Part-of-Speech Tagging for Arabic Gulf Dialect Using Bi-LSTM

Randah Alharbi[1,2], Walid Magdy[1], Kareem Darwish[3], Ahmed AbdelAli[3], Hamdy Mubarak[3]
[1]School of Informatics, University of Edinburgh
[2]College of Computer Science & Information Systems, Umm Al-Qura University
[3]QCRI, Hamad bin Khalifa University
raharbi@uqu.edu.sa, wmagdy@inf.ed.ac.uk
{kdarwish,hmubarak,aabdelali}@hbku.edu.qa

Abstract

Part-of-speech (POS) tagging is one of the most important addressed areas in the natural language processing (NLP). There are effective POS taggers for many languages including Arabic. However, POS research for Arabic focused mainly on Modern Standard Arabic (MSA), while less attention was directed towards Dialect Arabic (DA). MSA is the formal variant which is mainly found in news and formal text books, while DA is the informal spoken Arabic that varies among different regions in the Arab world. DA is heavily used online due to the large spread of social media, which increased research directions towards building NLP tools for DA. Most research on DA focuses on Egyptian and Levantine, while much less attention is given to the Gulf dialect. In this paper, we present a more effective POS tagger for the Arabic Gulf dialect than currently available Arabic POS taggers. Our work includes preparing a POS tagging dataset, engineering multiple sets of features, and applying two machine learning methods, namely Support Vector Machine (SVM) classifier and bi-directional Long Short Term Memory (Bi-LSTM) for sequence modeling. We have improved POS tagging for Gulf dialect from 75% accuracy using a state-of-the-art MSA POS tagger to over 91% accuracy using a Bi-LSTM labeler.
Keywords: Part-of-Speech (POS), Gulf Arabic (GA), Dialectal Arabic (DA), Bidirectional Long Short Term Memory (Bi-LSTM)

1. Introduction

Part-of-speech (POS) tagging is one of the main building blocks in many Natural language processing (NLP) applications (Jurafsky and Martin, 2009). POS tagging of Arabic is challenging due to its highly inflectional nature. Arabic language has two variants, namely: Modern Standard Arabic (MSA), the formal language that used in news and official business, and Dialectal Arabic (DA), the more informal version used in everyday life. Although they share some common characteristics, they differ at many linguistic levels (Katz and Diab, 2011). Most state of the art POS taggers for Arabic are designed and trained for MSA. Though the accuracy of MSA taggers is high (around 96%), these taggers fail to achieve high scores for DA (Pasha et al., 2014). For example, as we show in this work, a state-of-the-art Farasa (Abdelali et al., 2016) MSA tagger achieves only 75% accuracy on Gulf Arabic(GA) dialect. With the wide spread of social media websites and chatting applications, DA became widely used (Diab et al., 2010). There is a need to develop NLP tools and applications for them. Hence the need for designing a specific tool for POS tagging for DA is of utmost importance. This paper aims at developing a POS tagger for one of the most widely used dialects, namely Gulf Arabic (GA).

This work answers the question "how much gain in accuracy can be achieved by designing a dedicated DA POS tagger rather than utilizing MSA specific tools to adapt to dialects?". Our results show that we can achieve higher accuracy when DA POS tagger is used. Thus our Gulf POS tagger has achieved 91.2% accuracy for POS tagging GA using Bi-LSTM, which is 16% higher than the state-of-the-art MSA POS tagger.

The contributions of this work are as follows:

- We offer an annotated data set for GA POS tagging task along with annotation guidelines used, and we make it freely accessible for the research community[1].
- We asses the state of the art MSA POS tagger on Gulf dialect and provide analyses of failures and successes.
- We offer the first POS tagger for Gulf dialect that achieves an accuracy of 91.2%.

2. Background

In this section we will cover Arabic language characteristic, POS tagging for DA and similar work in literature.

2.1. Dialectal Arabic

Arabic language has two variants: MSA and DA. MSA is the primary language of news, media and education in the Arab world (Khoja, 2001). It is mostly written than spoken (Habash, 2010; Abuata and Al-Omari, 2015). DA is the language used in daily informal communication. It was mostly spoken than written, but it gradually became the mean of communication in social media (Darwish et al., 2012).

