Journal of King Saud University – Computer and Information Sciences (2014) 26, 372–378

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

ADAM: Analyzer for Dialectal Arabic Morphology

Wael Salloum [a,*], Nizar Habash [b]

[a] Center for Computational Learning Systems, Columbia University, United States
[b] New York University Abu Dhabi, United Arab Emirates

Available online 2 October 2014

KEYWORDS

Arabic natural language processing;
Dialectal Arabic;
Arabic morphology;
Machine translation

Abstract While Modern Standard Arabic (MSA) has many resources, Arabic Dialects, the primarily spoken local varieties of Arabic, are quite impoverished in this regard. In this article, we present ADAM (Analyzer for Dialectal Arabic Morphology). ADAM is a poor man's solution to quickly develop morphological analyzers for dialectal Arabic. ADAM has roughly half the out-of-vocabulary rate of a state-of-the-art MSA analyzer and is comparable in its recall performance to an Egyptian dialectal morphological analyzer that took years and expensive resources to build.

1. Introduction

Arabic dialects, or the primarily spoken local varieties of Arabic, have recently received increased attention in the field of natural language processing (NLP). An important challenge for work on these dialects is to create morphological analyzers, or tools that provide for a particular written word all of its possible analyses out of context. While Modern Standard Arabic (MSA) has many such resources (Graff et al., 2009; Smrž, 2007; Habash, 2007), Dialectal Arabic (DA) is quite impoverished (Habash et al., 2012b). Furthermore, MSA and the dialects are quite different morphologically: Habash et al., 2012b reported that only 64% of Egyptian Arabic words are analyzable using an MSA analyzer. Thus, using MSA resources to process the dialects will have limited value. Additionally, as for any language or dialect, developing good large-scale coverage lexicons and analyzers can require much time and effort.

In this article, we present ADAM (Analyzer for Dialectal Arabic Morphology). ADAM is a poor man's solution for developing a quick and dirty morphological analyzer for dialectal Arabic. ADAM can be used as is or can function as the first step in bootstrapping analyzers for Arabic dialects. It covers all part-of-speech (POS) tags just as any other morphological analyzer; however, because we use ADAM mainly to process text, we do not model phonological differences between Arabic dialects and we do not evaluate the differences in phonology. In this work, we apply ADAM extensions to MSA clitics to generate proclitics and enclitics for different Arabic dialects. This technique can also be applied to stems to generate dialectal stems; however, that is outside the scope of this work.

In Section 2, we review some of the challenges of processing Arabic in general and Arabic dialects in particular. We discuss related work in Section 3, and we outline and detail our approach in Section 4. Finally, in Section 5, we present several detailed evaluations using a variety of metrics and compare against state-of-the-art analyzers of MSA and Egyptian Arabic.

* Corresponding author.
E-mail addresses: wael@ccls.columbia.edu (W. Salloum), nizar.habash@nyu.edu (N. Habash).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.06.010
1319-1578

2. Arabic language facts and challenges

In this section, we discuss the challenges of processing Arabic in general and dialectal Arabic (DA) in particular.

2.1. Arabic linguistic challenges

The Arabic language is quite challenging for NLP. Arabic is a morphologically complex language that includes rich inflectional morphology, expressed both templatically and affixationally, and several classes of attachable clitics. For example, the Arabic word وسيكتبونها (w + s + y − ktb − wn + hA[1], 'and they will write it') has two proclitics (+ و w+, 'and,' and +س s+, 'will'), one prefix (–ي y–, '3rd person'), one suffix (ون- –wn, 'masculine plural') and one pronominal enclitic (ها+ +hA, 'it/her'). Additionally, Arabic is written with optional diacritics that specify short vowels, consonantal doubling and the nunation morpheme. The absence of these diacritics together with the language's rich morphology lead to a high degree of ambiguity: e.g., the Buckwalter Arabic Morphological Analyzer (BAMA) (Buckwalter, 2004) produces an average of 12 analyses per word. Moreover, some Arabic letters are often spelled inconsistently, which leads to an increase in both sparsity (multiple forms of the same word) and ambiguity (the 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 (ElKholy and Habash, 2010). Arabic complex morphology and ambiguity are handled using tools for analysis, disambiguation and tokenization (Habash and Rambow, 2005; Diab et al., 2007). In this article, we focus on the problem of morphological analysis, which is concerned with identifying all and only the possible readings (or analyses) for a word out of context (Habash, 2010).

