Creating Resources for Dialectal Arabic from a Single Annotation: A Case Study on Egyptian and Levantine

Ramy Eskander, Nizar Habash†, Owen Rambow and Arfath Pasha
Columbia University, USA
†New York University Abu Dhabi, UAE
rnd2110@columbia.edu, nizar.habash@nyu.edu
rambow@ccls.columbia.edu, arfath@ccls.columbia.edu

Abstract

Arabic dialects present a special problem for natural language processing because there are few Arabic dialect resources, they have no standard orthography, and they have not been studied much. However, as more and more written dialectal Arabic is found on social media, natural language processing for Arabic dialects has become an important goal. We present a methodology for creating a morphological analyzer and a morphological tagger for dialectal Arabic, and we illustrate it on Egyptian and Levantine Arabic. To our knowledge, these are the first analyzer and tagger for Levantine.

1 Introduction

The goal of this paper is to show how a particular type of annotated corpus can be used to create a morphological analyzer and a morphological tagger for a dialect of Arabic, without using any additional resources. A morphological analyzer is a tool that returns all possible morphological analyses for a given input word taken out of any context. A morphological tagger is a tool that identifies the single morphological analysis which is correct for a word given its specific context in a text. We illustrate our work using Egyptian Arabic and Levantine Arabic. Egyptian Arabic is in fact relatively resource-rich compared to other Arabic dialects, but we use a subset of available data to simulate a resource-poor dialect.

For Arabic and its dialects, we are faced with a set of well-known challenges (Habash, 2010): they have a rich morphology, the orthography encourages ambiguity, and the dialects do not have standard orthographies (and are generally not well documented linguistically). However, we can also exploit similarities among the dialects and between Dialectal Arabic (DA) and Modern Standard Arabic (MSA), and in fact the writing system, while encouraging ambiguity, also reduces the differences between these variants.

The approach we present in this paper is based on the morphological annotation of a text corpus. The annotator provides for each word a conventional orthography, a segmentation, a set of features, and a lemma. We use this information to hypothesize unseen morphological forms, and use all forms to build an analyzer for the new dialect. Because of the close relationship among the variants of Arabic, we also make use of an existing analyzer for MSA, and (in the case of Levantine), an existing Egyptian analyzer. The resulting analyzer is then used in a tagger, which uses the annotated corpus to learn classifiers that choose among all possible analyses. Crucially, we do not require the corpus to be fully annotated, allowing the annotator to concentrate on the most frequent words. We are currently annotating five more dialects with this sort of corpus, with initial corpora available for two dialects (Al-Shargi et al., 2016).

The primary contribution of this paper is that we describe a methodology for creating morphological analyzers and taggers for any Arabic dialect. We show the effectiveness of our approach by measuring performance based on training corpora of different sizes. A secondary contribution of this paper is that we present new resources for Levantine Arabic (a morphological analyzer and a morphological tagger), which to our knowledge are the first of their kinds.

While we restrict our attentions to Arabic and its dialects, we believe our approach may be relevant to other situations in which we face a large number of related low-resource languages or language variants.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3455–3465, Osaka, Japan, December 11-17 2016.

This paper is structured as follows: Section 2 presents related work. Section 3 presents the data we use. Section 4 discusses the creation of morphological analyzers, and Section 5 the morphological taggers. Section 6 evaluates the morphological analysis and tagging. Finally, Section 7 concludes and outlines future plans.

2 Related Work

There has been a considerable amount of research on MSA morphological analysis, disambiguation, part-of-speech (POS) tagging, tokenization, lemmatization and diacritization, see for example (Habash and Rambow, 2005; Zitouni et al., 2006; Diab et al., 2007; Pasha et al., 2014).

In the early efforts on DA processing, researchers focused on exploiting MSA resources and tools. Duh and Kirchhoff (2005) adopted a minimally supervised approach that requires raw data from several DAs, and an MSA morphological analyzer. They reported a POS accuracy of $\sim 71\%$ on a coarse-grained POS tagset (17 tags). Similarly, Chiang et al. (2006) were the first to attempt to do parsing on Arabic dialects using MSA training data. Other notable efforts to create dialectal morphological analyzers using MSA morphological analyzers include work reported by Abo Bakr et al. (2008) and Salloum and Habash (2011).

In the last few years, an important shift in the research on DA processing occurred, as researchers started to create resources that target DA and focused less on exploiting MSA resources. This can be seen in the rise of dialectal corpora and annotations (Gadalla et al., 1997; Diab et al., 2010; Al-Sabbagh and Girju, 2012b; Mohamed et al., 2012; Maamouri et al., 2014; Bouamor et al., 2014; Jarrar et al., 2014; Masmoudi et al., 2014; Smaïli et al., 2014; Voss et al., 2014; Khalifa et al., 2016).

