Segmentation for Domain Adaptation in Arabic

Mohammed Attia and **Ali Elkahky**
Google LLC
New York, USA
{attia,alielkahky}@google.com

Abstract

Segmentation serves as an integral part in many NLP applications including Machine Translation, Parsing, and Information Retrieval. When a model trained on the standard language is applied to dialects, the accuracy drops dramatically. However, there are more lexical items shared by the standard language and dialects than can be found by mere surface word matching. This shared lexicon is obscured by a lot of cliticization, gemination, and character repetition. In this paper, we prove that segmentation and base normalization of dialects can help in domain adaptation by reducing data sparseness. Segmentation will improve a system performance by reducing the number of OOVs, help isolate the differences and allow better utilization of the commonalities. We show that adding a small amount of dialectal segmentation training data reduced OOVs by 5% and remarkably improves POS tagging for dialects by 7.37% f-score, even though no dialect-specific POS training data is included.

1 Introduction

Processing of informal and dialectal data is increasingly becoming the focus of attention for many NLP tasks particularly due to the growing popularity of the various social media platforms and messaging apps which have transformed the way people interact and communicate with each other on daily basis and accelerated the pace of change of the language used on the web. Today, many people write in the language they speak, leading to the influx of informal and dialectal data with the huge challenges they pose, most prominently among them are the non-standard orthography (like repeated characters for emphasis), abbreviations, non-conventional syntactic structures, spelling variability as well as mispellings, and code-switching. These phenomena have been largely ignored in mainstream language processing models which mostly relied on (and also expected) standard, monolingual, clean, and edited texts.

Moreover, the emergence of intelligent personal assistant systems (such as Siri, Alexa, Cortana and Google Assistant) have created a paradigm shift in how people interact with smart devices. Instead of issuing key words searches and formal questions, they are now more tempted to speak casually with these systems using their everyday language, which lays a growing burden on virtual assistants to accommodate unconventional (and previously unseen) queries and requests.

In this paper we show how NLP applications can scale up their performance on dialectal data by integrating a basic and simple preprocessing step, i.e. segmentation. The process of segmentation is important for languages where the notion of word does not straightforwardly align with the common concept of a space-delimited string. Arabic is a clitic language, where syntactic units can attach to other lexemes, and segmentation means identifying and splitting these syntactic units from the main lexemes or from each others. This is not a deterministic process, as we need to tell, for example, whether the letter و wa is a conjunction as in وخالد wa-Khaled "and Khaled" or part of the internal word build-up as in وحيد wahid "Wahid".

This paper is structured as follows. Section 2 gives a brief account of the related research on standard and dialectal segmentation of Arabic. In Section 3 we introduce our segmentation annotation scheme, explaining the meaning of clitics and how different they are from affixes, and compare our annotation convention to other approaches. Section 4 gives the details of our work on dialectal data collection, explaining the challenges facing extraction, filtration and sampling. Section 5

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 119–129
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

spells out our hypothesis on how segmentation can help in domain adaptation and the approach we follow to test this hypothesis. In Section 6 we describe our parsing system and the features used. In Section 7 we explain our experimental setup and discuss the results, and finally Section 8 concludes.

2 Related Work

Segmentation of MSA has frequently been handled as part of a pipeline with multiple processes (including morphological analysis, and POS tagging). For example, MADA (Habash and Rambow, 2005; Habash et al., 2009) is a system that uses an SVM-based classifier to disambiguate the output of the Buckwalter morphological analyzer which conveniently also provided diacritization and English glosses. By contrast, AMIRA (Diab et al., 2004; Diab, 2009) is a lexicon-independent system for Arabic that conducts segmentation as well POS tagging and base-phrase chunking. Both systems are trained on the LDC's Arabic Treebank (ATB) data and both report an accuracy above 99%. The high accuracy is probably attributed to the high quality and low noise in this edited data.

Treating segmentation as a specialized task, Aliwy (2012) developed a hybrid system for Arabic segmentation trained on a manually-annotated dataset of 29k words extracted from the Al-Watan corpus and reports an accuracy score of 98.83%. Abdelali et al. (2016) developed a segmenter for their tool, Farasa, using SVM and trained on the ATB data with reported accuracy of 98.94%. Moreover, Mohamed (2018) developed a memory-based learning segmenter for Arabic religious texts trained on a manually annotated in-domain corpus of 27k words combined with the ATB data with reported accuracy of 95.70%.