2.2. Dialectal Arabic challenges

Contemporary Arabic is a collection of varieties: MSA, which has a standard orthography and is used in formal settings, and DAs, which are commonly used informally and with increasing presence on the web but do not have standard orthographies. There are several DA varieties that vary primarily geographically, e.g., Levantine Arabic, Egyptian Arabic, and so on (Habash, 2010). DAs differ from MSA phonologically, morphologically and, to a lesser degree, syntactically. The differences between MSA and DAs have often been compared to those between Latin and the Romance languages (Habash, 2006). The morphological differences are most noticeably expressed in the use of clitics and affixes that do not exist in MSA. For instance, the Levantine and Egyptian Arabic equivalent of the MSA example above is وحيكتبوها, (*w + H + y − ktb − w + hA*, 'and they will write it').[2] The optionality of vocalic diacritics helps hide some of the differences resulting from vowel changes; compare the diacritized forms: *wHayuktubuwhA* (Levantine), *waHayiktibuwhA* (Egyptian) and *wasayaktubuwnahA* (MSA) (Salloum and Habash, 2011). It is important to note that Levantine and Egyptian differ significantly in phonology, but the orthographical choice of dropping short vowels bridges the gap between them. For extended discussion about the difference between the two dialects, we refer the reader to the following books: Omar, 1976; Abdel-Massih et al., 1979; Cowell, 1964. In this work, we focus on processing text, and therefore, we do not model short vowels.

All of the NLP challenges of MSA described above are shared by DA. However, the lack of standard orthographies for the dialects and their numerous varieties poses new challenges (Habash et al., 2012a). Additionally, DAs are rather impoverished in terms of available tools and resources compared to MSA; e.g., there are very few parallel DA-English corpora and almost no MSA-DA parallel corpora. The number and sophistication of morphological analysis and disambiguation tools for DA are very limited in comparison to those of MSA (Duh and Kirchhoff, 2005; Habash and Rambow, 2006; Abo Bakr et al., 2008; Habash et al., 2012b). MSA tools cannot be effectively used to handle DA: Habash and Rambow, 2006 reported that less than two-thirds of Levantine verbs can be analyzed using an MSA morphological analyzer and Habash et al., 2012b reported that only 64% of Egyptian Arabic words are analyzable using an MSA analyzer.

Salloum and Habash (2011) reported that 26% of out-of-vocabulary (OOV) terms in dialectal corpora have MSA readings or are proper nouns. The rest, 74%, are dialectal words. They classify the dialectal words into two types: words that have MSA-like stems and dialectal affixational morphology (affixes/clitics) and those that have dialectal stems and possibly dialectal morphology. The former set accounts for almost half of all OOVs (49.7%) or almost two-thirds of all dialectal OOVs. In this article, like Salloum and Habash, 2011, we only target dialectal affixational morphology cases, as they are the largest class involving dialectal phenomena that do not require extension to stem lexica.

3. Related work

There has been a large amount of works on Arabic morphological analysis with a focus on MSA (Beesley et al., 1989; Kiraz, 2000; Buckwalter, 2004; Al-Sughaiyer and Al-Kharashi, 2004; Attia, 2008; Graff et al., 2009; Altantawy et al., 2011; Attia et al., 2013). In comparison, only a few efforts have targeted DA morphology (Kilany et al., 2002; Habash and Rambow, 2006; Abo Bakr et al., 2008; Salloum and Habash, 2011; Mohamed et al., 2012; Habash et al., 2012b; Hamdi et al., 2013).

Efforts for modeling dialectal Arabic morphology generally fall in two camps. First are the solutions that focus on extending MSA tools to cover DA phenomena. For example, Abo Bakr et al., 2008 and Salloum and Habash, 2011 extended the BAMA/SAMA databases (Buckwalter, 2004; Graff et al., 2009) to accept DA prefixes and suffixes. Such efforts are interested in mapping DA text to some MSA-like form; as such, they do not model DA linguistic phenomena. These solutions are fast and cheap to implement.