In the context of morphological analysis and tagging for Arabic dialects, recent efforts focused principally on Egyptian Arabic (Mohamed et al., 2012; Al-Sabbagh and Girju, 2012a; Habash et al., 2012b; Habash et al., 2013). Mohamed et al. (2012) annotated a small corpus of Egyptian Arabic for morphological segmentation and learned tokenization models using memory-based learning (Daelemans and van den Bosch, 2005). Their system achieves a 91.90% accuracy on the task of morpheme-segmentation. Al-Sabbagh and Girju (2012a) describe a supervised tagger for Egyptian Arabic social media corpora using transformation-based learning (Brill, 1995). They report 87.6% on POS tagging. Finally, we previously created a morphological analyzer for Egyptian (Habash et al., 2012b) using the lexicon of Kilany et al. (2002); then we used this analyzer to create a morphological tagger (Habash et al., 2013). Our previous work used rich Egyptian-specific resources — the lexicon derived from the CallHome corpus for Egyptian Arabic (Kilany et al., 2002) and the Egyptian Arabic Treebank (Maamouri et al., 2014). In contrast, we now wish to explore how much can be done with a small annotation effort. In both our previous work and our new one, we use an analyze-and-choose approach to morphological tagging, following the work of Hajič (2000) (also used by Habash and Rambow (2005) for MSA). We also compare against our previous work in our evaluation.

3 Data

3.1 Orthography

Arabic dialects do not have a standard orthography. This is a big challenge to the annotation process as it allows the coexistence of uninteresting orthographic variations. To address this challenge, we previously developed the *Conventional Orthography for Dialectal Arabic* (CODA) (Habash et al., 2012a). The CODA choices aim at reducing differences between variants (DA and MSA) when possible while maintaining the distinctive morphological inventories of the different variants. The first CODA specifications were developed for Egyptian Arabic (henceforth EGY) and utilized in the EGY corpus which we also use (Maamouri et al., 2012). The EGY CODA guidelines were extended to Levantine Arabic (henceforth LEV) by the creators of the LEV corpus we use (Jarrar et al., 2014). Since the LEV corpus was annotated without diacritics, all diacritics were also stripped from the EGY corpus for the study we present in this paper. The only exception is that lemmas are represented using diacritics in the corpora for both dialects so that fine-grained distinctions between different lexemes can be made.

Word	CODA	Lemma	Gloss	BW Tag	POS	POS5	Stem
سألو sÂlw	سأله sÂlh	saÂal	ask	PV+PVSUFF_SUBJ:3MS+PVSUFF_DO:3MS	verb	VRB	sÂl
ابوه Abwh	ابوه Abwh	Ab	father	NOUN+POSS_PRON_3MS	noun	NOM	Ab
ليش lyš	ليش lyš	layš	why	INTERROG_ADV	adv_interrog	PRT	lyš
اتأخرت AtÂxrt	اتأخرت AtÂxrt	AitÂax∼ar	be late	PV+PVSUFF_SUBJ:2MS	verb	VRB	AtÂxr
لهلوكت lhlwkt	لهالوقت lhAlwqt	waqt	time	PREP+DEM_PRON+DET+NOUN	noun	NOM	wqt
؟ ?	؟ ?	?	?	PUNC	punc	PNX	?

Table 1: An example Levantine sentence سألو ابوه ليش اتأخرت لهلوكت؟ *sÂlw Abwh lyš AtÂxrt lhlwkt?* 'His father asked him why he was late?'. The various columns are for the CODA spelling and different morphological features: lemma, gloss, Buckwalter POS tag, two reduced POS tags and stem.

3.2 Egyptian Data

Corpus We use the Egyptian Arabic corpora developed by the Linguistic Data Consortium (LDC) (Maamouri et al., 2012; Eskander et al., 2013a). The corpora are morphologically annotated in a similar style to the annotations done at the LDC for MSA. Words are provided with contextually appropriate CODA form, lemmas, POS tags (Buckwalter, 2004), and English glosses.

We ran the EGY corpus through our EGY morphological analyzer CALIMA$_{\text{Egy}}$ (Habash et al., 2012b) in order to generate morphological features similar to the ones described in MADAMIRA (Pasha et al., 2014). In this process, we replaced the human-annotated corpus analysis by the closest CALIMA$_{\text{Egy}}$ analysis when it does not match any of the CALIMA$_{\text{Egy}}$ analyses for a given word. In the cases where CALIMA does not produce an analysis, a word is analyzed as a proper noun as a back-off. The back-off happened in 4.9% of the words. Finally, we removed diacritics from the corpus except for the lemmas as described above.

Data Splits We follow the corpus splits described in (Diab et al., 2013). The whole DEV (45K words) and TEST (46K words) are used, while only a portion of 135K words of TRAIN is utilized for the purpose of this work.

3.3 Levantine Data

Corpus We use the Curras Corpus of Palestinian Arabic developed at Birzeit University (Jarrar et al., 2014) as the LEV data. Palestinian Arabic is a sub-dialect of Levantine Arabic. The Corpus is around 57,000 words, half of which come from transcripts of a TV show and the rest of which comes from a mix of sources such as Facebook, Twitter, blogs and web forums.

The corpus is morphologically annotated in a similar style to the annotations in the EGY corpus. Procedurally, the developers of the Curras Corpus used MADAMIRA Egyptian (Pasha et al., 2014) to provide a starting point for the manual annotation. In this paper, we used a version of Curras that is only 82% manually annotated. Gaps in annotation exist only in the training corpus, but not in the development or test corpora, which are fully manually annotated. At the time of this publication, the Curras corpus is fully annotated.

Table 1 presents an example of a single Levantine sentence with the following associated annotations:[1] the CODA spelling, the lemma, its gloss, the full Buckwalter POS tag, two POS tags from different tagsets of Arabic and the stem. We discuss them further and evaluate against them in Section 6.