Regarding Egyptian segmentation, Mohamed et al. (2012) developed a memory-based segmenter for Egyptian Arabic trained on manually-annotated user-generated data including 20k words combined with the ATB data and reported an accuracy of 91.90%. Habash et al. (2013) developed MADA-ARZ as an Egyptian extension to MADA, the MSA morphological processor. The approach they took was to replace the MSA analyzer SAMA with the ARZ analyzer CALIMA, and again disambiguate the output using an SVM classifier, and reported a segmentation adccuracy of 97.5%. Monroe et al. (2014) augment a previously developed character-level CRF-based segmenter for MSA with more features to accommodate Egyptian Arabic achieving an f-score of 92.09% on an Egyptian test data.

More recently Samih et al. (2017a) developed an Egyptian segmenter using neural architecture of Bi-LSTM with a CRF optimizer trained on a small dataset of 350 Egyptian tweets (8k words) and reported an f-score of 92.65%. They later extended their work to cover Gulf, Moroccan and Levantine Arabic (Samih et al., 2017b; Eldesouki et al., 2017).

3 Dialect Segmentation Convention

Clitics are prevalent and highly frequent in Arabic as they span a large class of function morphemes including conjunctions, negation, progressive and future particles, object and possessive pronouns, and the definite article. And these function morphemes attach to verbs (as in Table 1), nouns (as in Table 2), or other function words or morphemes.

3.1 Annotation Guidelines

Token	Sub-Type	Possible Values
Proclitic	Conj	و wa "and"
	Neg.	م ma "not" ⋆
	Compl.	ل li "to"
	Particle	ب bi "prog." ⋆
	Particle	سـ, sa, هـ ha ⋆ "will"
Stem		يحب yuhib "like"
Enclitic	Obj pron	هـ hu, و uw ⋆ "him"
	Post-Prep	ل li "to" ⋆
	PObj Pron	هـ hu, و uw "him" ⋆
	Post Neg	ش shi "not" ⋆

Table 1: clitics with a verb. Note that the progressive and future particles are in complementary distribution. ⋆ used in EG only.

The annotation guidelines are fairly straightforward. Here are the main instructions followed during annotation.

- Segment words in a way that would reflect the correct number of part of speech tags as in Tables 1 and 2 above.
- Words merged with other words should be separated, e.g. عبدالله "Abdullah"

ما+شاء+الله "God willing".

- When the post-preposition is fused with the last letter of the stem, the post-preposition should be retained at the expense of the stem, e.g. بـ+قول+لـ+ك "I am saying to you".
- Hashtags, emoticons and user names are treated as single units, e.g. #الحمد_لله "Thank_God", :-), and @mohamed_ali.
- Sometime letters are repeated for emphasis, in this case the token boundary is maintained, e.g. ووو+أخييييرا!!!! "annnnd finallllly".
- With spelling errors, we segment as if words are written correctly, e.g. رحلتة "his trip".
- Interrogative words and interjections are treated as one unit, e.g. مش "not" بلاش "don't" ليه "why".
- Some words are common, but nonetheless should be tokenized, e.g. م+ حد+ش "no one", م+ في+ش "nothing".
- When vowels on prepositions are changed from short to long, the long vowel is considered as part of the preposition, e.g. معا+ه "with him", لي+هم "to them" بي+هم "by them".

Token	Sub-Type	Possible Values
Proclitic	Conj	و wa "and"
	Prep.	ل li "to"
	Det.	ال Al "the"
Stem		كتاب kitAb "book"
Enclitic	Poss pron	هـ hu "his"

Table 2: clitics with a noun. Note that the determiner and the possessive pronoun are in complementary distribution.

3.2 Clitics vs. Affixes

Clitics are different from affixes in that prefixes and suffixes are **morphological markers** that indicate tense, number, person, gender, case, etc., while clitics are **syntactic units** (like prepositions, conjunctions, pronouns and particles) with separate part of speech functions, but happen to attached to other words. The difference is shown further by the example in the syntactic tree in Figure 1. Note how the verb retains the imperfective and plural markers, and the noun maintains the feminine marker.