The second camp is interested in modeling DA directly. However, the attempts at doing so are lacking in coverage in one dimension or another. The earliest effort on Egyptian that

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

[2] A spelling variation for this Egyptian Arabic word is وهيكتبوها *w + h + y − ktb − w + hA*.

we know of is the Egyptian Colloquial Arabic Lexicon (Kilany et al., 2002). This resource was the basis for developing the CALIMA Egyptian morphological analyzer (Habash et al., 2012b, 2013). Another effort is the work by (Habash and Rambow, 2006), which focuses on modeling DAs together with MSA using a common multi-tier finite-state-machine framework. Mohamed et al., 2012 annotated a collection of Egyptian for morpheme boundaries and used these data to develop an Egyptian tokenizer. Eskander et al., 2013b presented a method for automatically learning inflectional classes and associated lemmas from morphologically annotated corpora. Hamdi et al., 2013 took advantage of the closeness of MSA and its dialects to build a translation system from Tunisian Arabic verbs to MSA verbs. This approach to modeling Arabic dialect morphology usually results in better quality morphological analyzers compared to the shallow techniques presented by the first camp. However, they are expensive and require significantly more resources and efforts. Furthermore, they are harder to extend to new dialects because they require annotated training data and/or hand-written rules for each new dialect.

The work presented in this article is closer to the first camp. We extend beyond this previous work in covering more dialects and presenting detailed evaluations of coverage and recall against two state-of-the art systems: SAMA for MSA and CALIMA for Egyptian Arabic.

4. Approach

In this section, we describe our approach for developing ADAM, the Analyzer of Dialectal Arabic Morphology.

4.1. Motivation

ADAM is intended for the use on dialectal Arabic text to improve machine translation (MT) performance; thus, we focus on orthography as opposed to phonology. While consonants and long vowels are written in Arabic as letters, short vowels are optional diacritics over or under the letters. This leads to people ignoring short vowels in writing because the interpretation of the work can be inferred from the context. Even when people write short vowels, they are inconsistent and the short vowels might end up over or under the wrong letter due to visual difficulties. Research in MT, therefore, tends to drop short vowels completely, and because ADAM is built to improve MT performance, we choose to drop short vowels from ADAM.

Morphemes of different Arabic dialects (at least the ones we are addressing in this work: Levantine, Egyptian, and Iraqi) usually share similar morpho-syntactic behavior, such as future particles, progressive particles, verb negation, pronouns, indirect object pronouns, and propositions. Furthermore, many morphemes are shared among these dialects, especially when dropping short vowels. Therefore, modeling orthographic morphology of multiple dialects in one system seems reasonable. When querying ADAM, the user has the option to specify the dialect of the query word to exclude other dialects' readings.

4.2. Databases

ADAM is built on top of the SAMA databases (Graff et al., 2009). The SAMA databases contain three tables of Arabic stems, complex prefixes and complex suffixes and three additional tables with constraints on matching them. We define a *complex prefix* as the full sequence of prefixes/proclitics that may appear at the beginning of a word. *Complex suffixes* are defined similarly. MSA, according to the SAMA database, has 1208 complex prefixes and 940 complex suffixes, which correspond to 49 simple prefixes and 177 simple suffixes, respectively. The number of combinations in prefixes is much larger than that in suffixes, which explains the different proportions of complex affixes to simple affixes.

ADAM follows the same database format as the ALMOR morphological analyzer/generator (Habash, 2007), which is the rule-based component of the MADA system for morphological analysis and disambiguation of Arabic (Habash and Rambow, 2005; Roth et al., 2008). As a result, ADAM outputs analyses as lemma and feature-value pairs including clitics. This makes it easier to replace the ALMOR database with the ADAM database in any MSA NLP system that uses ALMOR to extend it to the dialects processed by ADAM. The model, however, has to be re-trained on dialectal data. For example, MADA can be extended to Levantine by plugging the ADAM database in place of the ALMOR database and training MADA on the Levantine TreeBank.