Data Splits We divided the provided corpus into three data sets; TRAIN, DEV and TEST, corresponding to 78% (45K words), 11% (6K words) and 11% (6K words) of the corpus size, respectively. DEV is selected to be the first 371 sentences of the Facebook threads in Curras, in addition to the first four documents from the TV show *Watan Aa Watar*, while TEST represents the rest of the Facebook threads and the next four documents from the TV show. The remaining part of the Curras corpus forms TRAIN.

[1] Arabic transliteration in this paper is presented in the Habash-Soudi-Buckwalter scheme (Habash et al., 2007).

4 Creating the Morphological Analyzers

We build two morphological analyzers, one for EGY and one for LEV, based on the corpus annotations. The two analyzers are built in the same manner based on completing the inflectional classes (ICs) generated from TRAIN. We follow our work on the automatic extraction of morphological lexicons from corpora to build morphological analyzers given the corpus annotations (Eskander et al., 2013b). However, the work described in (Eskander et al., 2013b) was performed only on verbs with no clitics. This would not allow us to build wide-coverage morphological analyzers. In this paper, we extend the work to cover words of any POS types, whether with clitics or without, so that we obtain complete morphological analyzers. We also apply the approach to Levantine for the first time.

For each POS, we collect all the possible morphosyntactic feature combinations found in TRAIN for words of that POS. This list of feature combinations defines the set of slots for inflected forms found in all inflectional classes (ICs) for lemmas with that POS. For each lemma in TRAIN and for all of its inflected forms found in TRAIN, we then create an inflectional class (IC) that lists the prefix, stem and suffix information in the appropriate slots in the IC. Typically, many slots remain empty for these ICs. The stem is represented as an abstraction, where the letters in the stem are replaced by placeholders, except for the letters ا *A*,أ *Â*, إ *Ǎ*, آ *Ā*, ؤ *ŵ*, ئ *ŷ*, ء *'*, و *w*, ي *y* and ى *ý*. This approach simulates in a simple manner the templatic (or "root and pattern") morphology of Semitic langauges. We then automatically complete the ICs to fill in the missing slots, and obtain all inflections of all the lexemes. Each IC in this set of complete ICs is associated with a set of compatible roots, such that each lexeme corresponds to a root and an IC. We do the completion process for each POS type separately (a total of 33 POS types), as every POS type has its own set of features with which it is compatible.

After completing the ICs, we create ALMOR databases (Habash, 2007) that represent the EGY and LEV morphological analyzers. The list of prefixes and suffixes are generated directly from the ICs. For the construction of the stems, each of the roots associated to an IC is plugged into the stem templates to generate the concrete stems. Finally, the compatibility tables are generated according to the correlation among the prefixes, stems and suffixes in the ICs.

We create morphological analyzers for different training sizes. In the case of EGY, the sizes of the analyzers are 5K, 15K, 45K and 135K, while the sizes of the LEV analyzers are 5K, 15K and 45K. Thus, we can evaluate the performance of the EGY and LEV analyzers at different sizes and compare them. Additionally, the big EGY analyzer of 135K words allows us to see the performance when more data is available.

The orthographic transformations between the input words and the surface-form annotations that appear in TRAIN are added into the analyzers as extensions. This allows the analyzers to convert a input in spontaneous orthography that is not in CODA into a CODA-compliant form that the analyzers know how to handle. The orthographic extensions are added for each of the prefix, stem and suffix entries, separately. However, a transformation that only appears once in TRAIN is omitted. This avoids having over-generating extensions that are due to infrequent typos.

If an input word is not given an analysis by the analyzer, we perform a back-off to a proper-noun analysis. In this case, the lemma, CODA, and stem are given the form of the input word.

5 Creating the Morphological Taggers

The morphological taggers were created by extending MADAMIRA for the EGY data and for LEV. MADAMIRA (Pasha et al., 2014) is a system for morphological analysis and disambiguation of Arabic text. It utilizes a morphological analysis component, a feature modeling component and an analysis ranking component to produce a list of analyses for each word in a given sentence. The morphological analyzer returns a list of all possible analyses (independent of context) for each word. The feature modeling component applies classifiers to derive predictions for the word's morphological features in context. The analysis ranking component then scores each word analysis list based on how well each analysis agrees with the model predictions, and then sorts the analyses based on that score.

For the purpose of this paper, two sets of MADAMIRA systems were developed using EGY and LEV

analyzers and classifiers built from different sizes of data sets, as described in Section 4. We note that a MADAMIRA system already exists for EGY, which uses a large amount of training data. In this paper, we do not use that system, and artificially reduce the amount of training data to simulate a resource-poor dialect.

Each system is trained on the TRAIN sets described in 3. Tuning of the individual classifiers was conducted by randomly selecting about 10% of the total word volume from TRAIN to be used as a tuning set. The tuning set was used to generate a set of feature weights that are required by the Analysis Ranking component. The tuning set was later merged back into TRAIN and the classifiers were then trained using all the training data.