Example: هيدفعوها للحكومة will-pay-it to-the-government (2 words = 6 token sentence)

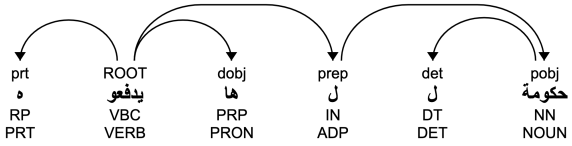

Figure 1: Clitics on a Syntactic Tree

Clitics can be challenging for intelligent virtual assistant applications dealing with Arabic in many areas. The problem is that without proper segmentation, it's impossible for the system to correctly identify the triggering phrase or the span of an argument, may it be a message, contact name, location, or artist name. Here are a few examples categorized by topic:

1. Planning:
   - فكرني بشراء اللبن fak~arni bi-$ira' Al-laban "Remind me to-buy milk".
     Type of attached argument: reminder subject
2. Communication:
   - اتصل بأحمد it~asil bi-Ahmed "Make a call to-Ahmed".
     Type of attached argument: contact name
3. Media:
   - أغنية لعمرو دياب >ugniyah li-Amr Diab "A song by-Amr Diab".
     Type of attached argument: artist
4. Device Control:
   - علي الصوت بأربع نقاط Eal~i al-Suwt bi->arbaE niqaT "Raise volume by-five points".
     Type of attached argument: numeric value
5. Local directions:

- المسافة بين البيت والعمل al-masafap bayn al-bayt wa-al-Eamal "distance between home and-work".
  Type of attached argument: location

3.3 The definite article dilemma

Arabic has only one determiner, the definite article ال Al "the". However, different conventions conflicted on whether to consider it as a morphological marker or a syntactic unit (clitic). While all other clitics have some free-form counterparts of their own category, e.g. و wa "and" (a bound conjunction), ثم vum~a "then" (a free conjunction), ب bi "in" (a bound preposition), and في fiy "in" (a free preposition), the definite article is unique in its category. It constitutes one of two ways by which a noun can be definite, the other being through idafa or compounding in a genitive/possessive sense, such as كتاب الطالب kitAb Al-Talib "the student's book". Therefore, the definite article bears similarities to both morphological markers and clitics. In a parse tree it can be either separated from the noun and be represented as a determiner headed by the noun, or stay merged with the noun and a feature called "state" is marked as 'definite'.

In the LDC Arabic treebank, the definite article is treated as morphological marker (i.e. not considered as a separate token), and therefore, most NLP applications based on this data model reflect this convention. In most other research efforts, such as (Abdelali et al., 2016; Aliwy, 2012; Mohamed, 2018; Habash et al., 2012), it is considered as a clitic and is segmented away from the nouns and adjectives they attach to.

It is also observed that the affinity of the determiner to the modified element changes by the type of the noun or adjective it is attached to. While it is perfectly separable with common nouns, e.g. الكتاب Al-kitAp "the-book" and الطالب Al-TAlib "the-student", it becomes more rigid with proper nouns such as, البرادعي Al-Baradei and السيسي Al-Sisi. However, the boundaries are not always clear and the distinction become somewhat blurry when a proper name has a composition meaning, such as شارع الاستقلال $ariE Al-<isotiqolAl "Street of (the) Independence", or a homograph, such as الخنافس Al-xanAfis "The Beatles" or "the beatles". Even with proper nouns where the definite article seems frozen, it needs to be omitted in cerain cases, partcularly when the noun is preceded by a vocative particle, e.g. يا سيسي yA sisi "O, Sisi".

3.4 Comparison of Segmentation Conventions

Our segmentation convention matches with (Aliwy, 2012; Mohamed, 2018; Habash et al., 2012) where clitics are split from words and the of notion of clitics is aligned as the syntactic units that can be assigned a POS tag and can occupy a node on the syntactic tree. It is also similar to the Penn Arabic Treebank (ATB) (Maamouri et al., 2004) with the exception of the definite article where we consider it as a clitic while in the ATB it is taken as a definiteness marker.

However, the segmentation scheme adopted here is significantly different from that of Farasa (Abdelali et al., 2016; Samih et al., 2017a,b; Eldesouki et al., 2017) in a number of ways. While Farasa segments all clitics as we do, they also split a number of additional morphemes as follows:

- The feminine marker is split from the noun, e.g. طالبة "student.fem" is split as طالب+ة. This convention, however, fails to recognize the fact that in Arabic the gender marker can indicate natural gender, as in the example above, or just a grammatical gender, such as ساعة "watch", حاجة "thing", and نسبة "ratio". Splitting the feminine marker in the later cases results in incomplete stems, or non-words.

- Dual and plural suffixes with nouns are split, such as كتاب+ين "book.dual", مدرس+ون "teacher.pl" and طالب+ات "student.fem+pl". The problem of oversegmentation shows again with the feminine plural with the grammatical gender, e.g. حاج+ات "things" and ساع +ات "watches". And while it normalizes stems for sound plurals, it leaves broken plurals unhandled, e.g. كتب kutub "books" the plural of كتاب kitAb "book".