4.3. SADA rules

We extend the SAMA database through a set of rules that add Levantine, Egyptian, and Iraqi dialectal affixes and clitics to the database. We call this *Standard Arabic to Dialectal Arabic* mapping technique SADA.[3] To add a dialectal affix (or clitic), we first look for an existing MSA affix with the same morpho-syntactic behavior and then write a rule (a regular expression) that captures all instances of this MSA affix (either by itself or within complex affixes) and replaces them with the new dialectal affix. In addition to changing the surface form of the MSA affix, we change any feature in the retrieved database entry if needed, such as part-of-speech (POS), proclitics and enclitics, along with adding new features if needed, such as 'dia,' which gives the dialect of this new dialectal affix. Finally, the newly updated database entries are added to the database while preserving the original entries to maintain analyses of MSA words.

SADA rules were created by one of the authors, who is a native speaker of Levantine Arabic with good knowledge of Egyptian and Iraqi. Writing the rules required approximately 70 h of work and did not require any computer science knowledge. The task does not require a linguist either; any native speaker with basic understanding of morphology (especially POS) can write these rules. Therefore, using crowdsourcing, ADAM can be extended easily and cheaply to other dialects or sub-dialects compared to other approaches (such as MAGEAD and CALIMA) that may take months if not years to cover a new dialect. Moreover, because SADA rules can be applied to any ALMOR-like database, both MAGEAD and CALIMA can be extended by SADA to create a version of ADAM superior to these analyzers. We extend CALIMA with SADA and evaluate it in Section 5.

To create the list of rules, we started with a list of highly frequent dialectal words that we acquired from Raytheon BBN Technologies in 2010. The process of creating the word list

[3] SADA, صدى, *Sadý*, means 'echo' in Arabic.

started by extracting all of the words that are in annotated non-MSA regions in the GALE transcribed audio data (roughly 2000 h) and intersecting them with words in the GALE web data (Webtext). Normally, many of these words are MSA, and they had to be excluded either automatically or manually to end up with a list of 22,965 types (821,700 tokens) that are, for the most part, dialect words. Each dialectal word occurred with different frequencies in the two corpora above. The maximum of the two frequencies was picked as the word frequency, and the list was ordered according to this frequency. We annotated the top 1000 words in this list for dialect and POS to study the dialectal phenomena we are dealing with. We analyzed the morphology of these words to identify the frequent types of morphemes and their spelling variations, along with the common morphemes and shared morpho-syntactic behavior among dialects. This analysis led the creation of the first version of SADA rules. New rules were added later after obtaining more dialectal text to analyze.

4.4. Examples

We discuss two examples that represent two different classes of extensions: dialectal affixes with comparable MSA equivalents and dialectal affixes that have no MSA equivalent. For the first type, we consider the dialectal future prefix +ح *H*+ 'will' (and its orthographical variations: the Levantine +رح *rH*+ and the Egyptian +ه *h*+. This prefix has a similar behavior to the standard Arabic future particle +س *s*+. As such, an extension rule would create a copy of each occurrence of the MSA prefix and replace it with the dialectal prefix. SADA uses this rule to extend the SAMA database and adds the prefix Ha/FUT_PART and many other combinations involving it, e.g., `wa/PART + Ha/FUT_PART + ya/IV3MS`, `Ha/FUT_PART + na/IV1P`, and so on.

For the second type, we consider the Levantine dialect demonstrative prefix +ه *h*+ 'this/these' that attaches to nouns on top of the determiner particle +ال *Al*+ 'the'. Because this particle has no equivalent in MSA, we have a rule that extends the determiner particle +ال *Al*+ 'the' to allow the new particle to attach to it. This is equivalent to having a new particle +هال *hAl*+ 'this/these the' that appears wherever the determiner particle is allowed to appear.

The rules (1,021 in total) introduce 16 new dialectal prefixes (plus spelling variants and combinations) and 235 dialectal suffixes (again, plus spelling variants and combinations). Table 1 presents a sample of the new proclitics/enclitics added by SADA.

As an example of ADAM output, consider the second set of rows in Fig. 1, where a single analysis is shown.

Table 1 An example list of dialectal affixes added by SADA. 'L' is for Levantine, 'E' for Egyptian, 'I' for Iraqi, and 'M' for multi-dialect. PNG is for Person-Number-Gender.