6 Evaluation

6.1 Components of the Evaluation

In our evaluations for both the morphological analyzer and the tagger, we evaluate for the following components of the output of the analyzer or tagger:

- **POS** is the core POS tag of the word. We use the stem tagset in MADAMIRA, whose size is 36.
- **POS5** is a reduced tag set: *NOM* (all nominals including adjectives and adverbs), *PROP* (proper nouns), *VRB* (verbs), *PRT* (all particles), and *PNX* (punctuation). This tagset is a variant of the Columbia Arabic Treebank tagset, which is based on traditional Arabic grammar (Habash et al., 2009).
- **Lemma** is the fully diacritized lemma.
- **CODA** is the undiacritized conventional spelling of the input word with normalized Alifs (Habash et al., 2012a).
- **Stem** is the undiacritized stem of the word with normalized Alifs. The evaluation of this component represents the ability of the analyzer or tagger to segment a word into three parts, corresponding to all prefixes conjoined, a stem, and all suffixes conjoined.
- **ALL** represents the conjunction of all five preceding metrics, i.e., they all need to be correct in the same answer.

We describe the specific evaluations for the analyzers and for the taggers in the next subsections.

6.2 Evaluating the Analyzers

We now present the evaluation of the dialectal morphological analyzers created from the annotated corpora and compare them to existing state-of-the-art analyzers. Tables 2 and 3 present the results on DEV for EGY and LEV, respectively.

Metrics We use several evaluation metrics to measure the effectiveness of the analyzers. First is the **Analyzer Token Recall**, which measures for each token and for each analysis criterion whether the hand-tagged analysis is generated by the morphological analyzer (possibly among others). It is an upper limit on our ability to correctly tag a word. As the name implies, this metric counts all tokens, not just unique types. We apply this metric to all components listed in Section 6.1. The second metric is **OOV**, which represents the cases that are not recognizable by the analyzer, i.e., out of vocabulary. Finally, we present the number of **Analyses per Word**. This is a measure of the degree of ambiguity that should be considered together with the other two (recall and OOV). Overall we want the recall to be high, but we want the ambiguity and OOV rates to be low. Recall that if a word is OOV, a proper-noun back-off analysis is assigned, where the lemma, CODA and stem are given the form of the input word.

Systems For both EGY and LEV, we compare the use of the SAMA analyzer (Graff et al., 2009) (our MSA baseline); a rule-based dialect-affix extended version of SAMA (SAMA$_{ext}$) based on the work of Salloum and Habash (2014); and a combination of CALIMA$_{\text{Egy}}$ with SAMA$_{ext}$. For EGY, the latter is a state-of-the-art comparison point that we do not expect to beat in all of the metrics since it was carefully

developed over years, and using more data than we are. For LEV, we expect CALIMA$_{\text{Egy}}$+SAMA$_{ext}$ to be a very good baseline (because of the similarities between EGY and LEV), but we expect to improve on it.

For both EGY and LEV, we compare different training data sizes, namely 5K, 15K and 45K words. For EGY only, we add an extra step of 135K words since more data is available. For each training size, we compare three settings: (a) a **Lookup** baseline that assumes no learning by paradigm completion, (b) a **ParaFill** setting, which uses paradigm completion as discussed in Section 4, and (c) a combination of **ParaFill** with additional pre-existing resources. The additional resources we use for **ParaFill** differ by dialect. In the case of EGY, we only use SAMA$_{ext}$ (since we are using EGY to simulate a low-resource dialect, we cannot use CALIMA$_{\text{Egy}}$); but we use the CALIMA$_{\text{Egy}}$ analyzer and also SAMA$_{ext}$ for LEV (since for dialects other than EGY, we can always make use of the richer resources available for EGY).

		Analyzer Token Recall							
System		**POS**	**POS5**	**Lemma**	**CODA**	**Stem**	**All**	**OOV**	$\frac{Analyses}{Word}$
SAMA		79.8	97.9	62.2	94.7	85.3	57.1	8.4	11.1
SAMA$_{ext}$		83.2	98.3	64.3	95.8	87.9	58.9	5.0	14.7
CALIMA$_{\text{Egy}}$+SAMA$_{ext}$		94.6	99.5	86.9	97.3	94.8	81.9	1.7	22.0
Train 5K	Lookup$_{\text{Egy}}$	60.6	81.1	57.6	89.9	67.9	55.9	43.2	1.0
	ParaFill$_{\text{Egy}}$	74.5	87.9	69.9	92.6	80.2	67.2	26.1	2.2
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	91.9	98.9	83.8	97.1	93.6	79.9	3.1	12.2
Train 15K	Lookup$_{\text{Egy}}$	69.9	86.2	67.6	91.6	74.7	65.7	32.9	1.3
	ParaFill$_{\text{Egy}}$	85.3	93.8	81.5	95.0	88.3	78.7	14.8	3.9
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	93.6	99.3	88.4	97.4	94.6	85.1	2.4	13.9
Train 45K	Lookup$_{\text{Egy}}$	78.9	91.2	77.1	93.3	81.8	75.5	23.1	1.8
	ParaFill$_{\text{Egy}}$	91.9	97.4	89.2	96.6	93.4	86.6	7.9	6.9
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	94.8	99.6	91.3	97.6	95.5	88.5	1.9	16.9
Train 135K	Lookup$_{\text{Egy}}$	85.7	94.5	84.0	94.4	87.1	82.7	15.9	2.4
	ParaFill$_{\text{Egy}}$	94.5	98.7	92.6	97.1	95.4	90.1	4.6	10.2
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	95.4	99.8	93.1	97.8	96.2	90.5	1.5	20.1

