript corpus, it is quite clean and free of spelling variations. We use a technique described in more detail in Habash et al. (2012b) to combine the phonological form and undiacritized Arabic script into diacritized Arabic script, i.e. DIAC. For example, the undiacritized word عينه *ςynh* 'his eye' is combined with its pronunciation /ςe:nu/ producing the diacritized form *ςaynuh*.

CMUEAC The CMUEAC corpus includes about 23K words that are only annotated for morph splits. The corpus text includes spontaneously written Egyptian Arabic text collected off the web. To use the same example as above, the word عينه *ςynh* 'his eye' is segmented as *ςyn+h* indicating that there is a base word plus an enclitic.

5.2 Approach

Our approach to morphological extension is to automatically annotate the corpus using a very rich morphological tagger, and then use the limited manual annotations to adjust the morphological choice. We use a morphological tagger, MADA-ARZ (Morphological Analysis and Disambiguation for Egyptian Arabic) (Habash et al., 2013). MADA-ARZ produces, for each input word, a contextually ranked list of analyses specifying all the morphological interpretations of that word as provided by the CALIMA+SAMA morphological analyzer.

CHE In the case of CHE, we select the first choice from the ranked list of analyses whose DIAC matches the diacritized word in CHE. For example, for the word عينه *ςynh* MADA-ARZ generates 45 different morphological analyses with different lemmas, POS, orthographies and diacritics: *ςayn+uh* 'his eye', *ςay∼in+aħ* 'sample' and *ςay∼in+uh* 'he appointed him'. The diacritized word *ςayn+uh* allows us to select the following full analysis:

RAW	*Eynh*
DIAC	*Eaynuh*
MORPH	*Eayn+uh*
POS	NOUN+POSS_PRON_3MS
LEM	*Eayn_1*

Although this example may not require the full power of a tagger, but just the out-of-context analyzer, other cases involving POS ambiguity unrealized through diacritization necessitate the use of a tagger, e.g., the word كاتب *kAtib* can be an ADJ meaning 'writing' or a NOUN meaning 'writer/author'.

CMUEAC In the case of CMUEAC, we select the first choice from the ranked list of analyses whose undiacritized MORPH splits match the word tokenization. In the case of the word عينه *ςyn+h*, the tokenization cannot distinguish between the noun reading *ςayn+uh* 'his eye' and the verbal reading *ςay∼in+uh* 'he appointed him'. MADA-ARZ effectively selects in such cases. We expect the performance on CMUEAC to be worse than CHE given the difference in the amount of information between the two corpora.

5.3 Results and Error Analysis

We evaluate the accuracy of the morphological extension process on both CHE and CMUEAC using two 300 word samples that were manually enriched. Table 2 presents the accuracies of the assigned LEMs, POS tags, DIAC forms and MORPHs, in addition to the complete morphological analysis. All results are token-based.

CHE CHE analyses have high accuracies ranging between 95.2% and 97.2% for the different analysis features, with the complete analysis having an accuracy of 92.8%.

Metric	CHE	CMUEAC
LEM	97.2	82.0
POS	95.2	79.6
MORPH	96.8	77.6
DIAC	97.2	78.4
POS+MORPH	92.8	74.0
All	92.8	72.0

Table 2: Accuracy of automatic morphological extension of CHE and CMUEAC.

28% of the errors are due to wrong verbal features (person, number and gender) for forms that are not distinguishable in DIAC, e.g., كتبت *katabt* 'I/you wrote' and تكتب *tiktib* 'you write/she writes'. One fifth of the errors is due to gold diacritization errors in the CHE corpus. while 11% of the errors are because of failure in assigining the correct diacritization. The rest of the errors are because of failure in assigning the correct POS tags for nouns, particles and verbs with percentages of 22%, 11% and 6%, respectively.

CMUEAC CMUEAC analyses have much lower accuracies compared to CHE, ranging between 77.6% and 82.0% for different features, with the complete analysis accuracy at 72.0%. The CMUEAC is much harder to extend for two reasons: the text, being naturally occurring, contains a lot of orthographic noise; and tokenization information is not sufficient to disambiguate many analyses. For CMUEAC, a quarter of the errors is due to gold tokenization errors in the original CMUEAC corpus. Another quarter of the errors results from MADA-ARZ assigning an MSA analysis instead of an Egyptian Arabic analysis.[13] Failure to assign the correct POS tags for particles, verbs and nouns represents 14%, 10% and 7% of the errors, respectively. Other errors are because of wrong verbal features (13%) and wrong diacritization (6%).

As expected, relatively richer annotations (i.e., diacritics) are easier to extend to full morphological information that relatively poorer annotations (i.e., tokenization). Of course, the tradeoff is still there as tokenizations are much easier and cheaper to annotate. We plan to explore the question of what would be an optimal set of poor annotations that can help us extend to the full morphology at high accuracy in the future.

[13] MADA-ARZ is trained on a combination of MSA and Egyptian Arabic text and as such may select an MSA analysis in cases that are ambiguous.

6 Egyptian Corpus

After applying morphological corrections to prerelease ARZ and morphological extensions to CHE and CMUEAC, we have now three big corpora that are automatically adjusted to include the same rich morphological information, that is: lemma, POS tag, diacritized morphemes, and diacritized surface. We combine the three resources together in one morphologically rich corpus that contains about 46K sentences and 447K words, representing 61K unique lemmas. We intend to make these automatic corrections and extensions available in the future to provide extensive support for Egyptian Arabic processing for different purposes.