	Dialect	POS	Comments
Prefix			
b	L, E	PROG_PART	Simple present
mn	L	PROG_PART	Simple present (with n/IV1P)
d	I	PROG_PART	Simple present
Em, Eb	L	PROG_PART	Continuous tense
H	M	FUT_PART	Future particle
h	E	FUT_PART	Future Particle
rH	L	FUT_PART	Future particle
mA, m	M	NEG_PART	Negation
t	L	JUS_PART	'in order to'
hAl	L, I	DEM_DET_PART	'this/these' the
E	L, I	PREP_PART	'on/to/about' 'on/to/about the'
EAl, El	M	PREP_DET_PART	
yA	M	VOC_PART	Vocative particle
Suffix			
l + [pronPGN]	M	PREP + VSUFF_IO:[PGN]	Indirect object, e.g., lw, lhA, etc.
$	E, L	NEG_PART	Negation suffix
$	I	PRON_2MS	Suffixing pronoun
j	I	PRON_2FS	Suffixing pronoun
ky	L	PRON_2FS	When preceded by a long vowel
yk	L	PRON_2FS	When preceded by a short vowel
ww	L	VSUFF_SUBJ:3P + VSUFF_DO:3MS	Suffix: subject is 3P, object is 3MS

Lev. Word English Equiv.	وماحيكتبلو *wmAHyktblw* 'And he will not write to him'					
Analysis:	**Proclitics**			**[Lemma & Features]**	**Enclitics**	
Levantine:	w+	mA+	H+	yktb	+l	+w
POS:	conj+	neg+	fut+	[katab IV subj:3MS	+prep	+pron$_{3MS}$
English:	and+	not+	will+	voice:act] he writes	+to	+him

Figure 1 An example illustrating the ADAM analysis output for a Levantine Arabic word.

5. Evaluation

In this section, we evaluate ADAM against two state-of-the-art morphological analyzers: SAMA (v 3.1) (Graff et al., 2009) for MSA and CALIMA (v0.6) (Habash et al., 2012b) for Egyptian Arabic. We apply the SADA extensions to both SAMA and CALIMA to produce two ADAM versions: $ADAM_{sama}$ and $ADAM_{calima}$.

We compare the performance of the four analyzers on two metrics: out-of-vocabulary (OOV) rate and in-context part-of-speech recall. We consider data collections from Levantine and Egyptian Arabic. In this work, we do not evaluate the performance of our system on Iraqi Arabic.

Finally, we report on the contribution of ADAM in a machine translation (MT) task.

5.1. Evaluation of coverage

We compare the performance of the four analyzers outlined above in terms of their OOV rate: the percentage of analyzable types or tokens out of all types or tokens. This metric does not guarantee the correctness of the analyses, just that an analysis is available. For tasks such as undiacritized tokenization, this may actually be sufficient in some cases.

We use the dialectal side of a *DA-English* parallel corpus of approximately 3.8 M untokenized words, which was used by (Habash et al., 2013). ~2.7 M tokens (and ~315 K types) are in Egyptian Arabic, and ~1.1 M tokens (and ~137 K types) are in Levantine Arabic.

Table 2 shows the performance of the four morphological analyzers on both Levantine and Egyptian data in terms of type/token OOV rate. $ADAM_{sama}$ and $ADAM_{calima}$ improve over the base analyzers they extend (SAMA and CALIMA, respectively). For SAMA, $ADAM_{sama}$ reduces the OOV rates by over 50% in types and 66% in tokens for Levantine. The respective values for Egyptian Arabic types and tokens are 29% and 50%. The performance of $ADAM_{sama}$ is quite competitive with that of CALIMA, a system that took years and great resources to develop. The OOV rates on Egyptian Arabic for $ADAM_{sama}$ and CALIMA are almost identical, but $ADAM_{sama}$ outperforms CALIMA on Levantine Arabic, which CALIMA was not designed for. Furthermore, $ADAM_{calima}$ improves over CALIMA by a smaller percentage, suggesting that the ADAM approach can be useful even with well-developed dialectal analyzers.