- Number, gender, and person suffixes with verbs are split, such as ذهب+وا "went.pl" and ذهب+ت "went.fem". Farasa considers these suffixes as subject pronouns. However, this approach fails to acknowledge that Arabic is a pro-drop language, and the person, number,

and gender affixes are just added to permit the dropping of the subject and allow for its semantic reconstruction.

- Case marker suffixes with nouns are split, such as كتابها "book.acc". This is clearly an affix, and splitting it causes a problem with frozen adverbs, such as أيضها "also" and طبعها "naturally".

Therefore, as illustrated above, the Farasa convention is a midway between a stemmer, and segmenter. It is to be noted that a stemmer aims to split all affixes and suffixes regardless of their nature, while a segmenter splits only bound morphemes that are syntactic units (or clitics) in nature.

4 Data Collection and Analysis

4.1 Challenges of Dialectal Data Collection

There are over 22 Arab countries with 22 national dialects and even larger number of sub-dialects. The population ranges from around 100m to less than 1m. Natural Language Understanding (NLU) systems that perform well on MSA are likely to face difficulties dealing with the various dialects. As dialects are becoming the main medium of the interaction between the Intelligent Personal Assistants and the Arabic speakers, it is important to have well-scaled NLP tools, with a good segmenter as a starting point. Here we develop a generic process for data collection and sampling that can be applied to one or more dialects.

With data collection, there are a number of challenges that need to be taken into consideration.

- Intra-sentential code switching: some user-generated data can contain a mix between MSA and dialects or dialect and a foreign language..
- Pan-Arab pages. Some web pages are popular across the Arab world and can attract audience from different regions, and therefore, it is not immediately obvious what dialect the comment is written in.
- Expatriates. Gulf states have a large number of expatriates. In Saudi Arabia, for instance, there are 2m Syrians, 1m Sudanese, and 1m Egyptians. For another example, only 17% of residents in Dubai are Emiratis. Therefore, relying on the location of the user or the webpage alone can be misleading.
- Neighboring dialects. Within a particular region, dialects can be significantly similar. So, how can we separate Moroccan from Algerian, Saudi from Kuwaiti and Lebanese from Syrian?

4.2 Dialect Filtration

To handle the challenges mentioned above, our approach to dialectal data collection consists of a two-stage filtration process. We apply this process to four dialects (Egyptian, Saudi, Moroccan and Algerian). The reason for selecting these four dialects in particular is that we wanted to see how our method performs on dialects from discrete regions (Egyptian, Saudi and Moroccan) as well as dialects from neighboring countries (Moroccan and Algerian).

1. By locale. Detecting the location of the webpage and user who made the comment.
2. By seed-words. We construct dialect-specific word lists that contains high frequency, high confidence lexical items.

In the first filtration stage, we crawl data from local news websites as well as user-generated data (blogs, user comments, and social media posts) from the target countries: Egypt, Saudi Arabia, Morocco and Algeria.

We observe that user-generated data is outpacing edited data, and the makeup and structure of data on the web is rapidly changing. It seems that social media and sites allowing free comments and reviews are giving people unprecedented and mostly uncensored freedom and expressive power, which they seem to utilize effectively.

The second filtration stage is the development of dialect seed word. Lists of dialectal words available online are very limited in size, not well maintained, and have no information on frequency. Therefore, we extract our own wordlists from corpora. The assumption is that dialectal words will fail when matched against a standard lexicon. We randomly select 1m words from the data that we crawled, and we match them against an MSA lexicon primarily meant for spell checking (Attia et al., 2012).

We observe that the rate of unknown words in user-generated data ranges from 6% to 7%, and it

can go up to as high as 20% with purely colloquial data, such as regional tales. We assume that the unknown words are most likely to be dialectal. To check the validity of this assumption, we select unknown words and order by frequency. We focus on top frequency words as these are assumed to contain function and common words that fit as good candidates for a seed list. Then we manually analyze the top 100 words for the Egyptian user-generated data. Figure 2 shows that over half of the words are actually dialectal, the remaining words are either spelling errors or names entities or standard words that happen not to be found in the spell checking wordlist.