Table 2: EGY Morphological Analysis Recall on DEV

		Analyzer Token Recall							
System		**POS**	**POS5**	**Lemma**	**CODA**	**Stem**	**All**	**OOV**	$\frac{Analyses}{Word}$
SAMA		77.7	92.7	72.5	91.5	87.2	62.1	8.7	9.3
SAMA$_{ext}$		81.2	93.5	74.9	92.7	89.6	64.0	6.0	12.3
CALIMA$_{\text{Egy}}$+SAMA$_{ext}$		86.8	94.8	87.6	93.4	92.5	77.4	3.9	17.3
Train 5K	Lookup$_{\text{Lev}}$	44.7	69.6	46.3	84.1	59.0	43.6	54.1	0.6
	ParaFill$_{\text{Lev}}$	57.4	76.5	57.5	87.9	71.4	52.9	38.8	1.2
	ParaFill$_{\text{Lev}}$+CALIMA$_{\text{Egy}}$+SAMA$_{ext}$	90.3	95.6	91.3	94.9	94.8	84.1	3.6	13.2
Train 15K	Lookup$_{\text{Lev}}$	54.0	75.2	55.1	85.5	65.6	52.5	44.7	0.8
	ParaFill$_{\text{Lev}}$	68.4	83.4	67.3	89.9	78.6	62.6	26.9	2.2
	ParaFill$_{\text{Lev}}$+CALIMA$_{\text{Egy}}$+SAMA$_{ext}$	91.1	95.9	91.7	95.1	94.8	85.3	3.5	14.2
Train 45K	Lookup$_{\text{Lev}}$	70.1	85.7	70.2	88.8	75.7	67.9	28.2	1.1
	ParaFill$_{\text{Lev}}$	79.9	90.6	78.9	93.2	86.5	74.5	15.4	4.0
	ParaFill$_{\text{Lev}}$+CALIMA$_{\text{Egy}}$+SAMA$_{ext}$	92.3	96.5	92.8	95.7	95.4	87.0	3.1	16.0

Table 3: LEV Morphological Analysis Recall on DEV

Results For DEV, for both EGY and LEV, as expected, among the pre-existing systems, SAMA$_{ext}$ outperforms SAMA; and CALIMA$_{\text{Egy}}$+SAMA$_{ext}$ does best of the three options. Also, across all training sizes, paradigm completion outperforms lookup; and using a combination of paradigm completion with a pre-existing state-of-the-art system for MSA (in the case of EGY) or for MSA and EGY (in the case of LEV) does best in terms of token recall. There is of course no guarantee that the tagger later on will select the analysis correctly: while we see that the search space is expanding as needed, we also see that as the training size increases and as additional resources are added, the number of analyses per word increases, adding more ambiguity for the tagger to select from.

System		All	OOV	$\frac{Analyses}{Word}$
SAMA		58.5	7.7	11.3
SAMA$_{ext}$		60.3	4.3	14.7
CALIMA$_{\text{Egy}}$+SAMA$_{ext}$		82.8	1.6	21.7
Train 5K	Lookup$_{\text{Egy}}$	54.4	45.8	0.9
	ParaFill$_{\text{Egy}}$	65.8	28.8	2.1
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	80.2	2.7	12.0
Train 15K	Lookup$_{\text{Egy}}$	63.5	36.5	1.2
	ParaFill$_{\text{Egy}}$	77.0	17.3	3.8
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	84.7	2.1	13.7
Train 45K	Lookup$_{\text{Egy}}$	72.9	27.2	1.7
	ParaFill$_{\text{Egy}}$	84.7	9.8	6.7
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	87.9	1.7	16.6
Train 135K	Lookup$_{\text{Egy}}$	80.5	19.5	2.3
	ParaFill$_{\text{Egy}}$	89.0	5.5	9.8
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	90.2	1.3	19.7

Table 4: EGY Analyzer Recall on TEST

System		All	OOV	$\frac{Analyses}{Word}$
SAMA		61.7	9.6	9.2
SAMA$_{ext}$		63.4	7.0	12.1
CALIMA$_{\text{Egy}}$+SAMA$_{ext}$		77.7	4.7	17.2
Train 5K	Lookup	43.3	54.2	0.6
	ParaFill$_{\text{Lev}}$	50.8	41.1	1.2
	ParaFill$_{\text{Lev}}$ +CALIMA$_{\text{Egy}}$+SAMA$_{ext}$	84.0	4.4	13.4
Train 15K	Lookup	51.5	45.6	0.8
	ParaFill$_{\text{Lev}}$	61.3	28.6	2.2
	ParaFill$_{\text{Lev}}$ +CALIMA$_{\text{Egy}}$+SAMA$_{ext}$	85.1	4.2	14.4
Train 45K	Lookup	66.7	29.3	1.1
	ParaFill$_{\text{Lev}}$	73.9	16.0	4.0
	ParaFill$_{\text{Lev}}$ +CALIMA$_{\text{Egy}}$+SAMA$_{ext}$	87.0	3.7	16.2