5.2. Evaluation of in-context part-of-speech recall

We evaluate the four analyzers discussed above in terms of their in-context POS recall (IPOSR). IPOSR is defined as the percentage of time an analyzer produces an analysis with the correct POS in context among the set of analyses for a particular word. To compute IPOSR, we require manually annotated data sets: the Levantine Arabic TreeBank (LATB) (Maamouri et al., 2006) and the Egyptian Arabic (ARZ) TreeBank (Eskander et al., 2013a). We report IPOSR in terms of types and tokens for Levantine and Egyptian on the four analyzers in Table 3.

Table 2 Coverage evaluation of the four morphological analyzers on the Levantine and Egyptian side of the MT training data in terms of types and tokens OOV rate.

Data set		Levantine		Egyptian	
Word count		*Type* 137,257	*Token* 1,132,855	*Type* 315,886	*Token* 2,670,520
System	Metric	*Type* (%)	*Token* (%)	*Type* (%)	*Token* (%)
SAMA	*OOV rate*	35.5	16.1	47.2	14.0
$ADAM_{sama}$	*OOV rate*	16.1	5.5	33.4	7.0
CALIMA	*OOV rate*	20.4	6.9	34.4	7.2
$ADAM_{calima}$	*OOV rate*	15.6	5.3	32.3	6.6

Table 3 Correctness evaluation of the four morphological analyzers on the Levantine and Egyptian TreeBanks in terms of types and tokens. Type* is the number of unique word-POS pairs in the TreeBank.

Data set		Levantine TB		Egyptian TB	
Word count		*Type** 4201	*Token* 19,925	*Type** 65,064	*Token* 309,386
System	Metric	*Type** (%)	*Token* (%)	*Type** (%)	*Token* (%)
SAMA	*OOV rate*	17.1	9.8	20.3	8.4
	POS recall	68.3	64.6	60.0	75.1
$ADAM_{sama}$	*OOV rate*	2.8	1.2	7.6	2.0
	POS recall	86.7	79.7	75.5	91.4
CALIMA	*OOV rate*	3.8	1.7	5.6	1.6
	POS recall	86.0	80.2	85.4	94.7
$ADAM_{calima}$	*OOV rate*	2.5	1.0	5.2	1.4
	POS recall	87.8	80.7	85.5	94.7

We observe, first of all, that the OOV rates in the TreeBank data are much lower than OOV rates in the data we used in the previous section on coverage evaluation. The reduction in OOV rate using the dialectal analyzers (beyond SAMA) is also more intense. This may be a result of the TreeBank data being generally cleaner and less noisy than the general corpus data we used. Next, we observe that SAMA has very low IPOSR rates that are consistent with previous research cited above. $ADAM_{sama}$ improves the overall IPOSR for both Levantine and Egyptian Arabic by approximately 27% and 23% relative for types and tokens, respectively. ADAM and CALIMA are almost tied in performance in Levantine Arabic, but CALIMA outperforms ADAM for Egyptian Arabic, as expected. Finally, $ADAM_{calima}$ improves a bit more on CALIMA for Levantine Arabic and makes less of an impact for Egyptian Arabic. All of this suggests that the ADAM solution is quite competitive with state-of-the-art analyzers given the ease and speed with which it was created. ADAM can make a good bootstrapping method for annotation of dialectal data or for building more linguistically precise dialectal resources.

We should note that this recall-oriented evaluation ignores possible differences in precision that are likely to result from the fact that the ADAM method tends to produce more analyses per word than the original analyzers it extends. In fact, in the case of Egyptian Arabic, $ADAM_{sama}$ produces 21.8 analyses per word compared to SAMA's 13.9 and $ADAM_{calima}$ produces 31.4 analyses per word as opposed to CALIMA's 26.3. Without a full, careful and large-scale evaluation of the produced analyses, it is difficult to quantify the degree of correctness or plausibility of the ADAM analyses.

5.3. Evaluation on machine translation tasks

We designed ADAM to be used as a part of machine translation tools and tasks to improve the output quality. In the following sub-sections, we summarize the previous results of the MT tools and tasks in which ADAM was used.