7 Conclusion and Future Work

We presented two methods for automatic correction and extension of morphological annotations and demonstrated their success on three different Egyptian Arabic corpora, which now have annotations that are automatically adjusted to include the same rich morphological information although at different degrees of quality that correspond to the amount of initial information.

In the future, we plan to study how to optimize the amount of basic information to annotate manually in order to maximize the benefit of automatic extensions. We also plan to provide feedback to the annotation process to reduce the percentage of errors generated by the annotators, perhaps through a tighter integration of the correction/extension techniques with the annotation process. We also plan on using the cleaned up corpus to extend the existing analyzer for Egyptian Arabic.

Acknowledgment

This paper is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under contracts No. HR0011-12-C-0014 and HR0011-11-C-0145. 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 also would like to thank Emad Mohamed and Kemal Oflazer for providing us with the CMUEAC corpus. We thank Ryan Roth for help with MADA-ARZ. Finally, we thank Owen Rambow, Mona Diab and Warren Churchill for helpful discussions.

References

Sarah Alkuhlani and Nizar Habash. 2011. A Corpus for Modeling Morpho-Syntactic Agreement in Arabic: Gender, Number and Rationality. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL'11)*, Portland, Oregon, USA.

Sarah Alkuhlani, Nizar Habash, and Ryan Roth. 2013. Automatic morphological enrichment of a morphologically underspecified treebank. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 460–470, Atlanta, Georgia, June. Association for Computational Linguistics.

Ahmed El Kholy and Nizar Habash. 2010. Techniques for Arabic Morphological Detokenization and Orthographic Denormalization. In *Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.

Ramy Eskander, Nizar Habash, Owen Rambow, and Nadi Tomeh. 2013. Processing spontaneous orthography. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 585–595, Atlanta, Georgia, June. Association for Computational Linguistics.

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. In *Linguistic Data Consortium, Philadelphia*.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

Nizar Habash and Owen 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.

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, Abdelhadi Soudi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Springer.

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, Montréal, Canada.

Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi 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.

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

Jan Hajič, Otakar Smrž, Tim Buckwalter, and Hubert Jin. 2005. Feature-based tagger of approximations of functional Arabic morphology. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*, Barcelona, Spain.

Ahmed Hassan, Sara Noeman, and Hany Hassan. 2008. Language Independent Text Correction using Finite State Automata. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP 2008)*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% Solution. In *NAACL '06: Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers on XX*, pages 57–60, Morristown, NJ, USA.

Okan Kolak and Philip Resnik. 2002. OCR error correction using a noisy channel model. In *Proceedings of the second international conference on Human Language Technology Research*.

Karen Kukich. 1992. Techniques for Automatically Correcting Words in Text. *ACM Computing Surveys*, 24(4).

Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. 2004. The Penn Arabic Treebank: Building a Large-Scale Annotated Arabic Corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, pages 102–109, Cairo, Egypt.

Mohamed Maamouri, Ann Bies, and Seth Kulick. 2009. Creating a methodology for large-scale correction of treebank annotation: The case of the Arabic treebank. In *MEDAR Second International Conference on Arabic Language Resources and Tools, Egypt*. Citeseer.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. 2012a. Egyptian Arabic Treebank DF Part 1 V2.0. LDC catalog number LDC2012E93.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. 2012b. Egyptian Arabic Treebank DF Part 2 V2.0. LDC catalog number LDC2012E98.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. 2012c. Egyptian Arabic Treebank DF Part 3 V2.0. LDC catalog number LDC2012E89.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. 2012d. Egyptian Arabic Treebank DF Part 4 V2.0. LDC catalog number LDC2012E99.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. 2012e. Egyptian Arabic Treebank DF Part 5 V2.0. LDC catalog number LDC2012E107.

Mohamed Maamouri, Ann Bies, Seth Kulick, Sondos Krouna, Dalila Tabassi, and Michael Ciul. 2012f. Egyptian Arabic Treebank DF Part 6 V2.0. LDC catalog number LDC2012E125.

Mohamed Maamouri, Sondos Krouna, Dalila Tabessi, Nadia Hamrouni, and Nizar Habash. 2012g. Egyptian Arabic Morphological Annotation Guidelines.

Walid Magdy and Kareem Darwish. 2006. Arabic OCR Error Correction Using Character Segment Correction, Language Modeling, and Shallow Morphology. In *Proceedings of 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 408–414, Sydney, Austrailia.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19(2):313–330, June.

Emad Mohamed, Behrang Mohit, and Kemal Oflazer. 2012. Annotating and Learning Morphological Segmentation of Egyptian Colloquial Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Istanbul.

Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. 1998. A rational design for a weighted finite-state transducer library. In D. Wood and S. Yu, editors, *Automata Implementation*, Lecture Notes in Computer Science 1436, pages 144–58. Springer.

Kemal Oflazer. 1996. Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction. *Computational Linguistics*, 22:73–90.

Martha Palmer, Olga Babko-Malaya, Ann Bies, Mona Diab, Mohamed Maamouri, Aous Mansouri, and Wajdi Zaghouani. 2008. A Pilot Arabic Propbank. In *Proceedings of LREC*, Marrakech, Morocco, May.

Khaled Shaalan, Amin Allam, and Abdallah Gomah. 2003. Towards Automatic Spell Checking for Arabic. In *Conference on Language Engineering, ELSE*, Cairo, Egypt.

Noah Smith, David Smith, and Roy Tromble. 2005. Context-Based Morphological Disambiguation with Random Fields. In *Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP05)*, pages 475–482, Vancouver, Canada.