Table 5: LEV Analyzer Recall on TEST

		DEV						TEST					
System		**POS**	**POS5**	**Lemma**	**CODA**	**Stem**	**All**	**POS**	**POS5**	**Lemma**	**CODA**	**Stem**	**All**
MADAMIRA-MSA		71.0	78.6	55.2	91.0	82.3	48.4	72.4	79.9	56.6	91.1	83.0	49.5
MADAMIRA-EGY		86.6	91.4	71.8	94.1	89.1	63.8	86.3	91.6	72.2	94.1	88.8	64.2
Train 5K	ParaFill$_{\text{Egy}}$	70.8	74.7	65.1	91.3	77.3	59.5	68.6	72.5	63.8	90.8	75.4	58.0
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	70.6	74.6	65.0	91.4	77.2	59.4	68.5	72.4	63.7	91.0	75.2	58.0
Train 15K	ParaFill$_{\text{Egy}}$	76.7	82.8	71.3	90.0	84.5	62.9	74.9	80.9	70.1	89.9	82.7	61.6
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	80.6	87.4	72.8	91.5	86.7	64.0	80.4	87.6	72.6	91.6	86.0	63.8
Train 45K	ParaFill$_{\text{Egy}}$	84.3	88.8	74.6	92.5	89.4	67.8	82.6	87.2	73.4	92.4	87.5	66.6
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	84.3	89.6	74.9	92.4	89.2	67.4	83.8	89.5	74.3	92.4	88.7	66.7
Train 135K	ParaFill$_{\text{Egy}}$	85.9	91.7	76.1	94.4	90.4	68.3	85.3	91.3	75.6	94.4	89.0	67.8
	ParaFill$_{\text{Egy}}$+SAMA$_{ext}$	84.0	91.4	76.6	94.5	90.2	66.9	83.8	91.5	76.3	94.7	89.3	66.4

Table 6: EGY Tagger Results on DEV and TEST

For some metrics, such as CODA and POS5, the baseline is rather high. Our EGY system trained on 45K words in conjunction with SAMA$_{ext}$ beats the highly engineered CALIMA$_{\text{Egy}}$ system on all metrics. In the case of LEV, our baselines are lower and we can incorporate CALIMA$_{\text{Egy}}$ as a pre-existing system. Our best system beats the best baseline on all metrics starting with 5K training data. We now discuss the performance on the harsh All metric, which is really the best indicator of quality overall, in more detail. For EGY, using paradigm completion on only 5K training data improves above the SAMA and SAMA$_{ext}$ baselines. By 15K we are able to beat the state-of-the-art system CALIMA$_{\text{Egy}}$+SAMA$_{ext}$ and consistently provide less ambiguity than it. This is due to the power of paradigm completion, which adds thousands of unseen inflected forms. We also get a very similar pattern for Levantine, where the error rate in All token recall can be cut by 30% against the high CALIMA$_{\text{Egy}}$+SAMA$_{ext}$ baseline by using a combination of paradigm completion and existing systems at the 5K level; and by 42% at the 45K level.

Tables 4 and 5 present the TEST results for EGY and LEV, respectively (only showing the All results). The results on TEST have the same pattern as those on DEV.

6.3 Evaluating the Taggers

We now discuss the performance of the morphological taggers that we build.

Metrics We use a single metric, **Accuracy**, where we ask whether our predicted result is correct. The evaluation was conducted across all the components described in 6.1 at the word token level.

Systems For each of EGY and LEV, we trained two sets of MADAMIRA systems using classifiers built from different sizes of data sets as described in Section 5. The two sets differed in the underlying

		DEV						TEST					
System		**POS**	**POS5**	**Lemma**	**CODA**	**Stem**	**All**	**POS**	**POS5**	**Lemma**	**CODA**	**Stem**	**All**
MADAMIRA-MSA		68.6	75.2	64.5	87.7	83.3	50.4	67.7	74.6	64.6	87.2	82.2	50.3
MADAMIRA-EGY		75.9	84.6	65.8	88.6	84.1	53.3	75.3	84.2	65.1	88.8	83.8	52.7
Train 5K	$\text{ParaFill}_{\text{Lev}}$	55.8	57.6	53.1	86.0	68.8	46.2	54.0	55.7	51.3	86.1	68.5	45.2
	$\text{ParaFill}_{\text{Lev}}$++	74.6	85.0	70.7	89.3	84.8	55.6	74.7	85.1	69.6	88.4	84.2	55.5
Train 15K	$\text{ParaFill}_{\text{Lev}}$	65.0	68.4	60.6	86.5	75.9	53.0	64.4	67.5	59.1	86.4	75.8	52.3
	$\text{ParaFill}_{\text{Lev}}$++	78.2	85.9	74.6	89.9	85.6	60.6	78.5	85.9	74.0	90.0	86.1	61.1
Train 45K	$\text{ParaFill}_{\text{Lev}}$	75.7	80.2	69.2	89.9	83.7	61.6	75.6	79.7	69.7	89.7	83.7	62.9
	$\text{ParaFill}_{\text{Lev}}$++	81.8	89.4	77.8	91.7	88.4	65.5	82.3	89.2	77.2	91.7	88.5	65.7

Table 7: LEV Tagger Results on DEV and TEST; $\text{ParaFill}_{\text{Lev}}$++ refers to $\text{ParaFill}_{\text{Lev}}$+$\text{CALIMA}_{\text{Egy}}$+$\text{SAMA}_{ext}$

morphological analyzer: the result of paradigm completion only ($\text{ParaFill}_{\text{Egy}}$ or $\text{ParaFill}_{\text{Lev}}$), or this system augmented with additional resources from other variants (MSA for EGY, MSA and EGY for LEV). These are the same analyzers evaluated above in Section 6.2. In addition, evaluations were conducted on MADAMIRA-MSA and MADAMIRA-EGY in order to compare the performance of the EGY and LEV systems to the standard systems. Note that for EGY, MADAMIRA-EGY represents a carefully engineered contrastive system, while for LEV, it represents a plausible alternative to dialect-specific efforts and thus a true baseline.