5.3.1. ADAM with ELISSA

ADAM is used as part of ELISSA (Salloum and Habash, 2013), a DA-to-MSA MT system that supports dialectal Arabic to English MT by pivoting (or bridging) on MSA. Salloum and Habash, 2011 showed how to use ADAM as a preprocessing step to tokenize dialectal Arabic OOV words into smaller units (tokens) to give them a better chance of being translated correctly into English. This method improved over their 36.16% BLEU[4] baseline by 0.34% BLEU even though they were targeting a tiny percentage of the test set (roughly 0.6% of all words). They also used ADAM in an Analysis/Transfer/Generation method applied on MT OOV words, which resulted in a 0.45% BLEU improvement over the same baseline mentioned above. Furthermore, Salloum and Habash, 2011 extended the selection of OOV words that needs to be handled to include low frequency words in the MT training data. They used ADAM to classify low frequency words into three categories, Dialect-Only, MSA-Only, and Dialect + MSA, and they empirically decided on a cutting threshold for each category. This classification helped their technique better select words for transfer into MSA and resulted in an improvement of 0.62%.

[4] BLEU (Papineni et al., 2002) is an evaluation metric for MT systems.

5.3.2. Dialect identification for MT system selection

ADAM is used in a sentence-level dialect identification approach for machine translation system selection when translating mixed dialect input (MSA and DA) (Salloum et al., 2014). We acquired two sets of training data: DA-to-English (5 M words) and MSA-to-English (57 M words). We built four MT systems from these parallel corpora: DA-to-English SMT, MSA-to-English SMT, DA + MSA-to-English SMT, and a DA-to-English hybrid MT system based on the ELISSA-based MSA-pivoting presented in Salloum and Habash, 2013. The fourth MT system was the best among the four, with a BLEU score of 33.9%. To leverage the use of these four MT systems, we propose a system selection approach to benefit from the strengths while avoiding the weaknesses. To do so, we trained a sentence-level four-class classifier that predicts, for an input Arabic sentence, the MT system that should translate this sentence based on linguistic features extracted from the Arabic sentence. Some of the features in this work are extracted from the sentence using ADAM to determine the dialectness of this sentence. A four-class classifier trained on these features alone resulted in an improvement of 0.9% BLEU over the best single MT system (i.e., the fourth system).

6. Conclusions and future work

In this work, we presented a cheap and easy method to develop morphological analyzers for dialectal Arabic. Our approach is to extend an MSA morphological analyzer's database through a set of handwritten rules to add new entries of dialectal affixes into the database. We evaluated ADAM's performance on Levantine and Egyptian. We showed that ADAM has approximately half the OOV rate of SAMA (MSA) and is comparable in its recall performance to CALIMA, an Egyptian dialectal morphological analyzer that required years and expensive resources to build. Furthermore, ADAM has been shown to help in machine translation tasks.

In the future, we plan to add new types of rules: rules that create new dialectal stems by copying and modifying existing MSA stems. We also plan to apply our approach to other Arabic dialects.

Acknowledgments

This research was supported by the Defense Advanced Research Projects Agency (DARPA) GALE program, contract HR0011-06-C-0022, and the DARPA BOLT program, contract No. HR0011-12-C-0014. Any opinions, findings, conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the view of DARPA. We would like to thank John Makhoul and Spyros Matsoukas for helpful discussions and feedback and for providing us with the data we used for initial analysis of dialectal phenomena.

References

Abdel-Massih, E.T., Abdel-Malek, Z.N., Badawi, E.S.M., 1979. A Reference Grammar of Egyptian Arabic. Georgetown University Press.

Abo Bakr et al., 2008 Abo Bakr, H., Shaalan, K., Ziedan, I., 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.

Al-Sughaiyer, I., Al-Kharashi, I., 2004. Arabic morphological analysis techniques: a comprehensive survey. J. Am. Soc. Inform. Sci. Technol. 55 (3), 189–213.

Altantawy, M., Habash, N., Rambow, O., 2011. Fast yet rich morphological analysis. In Proceedings of the 9th International Workshop on Finite-State Methods and Natural Language Processing (FSMNLP 2011), Blois, France.

Attia, M., 2008. Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation (PhD thesis). The University of Manchester, Manchester, UK.

Attia, M., Pecina, P., Toral, A., van Genabith, J., 2013. A corpus-based finite-state morphological toolkit for contemporary Arabic. J. Logic Comput., 070.