The Arabic language has many characteristics that make it challenging especially for NLP tasks. Mainly, there are three main categories for Arabic words: nouns, verbs and particles. Each one of them can be divided into subcategories which can be represented using up to 330,000 when choosing a detailed tag set(Habash, 2010). One of the main challenges of Arabic language is having multiple meanings and POS tags for the same Arabic form, especially when diacritics are absent. Diacritics are the symbols that represent Arabic short vowels and they are optional which introduce some ambiguity since there are words which have the same consonant letters but different part of speech and different pronunciations. Table1 presents differ-

[1] http://alt.qcri.org/resources/da_resources/

POS	Diacriticization	Gloss	Example sentence
Noun	دَوْرِي [**daw**.rɪy]	league	دوري الأبطال Champions League
Noun	دَورِي [**dow**.rɪy]	My turn	إنه دوري أنا It is my turn
Noun	دَورِي [**do**.rɪy]	My role	أديت دوري I fulfilled my role
Noun	دَورِي [**do**.rɪy]	My floor	إنه دوري It is my floor
Verb	دَوِّرِي [**daw.wɪ**.rɪy]	Search around	دوري هناك search around there
Verb	دُورِي [**du:**.rɪy]	Turn around	دوري يا دنيا Oh world turn around
Adjective	دَوْرِي [**daw**.rɪy]	Periodic	فحص دوري Periodic inspection
Adverbial phrase	دَوْرِي [**daw**.rɪy]	periodically	بشكل دوري periodically

Table 1: Example of one Arabic word that can have different pronunciations, meanings and different Parts-of-speech

ent forms of the word دوري [dwry] which can be diacriticised in many ways to form different meanings with different POS tags. Words in Arabic are formed by applying different patterns to a root in order to generate a stem (Zeroual et al., 2017). Patterns can indicate the words part of speech because it carries morphological information (Darwish et al., 2014). For example the pattern فعيل [faʕi:l][2] always indicates adjectives. Affixes (prefixes and suffixes) are attached to the stem. Prefixes can indicate information such as determination of a noun or tense of a verb (Boudlal et al., 2011). Suffixes can indicate gender and number. An example of Arabic complex segmentation is the word وسنجعلكم [wasanajaʕlkum] which means 'and we will make you'. It is segmented as و+س+نجعل+كم[wa+sa+najaʕl+kum] where نجعل [najaʕl] is the stem and each one of these segments is called clitic. For more explanation see (Darwish and Magdy, 2014; Habash, 2010).

Researchers usually consider five main dialects for DA, namely: Egyptian, Iraqi, Levantine, Maghribi, and Gulf (Samih et al., 2017). Although Gulf Arabic is the largest existing dialect in social media, there is very limited attention towards building NLP tools for it.

DA is derived from MSA; nevertheless, they differ at many linguistic levels. Some notable differences are in terms of:

- Vocabulary: Arabic dialects have richer vocabulary than MSA some of which are borrowed from other languages (Habash et al., 2012a).
- Word order: in dialects it is usually Subject-Verb-Object (SVO) while it is Verb-Subject-Object (VSO) in MSA(Diab and Habash, 2007).
- DA words are written as they are pronounced since there is no orthographic standards for dialects. This fact causes inconsistency in writing some words for example the word صدق [sˤɪdq], which means 'truth' is written as صج [sˤɪdʒ] in some Gulf dialects variants. Another result of writing words as they are pronounced is that some letters are dropped when pronounced. For example the word قاعد[qa:ʕɪd], which means 'he is sitting' is written as قاع [qa:ʕ], and the word طالع [tˤa:lɪʕ], which means 'look' is written as طاع [tˤa:ʕ] in Kuwaiti Gulf dialect.
- MSA has richer morphology than dialects for example most dialects do not have dual forms and do not differentiate among plural forms in terms of gender. Dialects have some affixes that do not exist in MSA such as, the prefix ح [ħa], which indicates the meaning of سوف [sawfa], which means 'I will' and the suffixes ج [ʒ] and س [ɪs], which indicate the meaning of the second person pronouns ك [ka] in Kuwaiti Gulf and Saudi Gulf, respectively.
- MSA has strict case ending rules in their grammars while dialects have no strict rules.

In this paper, we focus our study on GA, which is one group of dialects that share many characteristics. It is the dialect of countries surrounding the Arab Gulf, such as Saudi Arabia, Kuwait, Qatar, Bahrain, Oman, United Arab Emirates and Iraq. GA has additional characteristics that distinguish it from other dialects, for example:

- Phonologically: GA maintains the pronunciation of : ذ [ð], ث [θ] and ظ [ðˤ] unlike other dialects. Moreover, the sound ق [q] has different pronunciations e.g. قال [qa:l] , جال [dʒa:l] and كال [ka:l]] which means 'he said'(Khalifa et al., 2016).
- Morphologically: In most cases, there is no case inflection on GA words. Also, the prefix ب [ba] and the verb راح [raaħ] are used to indicate future tense. In addition, the words مب [mub], موب [mob], ما [ma:], ماهو [ma:hu] and ماهوب [ma:hu:b] are used for negation (Khalifa et al., 2016).

These differences emphasize the need for specially designed NLP tools for dialects to prevent the performance drop when using MSA tools.