Results The results for EGY are shown in Table 6 for DEV and TEST. Looking at the DEV results, we see as expected that MADAMIRA-EGY provides a high level of performance across the components of the evaluation. We do not beat this system on POS even with 135K training data. However, we beat it on POS5 and CODA at 135K; 45K is enough training data to beat MADAMIRA-EGY on Stem, and 15K is enough on Lemma and All. We see that performance increases with more training data, as expected. Interestingly, the increase in performance from 45K to 135K (a substantial amount of additional annotation) is much smaller than the previous increments, for all evaluation criteria. We also see that the addition of SAMA_{ext} to the morphological analyzer helps only at the medium amount of 15K (on All); we suspect that this is because at 5K, there is not enough training data to counteract the increased ambiguity that comes from adding these additional analyses, while at larger amounts of training data, the analyzer based on paradigm completion alone generates sufficiently rich databases. Results are roughly similar on TEST.

The results for LEV are shown in Table 7 for DEV and TEST. For LEV, MADAMIRA-EGY represents a baseline. Our system beats this baseline even with only 5K training data, however only if we include +$\text{CALIMA}_{\text{Egy}}$+$\text{SAMA}_{ext}$ (and not on POS). In fact, the results for using the $\text{ParaFill}_{\text{Lev}}$+$\text{CALIMA}_{\text{Egy}}$+$\text{SAMA}_{ext}$ analyzer are consistently better than those for $\text{ParaFill}_{\text{Lev}}$ analyzer alone (the one we get from paradigm completion), though this effect is stronger at smaller training sizes. This shows that the tagger is able to make the right choice despite the steep increase in ambiguity (as seen in Table 3), for example from 2.2 analyses per word for $\text{ParaFill}_{\text{Lev}}$ to 14.2 for $\text{ParaFill}_{\text{Lev}}$+$\text{CALIMA}_{\text{Egy}}$+$\text{SAMA}_{ext}$ at 15K training data. Finally, we see again that more data helps, and we do not see a major flattening of the learning curve at 45K words yet. As with EGY, the results on TEST mirror the ones on DEV for LEV.

Error Analysis We conducted error analyses for the $\text{ParaFill}_{\text{Egy}}$ and $\text{ParaFill}_{\text{Lev}}$ taggers, trained on 45K-word TRAIN and tested on DEV. For EGY, 20.7% of the errors occur because the gold answer is not provided by the morphological analyzer. An additional 5.6% of the errors are back-off cases at the time of preparing the data where $\text{CALIMA}_{\text{Egy}}$ does not produce an answer. OOV cases contribute a further 2.6% of the errors. The rest of the errors are cases where the analyzer provides the correct analysis, but the tagger fails to pick it in context. For LEV, the absence of the gold answer in the analyzer and the OOV entries contribute to 21.7% and 6.6% of the errors, respectively. The other errors occur as the tagger fails to select the correct answer provided by the analyzer.

7 Conclusions and Future Work

This paper has presented a methodology for deriving a morphological analyzer and a morphological tagger for an Arabic dialect. We have shown that this can be done successfully, even with a small amount of data. The approach requires a single type of annotation: a morphological annotation on running text which identifies the normalized spelling, the segmentation, the morphological features, and the lemma for each word. Both the analyzer and the tagger evaluations show the importance of using the paradigm completion approach to creating full inflectional classes: we obtain important error reductions over using just the morphological information supplied in the annotation. Furthermore, we have shown that for Levantine, using Egyptian resources helps performance in both analysis and tagging, even though using such resources greatly increases ambiguity, thus making the tagging task harder.

This paper has also presented a morphological analyzer and a morphological tagger for Levantine. To our knowledge, these are the first of their kind. We plan on using the complete version of the Levantine corpus and then making these resources available.

Our error analysis showed that most of the errors in the tagger come from the tagger itself, not the analyzer. This is understandable, as usually taggers are trained on much larger corpora. In future work, we will investigate whether we can improve the performance of the tagger by training the classifiers on combinations of Levantine text with Egyptian and/or MSA tagged text. Furthermore, we will also apply this methodology to other Arabic dialects; we are currently preparing annotations in the same style for five additional dialects from across the Arab world (see (Al-Shargi et al., 2016) for Moroccan and Sanaani Yemeni), and we will follow exactly the same methodology as laid out in this paper.

Acknowledgment

This paper is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA) under 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.

References

Hitham Abo Bakr, Khaled Shaalan, and Ibrahim Ziedan. 2008. A Hybrid Approach for Converting Written Egyptian Colloquial Dialect into Diacritized Arabic. In *The 6th International Conference on Informatics and Systems, INFOS2008*. Cairo University.

Rania Al-Sabbagh and Roxana Girju. 2012a. A supervised POS Tagger for Written Arabic Social Networking Corpora. In Jeremy Jancsary, editor, *Proceedings of KONVENS 2012*, pages 39–52. ÖGAI, September. Main track: oral presentations.

Rania Al-Sabbagh and Roxana Girju. 2012b. YADAC: Yet another Dialectal Arabic Corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2882–2889.