Beesley, K., Buckwalter, T., Newton, S., 1989. Two-level finite-state analysis of Arabic morphology. In Proceedings of the Seminar on Bilingual Computing in Arabic and English, page n.p.

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

Cowell, M.W., 1964. A Reference Grammar of Syrian Arabic. Georgetown University Press.

Diab, M., Hacioglu, K., Jurafsky, D., 2007. Automated methods for processing Arabic text: from tokenization to base phrase chunking. In: Van den Soudi, A., Soudi, A. (Eds.), Arabic Computational Morphology: Knowledge-based and Empirical Methods. Kluwer/Springer.

Duh, K., Kirchhoff, K., 2005. POS tagging of dialectal Arabic: a minimally supervised approach. In Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, Semitic '05, Ann Arbor, Michigan, pp. 55–62.

ElKholy, A., Habash, N., 2010. Techniques for Arabic morphological detokenization and orthographic denormalization. In Proceedings of the seventh International Conference on Language Resources and Evaluation (LREC), Valletta, Malta.

Eskander, R., Habash, N., Bies, A., Kulick, S., Maamouri, M., 2013a. Automatic correction and extension of morphological annotations. In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, Sofia, Bulgaria. Association for Computational Linguistics, pp. 1–10.

Eskander, R., Habash, N., Rambow, O., 2013b. Automatic extraction of morphological lexicons from morphologically annotated corpora. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA. Association for Computational Linguistics, pp. 1032–1043.

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

Habash, N., 2006. On Arabic and its dialects. Multilingual Mag. 17 (81).

Habash, N., 2007. Arabic morphological representations for machine translation. In: van den Bosch, A., Soudi, A. (Eds.), Arabic Computational Morphology: Knowledge-based and Empirical Methods. Springer.

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

Habash, N., Rambow, O., 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), Ann Arbor, Michigan, pp. 573–580.

Habash, N., Rambow, O., 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, Sydney, Australia, pp. 681–688.

Habash, N., Soudi, A., Buckwalter, T., 2007. On Arabic transliteration. In: Van den Bosch, A., Soudi, A. (Eds.), Arabic Computational Morphology: Knowledge-based and Empirical Methods. Springer.

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

Habash, N., Eskander, R., Hawwari, A., 2012b. A morphological analyzer for Egyptian Arabic. In NAACL-HLT 2012 Workshop on Computational Morphology and Phonology (SIGMORPHON2012), pp. 1–9.

Habash, N., Roth, R., Rambow, O., Eskander, R., Tomeh, N., 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.

Hamdi, A., Boujelbane, R., Habash, N., Nasr, A., et al. 2013. The effects of factorizing root and pattern mapping in bidirectional Tunisian-standard Arabic machine translation, MT Summit.

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

Kiraz, G.A., 2000. Multitiered nonlinear morphology using multitape finite automata: a case study on Syriac and Arabic. Comput. Linguist. 26 (1), 77–105.

Maamouri, M., Bies, A., Buckwalter, T., Diab, M., Habash, N., Rambow, O., Tabessi, D., 2006. Developing and using a pilot dialectal Arabic TreeBank. In Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC'06, Genoa, Italy.

Mohamed, E., Mohit, B., Oflazer, K., 2012. Annotating and learning morphological segmentation of Egyptian colloquial Arabic. In Proceedings of the Language Resources and Evaluation Conference (LREC), Istanbul.

Omar, M.K., 1976. Levantine and Egyptian Arabic: Comparative Study. Department of State.

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, pp. 311–318.

Roth, R., Rambow, O., Habash, N., Diab, M., Rudin, C., 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In Proceedings of ACL-08: HLT, Short Papers, Columbus, Ohio, pp. 117–120.

Salloum, W., Habash, N., 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, Edinburgh, Scotland, pp. 10–21.

Salloum, W., Habash, N., 2013. Dialectal Arabic to English machine translation: pivoting through modern standard 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.

Salloum, W., Elfardy, H., Alamir-Salloum, L., Habash, N., Diab, M., 2014. Sentence level dialect identification for machine translation system selection. In Proceedings of ACL-2014, Short Papers.

Smrž, O., 2007. ElixirFM—implementation of functional Arabic morphology. In Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, Prague, Czech Republic, ACL, pp. 1–8.