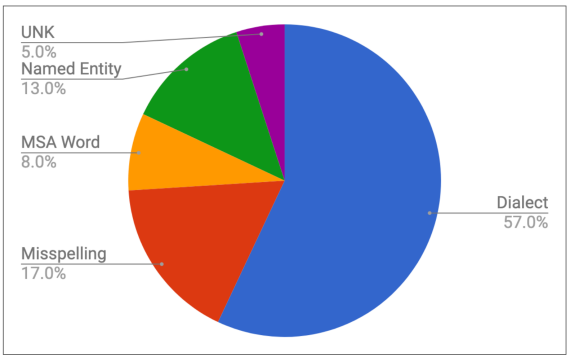

Figure 2: Analysis of Unknown Words in Corpora

4.3 Dialect Lexical Intersection

Having collected lists of potential seed words for the four target dialect and sorted them by frequency, now we try to evaluate how well can these seed words distinguish one dialect from the other. We test the distinctive nature of these lists by looking at the intersection between them with regards to the top 200 most frequent words. Figure 3 illustrates the results of the evaluation, where it shows two remarkable observations: 1) Dialects from different regions have lower intersection (below 20%), and 2) dialects from the same region have greater overlap (above 30%).

4.4 Data Sampling

Manual annotation of data is expensive and time consuming. Therefore, it is important to sample the data in such a way that we obtain the best possible coverage for the least possible amount of data. Data sampling is discussed in Active Learning as the need to strike the right balance between exploration and exploitation over the data space representation (Bouneffouf et al., 2014). The idea is that a system that only "exploits" will be too

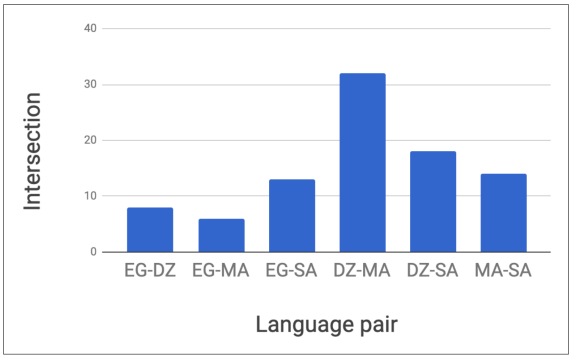

Figure 3: Lexical Overlap between Dialects. EG: Egyptian, DZ: Algerian, SA: Saudi, MA: Moroccan

specialized and unable to generalize, and a system that only "explores" does not improve its predictive power, and hence is the need to make the proper compromise between the two. In our sampling we try to select data that is representative, diverse, and lexically and syntactically varied. In order to achieve this goal, we rely on two criteria: sentence length and similarity matching.

1. Sentence Length. Different sentence lengths usually indicate different user fluency levels and represent different syntactic structures. We define 9 ranges for sentence length: 5–9, 10–14, 15–19, 20–24, 25–29, 30–34, 35–39, 40–44, 45–49. Then we extract an equal number of sentences from each length range. We excluded sentences shorter than 5 as they mostly included interjections and confirmation phrases, and longer than 49 as they include run-on sentences.

2. Similarity Matching. Exact repetitions, semi-repetitions, and similar sentences exist in any data collection, but they are particularly rampant in user-generated data. While it is straightforward to spot exact repetitions, or duplicates, and discard them, it is more challenging to identify similar sentences and to set out a threshold for this similarity, so that each sentence added to the annotation will ultimately carry an added-value to the system performance. There are mainly two paradigms for string matching: edit distance and longest common subsequence (LCS). Edit distance, as defined by Levenshtein (1966) tries to find one of three edit operations (insertion, deletion and substitution) when matching two strings. By contrast, the LCS (Wagner and Fischer, 1974) looks for

the longest subsequence that is common to two strings. To illustrate with an example, we evaluate these two stings using the two measures.

- شاهدت الحلقة الأولى. "I saw the first episode."
- شاهدت الحلقة الثانية. "I saw the second episode."

Using the edit distance, we obtain a similarity score[1] of 75.68% while with LCS, we get a score[2] of 81.08%, which means that LCS perceives the two sentences as more similar than the edit distance. By nature, the edit distance focuses on the differences, while LCS is more suited for finding similarities. Therefore we choose LCS (or SequenceMatcher) in our sampling method and set the threshold at 70%, so that any sentence that is similar to any existing sentence by this threshold or higher gets discarded.

5 Hypothesis and Approach

Dialects, by definition, are subsets of the standard language (or koiné) and they can easily, readily and freely draw from the larger repository. Therefore dialects should not be treated as separate and independent entities, but as a subtype that inherits from and extends a larger archetype. Dialects should be conceived of as the aggregate of the standard language and local variant.