Faisal Al-Shargi, Aidan Kaplan, Ramy Eskander, Nizar Habash, and Owen Rambow. 2016. Morphologically Annotated Corpora and Morphological Analyzers for Moroccan and Sanaani Yemeni Arabic. In *10th Language Resources and Evaluation Conference (LREC 2016)*.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A Multidialectal Parallel Corpus of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Eric Brill. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4):543–565.

Tim Buckwalter. 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. LDC catalog number LDC2004L02, ISBN 1-58563-324-0.

David Chiang, Mona Diab, Nizar Habash, Owen Rambow, and Safiullah Shareef. 2006. Parsing Arabic Dialects. In *Proceedings of EACL*, Trento, Italy.

Walter Daelemans and Antal van den Bosch. 2005. *Memory-Based Language Processing*. Studies in Natural Language Processing. Cambridge University Press, Cambridge, UK.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2007. Automated Methods for Processing Arabic Text: From Tokenization to Base Phrase Chunking. In Antal van den Bosch and Abdelhadi Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Kluwer/Springer.

Mona Diab, Nizar Habash, Owen Rambow, Mohamed Altantawy, and Yassine Benajiba. 2010. COLABA: Arabic Dialect Annotation and Processing. In *LREC Workshop on Semitic Language Processing*, pages 66–74.

Mona Diab, Nizar Habash, Owen Rambow, and Ryan Roth. 2013. LDC Arabic Treebanks and Associated Corpora: Data Divisions Manual. *arXiv preprint arXiv:1309.5652*.

Kevin Duh and Katrin Kirchhoff. 2005. POS Tagging of Dialectal Arabic: a Minimally Supervised Approach. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Semitic '05, pages 55–62, Ann Arbor, Michigan.

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

Ramy Eskander, Nizar Habash, and Owen Rambow. 2013b. Automatic Extraction of Morphological Lexicons from Morphologically Annotated Corpora. In *Proceedings of tenth Conference on Empirical Methods in Natural Language Processing*.

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, 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, 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, Reem Faraj, and Ryan Roth. 2009. Syntactic Annotation in the Columbia Arabic Treebank. In *Proceedings of MEDAR International Conference on Arabic Language Resources and Tools*, Cairo, Egypt.

Nizar Habash, Mona Diab, and Owen Rabmow. 2012a. Conventional Orthography for Dialectal Arabic. In *Proceedings of LREC*, Istanbul, Turkey.

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.

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 NAACL-HLT*, Atlanta, GA.

Nizar Habash. 2007. Arabic Morphological Representations for Machine Translation. In Antal van den Bosch and Abdelhadi Soudi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*. Kluwer/Springer.

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

Jan Hajič. 2000. Morphological Tagging: Data vs. Dictionaries. In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*, Seattle, WA.

Mustafa Jarrar, Nizar Habash, Diyam Akra, and Nasser Zalmout. 2014. Building a Corpus for Palestinian Arabic: a Preliminary Study. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 18–27, Doha, Qatar, October. Association for Computational Linguistics.

Salam Khalifa, Nizar Habash, Dana Abdulrahim, and Sara Hassan. 2016. A Large Scale Corpus of Gulf Arabic. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Portorož, Slovenia.

H. Kilany, H. Gadalla, H. Arram, A. Yacoub, A. El-Habashi, and C. McLemore. 2002. Egyptian Colloquial Arabic Lexicon. LDC catalog number LDC99L22.

Mohamed Maamouri, Ann Bies, Seth Kulick, Dalila Tabessi, and Sondos Krouna. 2012. Egyptian Arabic Treebank DF Parts 1-8 V2.0 - LDC catalog numbers LDC2012E93, LDC2012E98, LDC2012E89, LDC2012E99, LDC2012E107, LDC2012E125, LDC2013E12, LDC2013E21.

Mohamed Maamouri, Ann Bies, Seth Kulick, Michael Ciul, Nizar Habash, and Ramy Eskander. 2014. Developing an Egyptian Arabic Treebank: Impact of Dialectal Morphology on Annotation and Tool Development. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Abir Masmoudi, Mariem Ellouze Khmekhem, Yannick Esteve, Lamia Hadrich Belguith, and Nizar Habash. 2014. A Corpus and Phonetic Dictionary for Tunisian Arabic Speech Recognition. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

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.

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 *In Proceedings of LREC*, Reykjavik, Iceland.

Wael Salloum and Nizar Habash. 2011. Dialectal to Standard Arabic Paraphrasing to Improve Arabic-English Statistical Machine Translation. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 10–21, Edinburgh, Scotland.

Wael Salloum and Nizar Habash. 2014. ADAM: Analyzer for Dialectal Arabic Morphology. *Journal of King Saud University-Computer and Information Sciences*, 26(4):372–378.

Kamel Smaïli, Mourad Abbas, Karima Meftouh, and Salima Harrat. 2014. Building Resources for Algerian Arabic Dialects. In *15th Annual Conference of the International Communication Association Interspeech*.

Clare Voss, Stephen Tratz, Jamal Laoudi, and Douglas Briesch. 2014. Finding Romanized Arabic Dialect in Code-Mixed Tweets. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2249–2253, Reykjavik, Iceland, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L14-1086.

Imed Zitouni, Jeffrey S. Sorensen, and Ruhi Sarikaya. 2006. Maximum Entropy Based Restoration of Arabic Diacritics. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 577–584, Sydney, Australia.