Dialects diverge from the standard language and at the same time have a lot in common with this 'mother' language. Our hypothesis is that dialects can be accommodated fairly well without going through the lengthy and expensive acquisition of complete and new datasets, but through actively seeking and covering dialectal words, phrases and sentences as an add-on component that can be plugged in with the standard language.

Figure 4 demonstrates a prototype of our hypothesis showing the idea that if we inject specifically-targeted dialectal segmentation training data into the standard dataset and rebuild our model, we can achieve better support and coverage for dialects at a higher level of representation, namely POS tagging, by utilizing the shared lexicon and reducing the number of OOV's and without having any dialectal POS training data.

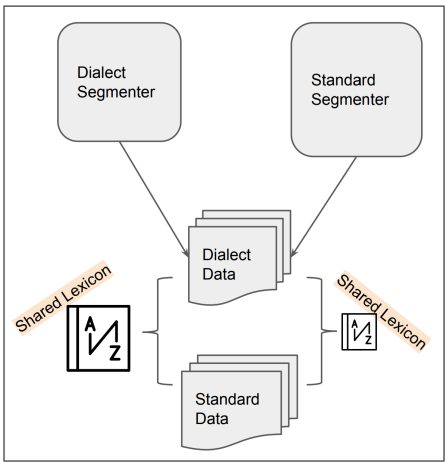

Figure 4: Anticipated Shared Lexicon Size

In order to test this hypothesis, we conduct the following three steps.

1. Manually annotate Egyptian dialectal segmentation data. After extracting and sampling the data, we manually annotate 1,058 sentences and split them into 739 sentences for training, 158 for validation and 157 for testing. Only the testing set is also annotated for POS tags besides segmentation.

2. Develop a segmentation model from the MSA data alone and another model from the combination of the MSA and dialectal data. The MSA data contains 9,717 sentences (399,774 tokens) and includes news articles (covering politics, sports, entertainment, business, health, sci-tech, arts), Wikipedia articles web articles (including blogs, forums, reviews).

3. Run the dialectal and standard segmenters on the dialect test set, evaluate how many words are shared with the dialect and MSA, and check the impact on the POS tagger. Hopefully the output of the model with dialectal data will have more shared lexicon with the Arabic standard dataset and improved POS tagging score.

6 System Description

In our experiments we use an arc-eager transition based dependency parser (Nivre, 2003) with a model trained using a linear SVM architecture similar to the one in Yamada and Matsumoto

[1] edit distance / (len(substr1)+len(substr2)/2) * 100

[2] As implemented in the SequenceMatcher in the difflib library

(2003). When experimenting with morphological features, we add the morphological attributes for both stack-top and buffer-top tokens.

Features:

- A window of +/- 3 characters of uni-grams and bi-grams around the current position.
- A tri-gram of current character of previous two character
- A tri-gram of current character of next two character
- whether the current character is punctuation
- whether the current character is a digit
- Word length and position within a word
- First and last two characters of the current word

Our segmenter is part of a dependency tree parser for Arabic. Computational implementation within the Dependency Grammars framework has been realized in the creation of dependency treebanks, such as the Prague Dependency Treebank (Hajič et al., 2001), the Stanford Dependencies (De Marneffe and Manning, 2008) and Universal Dependencies (Nivre et al., 2016; McDonald et al., 2013), and the development of dependency parsers, such as the Stanford parser (Chen and Manning, 2014), the inductive dependency parser (Nivre et al., 2004) and the MaltParser (Nivre et al., 2007).

A dependency parser complies with the Dependency Grammar formalisms. Within the Dependency Grammar, dependency relations can be represented either in a relational format or in a graph format. In a relational format, the representation is a triple which shows the relation between a pair of words. The head of the dependency relation is given as the first argument and the dependent as the second. This relationship is represented as follows:

```
relation(head, dependent)
```

For example, the sentence حضر الأولاد "the boys came" can be formulated as:

nsubj(حضر , أولاد) – det(أولاد, ال)

Similarly, in the graph representation the dependency arc points from the head category to the dependent category, and the relation (or grammatical function) is realized as a label on the arc as shown in Figure 5.

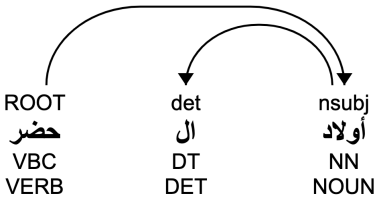

Figure 5: Sample Dependency Graphs

7 Experiments and Results

We have a high performance MSA segmenter, and when we adapt to the dialectal domain, we want to make sure that the performance on MSA data does not suffer from significant degradation. Therefore we build two models, one using the MSA data alone, and the other using MSA data combined with the Egyptian (EG) dialectal segmentation training data, and we evaluate both systems on the MSA and EG test sets.

Model trained on segmentation data from	Segmentation Eval	
	MSA	EG
MSA	97.91	82.56
MSA+EG	97.62	91.40

Table 3: Egyptian Segmentation Evaluation

As Table 3 shows, the model trained on MSA gives an F-1 score of 97.91% on the MSA test data and a remarkably lower score on the EG data (82.56%). For a task as basic as segmentation, this level of performance is not reliable to pass on to other downstream or upstream tasks such as IR or MT. When we train our model on the combined data of MSA+EG, there is a slight reduction in the performance on the MSA test set (about 0.3% absolute), while there a huge performance boost on the EG test set (8.84% absolute). The overall score on EG is 91.40%, which is not close to the performance on MSA data, but this is understandable given the small size of the training data, and it is still comparable to the scores reported in the literature: 91.90% by Mohamed et al. (2012), and 92.65% by Samih et al. (2017a). This also illustrates the need to invest in acquiring more annotated data for dialects.

Now we want to evaluate if this improvement on the EG segmentation will cascade up the processing pipeline and help the MSA POS tagger adapt to the dialectal domain. We run our POS tagger on three different segmentation inputs: predictions of the MSA segmenter, predictions of the MSA+EG segmenter, and gold segmentation. The reason we test on the gold segmentation is to see the headroom for improvement if we have a 'perfect' segmenter.

Model trained on segmentation data from	POS Eval	
	MSA	EG
MSA	94.36	66.70
MSA+EG	94.10	74.07
Gold data	96.66	81.33

Table 4: Egyptian POS Evaluation

Table 4 shows that the loss with MSA POS tagging from adding the new dialectal data is fractional (0.26% absolute). It also shows that using the MSA segmenter predictions as input, the POS tagger achieved only 66.70% f-1 measure on the EG test set. This has risen to 74.07% when using the MSA+EG segmenter predictions, a remarkable increase of 7.37% absolute. Improving the EG segmenter further can give a headroom up to 81.33%, which is another increase of 7.26% absolute. This is a significant improvement on the system performance that has been gained economically with few resources. This confirms our original hypothesis that segmentation can help with dialectal domain adaptation. One explanation of how the segmentation helps the POS tagging is that doing the right segmentation in EG data reduces the number of OOV tokens with respect to the POS tagging model, even when the POS tagger is trained with only MSA data. To verify that, we show , in Figure 7, the percentage OOV tokens for the POS tagger model when the data is segmented using the segmenter trained with MSA only, the MSA+EG segmenter or using the gold segmentation. MSA+EG segmeter reduced the OOV by 5% points absolute which is 25% relative reduction in OOV.

However, we observe that we cannot obtain POS tagging results for dialect comparable to MSA scores using segmentation alone. There will be a need for some in-domain POS training data, and we envision the optimal model of a parser training is to follow what we call a "data trapezoid", as shown in Figure 6. The data trapezoid has a wider base for segmentation training data, a medium base for POS tagging, and a narrower base for dependency annotation. As annotating data for POS and dependency is very costly and time-consuming, We believe that this model can achieve the right balance and compromise between resources to achieve reasonable system performance.

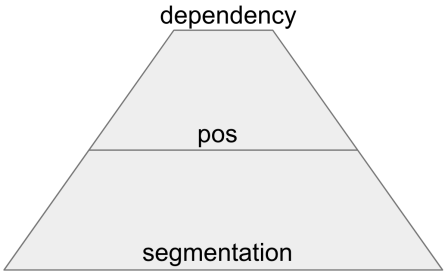

Figure 6: Data Trapezoid

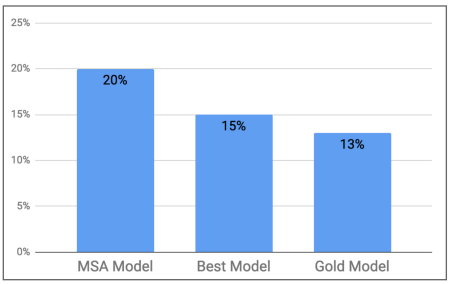

Figure 7: OOV percentage in POS Evaluation Data as segmented by different segmentation models

8 Conclusion

In this paper we have shown how segmentation helps in domain adaptation by scaling up the performance of a system trained on a standard language when it is applied to dialect. We showed how the injection of EG segmentation training data in a parser remarkably improves POS tagging despite the fact that no dialectal POS training data is included. From a few hundred dialectal segmentation sentences, we obtain a boost in POS tagging by 7.37% absolute. This does not per se eliminate the need for POS training data, but we suggest a data trapezoid model where there is a wide base of segmentation data, and a comparatively smaller amount of POS data and a yet smaller amount for dependency trees, a model that aligns with the time, effort and cost needed for each layer.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16.

Ahmed H. Aliwy. 2012. Tokenization as preprocessing for arabic tagging system. *International Journal of Information and Education Technology*, 2(4):348.

Mohammed Attia, Pavel Pecina, Younes Samih, Khaled Shaalan, and Josef Van Genabith. 2012. Improved spelling error detection and correction for arabic. *Proceedings of COLING 2012: Posters*, pages 103–112.

Djallel Bouneffouf, Romain Laroche, Tanguy Urvoy, Raphael Féraud, and Robin Allesiardo. 2014. Contextual bandit for active learning: Active thompson sampling. In *International Conference on Neural Information Processing*, pages 405–412. Springer.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Marie-Catherine De Marneffe and Christopher D. Manning. 2008. The stanford typed dependencies representation. In *COLING 2008: Proceedings of the Workshop on Cross-framework and Cross-domain Parser Evaluation*, pages 1–8. Association for Computational Linguistics.

Mona Diab. 2009. Second generation amira tools for arabic processing: Fast and robust tokenization, pos tagging, and base phrase chunking. In *2nd International Conference on Arabic Language Resources and Tools*, volume 110.

Mona Diab, Kadri Hacioglu, and Daniel Jurafsky. 2004. Automatic tagging of arabic text: From raw text to base phrase chunks. In *Proceedings of HLT-NAACL 2004: Short papers*, pages 149–152. Association for Computational Linguistics.

Mohamed Eldesouki, Younes Samih, Ahmed Abdelali, Mohammed Attia, Hamdy Mubarak, Kareem Darwish, and Kallmeyer Laura. 2017. Arabic multi-dialect segmentation: bi-lstm-crf vs. svm. *arXiv preprint arXiv:1708.05891*.

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012. A morphological analyzer for egyptian arabic. In *Proceedings of the twelfth meeting of the special interest group on computational morphology and phonology*, pages 1–9. Association for Computational Linguistics.

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.

Nizar Habash, Owen Rambow, and Ryan Roth. 2009. MADA+TOKAN: A toolkit for arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization. In *Proceedings of the 2nd international conference on Arabic language resources and tools (MEDAR), Cairo, Egypt*, volume 41, page 62.

Nizar Habash, Ryan Roth, Owen Rambow, Ramy Eskander, and Nadi Tomeh. 2013. Morphological analysis and disambiguation for dialectal arabic. In *HlT-Naacl*, pages 426–432.

Jan Hajič, Barbora Vidová-Hladká, and Petr Pajas. 2001. The prague dependency treebank: Annotation structure and support. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 105–114.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

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*, volume 27, pages 466–467. Cairo.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

Emad Mohamed. 2018. Morphological segmentation and part-of-speech tagging for the arabic heritage. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(3):22.

Emad Mohamed, Behrang Mohit, and Kemal Oflazer. 2012. Annotating and learning morphological segmentation of egyptian colloquial arabic. In *LREC*, pages 873–877.

Will Monroe, Spence Green, and Christopher D. Manning. 2014. Word segmentation of informal arabic with domain adaptation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 206–211.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. Maltparser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan T. McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666, Portorož, Slovenia.

Younes Samih, Mohammed Attia, Mohamed Eldesouki, Ahmed Abdelali, Hamdy Mubarak, Laura Kallmeyer, and Kareem Darwish. 2017a. A neural architecture for dialectal arabic segmentation. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 46–54.

Younes Samih, Mohamed Eldesouki, Mohammed Attia, Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, and Laura Kallmeyer. 2017b. Learning from relatives: unified dialectal arabic segmentation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 432–441.

Robert A. Wagner and Michael J. Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the 8th International Workshop on Parsing Technologies*, pages 195–206, Nancy, France.