[2]IPA is used to present Arabic words phonetically

2.2. POS Tagging for Arabic

The literature on DA POS tagging shows different approaches for producing dialects morphological analyzer and POS taggers. One is adapting MSA tools to dialects; the second is creating dialect specific tools. There are many strategies adopted by researchers to adapt MSA morphological analyzers for dialects. The work by (Duh and Kirchhoff, 2005) uses a list of possible POS tags produced by MSA morphological analyzer which is LDC-distributed Buckwalter stemmer to decide the tag and incorporates different methods to improve results. Their objective is to have minimally supervised POS tagging. The supervised system achieved 74.88% and the minimally supervised system achieved 68.48% after improvement. Another strategy is to pre-process the data by changing its representation (Habash and Rambow, 2006).MAGEAD(Habash and Rambow, 2006) is a morphological analyzer for MSA and Levantine family. MAGEAD used finite state machine on top of AT&T transducer. They changed the representation of the dialectal word. The new representation includes some features such as a root, a meaning index and morphological behavior class (MBC) which is considered variant independent. They only reported context recall 95.5% with 60% coverage of Levantine Arabic verb forms (Pasha et al., 2014). The second approach is to target dialect directly. CALIMA (Habash et al., 2012b) is a morphological analyzer that targets Egyptian dialect. It is rule-based and it has 4632 rules to predict the correct tags. Its accuracy on POS tagging task is 84%. MADAMIRA (Pasha et al., 2014) is also a morphological analyzer with two versions. One for MSA and another for Egyptian dialect. It is an amalgamation of the two analyzers AMIRA (Diab, 2009) and MADA (Rambow et al., 2009). They used SVM to predict the correct POS tag among all possible analyses produced by the analyzer. They achieved 92.4% on Egyptian data. The Egyptian version is slower than the MSA version because of the morphological complexity of the dialect. (Alsabbagh and Girju, 2012) proposed transformation-based Egyptian dialect POS tagger trained on Twitter Egyptian corpus. Functional based annotation scheme was used for POS tagging. They achieved 87.6% F-Measure scores. They did a pre-processing step to normalize the text in order to reduce spelling variations of dialectal words and speech effects while we used the input text as it is.

To the best of our knowledge there is no POS tagger for Gulf dialect and our work represents the first attempt to train a Gulf dialect POS tagger.

2.3. Bi-LSTM POS Tagging Relevant Work

Although not applied for dialect, but there are some works that used neural network approach, strictly speaking Bi-LSTM, for POS tagging (Ling et al., 2015; Plank et al., 2016; Darwish et al., 2017). This approach proved to have high accuracy scores even when used with rich morphology languages such as Turkish and Arabic (Plank et al., 2016). (Ling et al., 2015) worked on language modeling and POS tagging tasks for English. Their Bi-LSTM taggers achieved 97.36% without any features and 97.57% using some features which they did not specify. (Plank et al., 2016) proposed Bi-LSTM POS tagger and tested it on twenty-two languages including Arabic. They experimented with different word representations and the best representation for most languages was when combining word and character representation except for Arabic in which word representation was the best representation. They achieved 98.91% accuracy on MSA. Using word embedding combined with word and character representation achieved 98.87% which is less than when embedding is not used. (Darwish et al., 2017) used the same technique of (Ling et al., 2015) and proposed two word level features which were meta-type of the word and stem template for the word. Their Bi-LSTM tagger works at clitic level in which each word is segmented into its clitics using gold segmentation. Their best performing system achieved 96.1% accuracy when using both features and no word embedding.

3. POS Tagging Methodology

Part of speech tagging can be done in a supervised manner or unsupervised. There are many approaches to POS tagging: Rule-Based Approach, Markov Model Approach, Maximum Entropy Approach, Support Vector Machine(SVM) Approach and Neural Network Approach (Wilks, 1996). In this section we present our POS tagging approach; first we describe the set of features we extracted, then we discuss the two machine learning approaches we used, which are SVM and Bi-LSTM. It is worth mentioning that our taggers operate at clitic level instead of word level where a clitic is a word segment that has single POS tag.

3.1. SVM Based POS Tagger

SVM is used in many NLP classification tasks including POS tagging and proves to achieve high accuracy results with MSA (Darwish et al., 2017; Giménez and Màrquez, 2003). For this work, we used an SVM multi-class, specifically the SVM^{multiclass} tool developed by Thorsten Joachims (Joachims, 2008). SVM^{multiclass} uses regularization parameter C to prevent overfitting (Manning et al., 2009). Each tag of POS tags was considered as a class, and a set of features mentioned at the end of the section were extracted for each clitic and used to train the SVM classifier. In this work we use a combination of features that includes probabilistic, binary, and Arabic-specific features. For probabilistic features we used a combination of bigrams, trigrams, and 4-grams of tags and clitics. For binary features we used some features including meta-types of clitics, which indicate if a clitic is a number, a foreign word, a user mention or a URL. For Arabic specific features, we used stem template feature introduced by Abdelali et al. (2016). Where stem template represents the word pattern applied to the root mentioned in section 2.1 . The template for each clitic has been extracted and concatenated to word representation.

The set of used features for SVM are:

1. **Clitic features**: each unique clitic in our training set acted as a feature, and an additional feature is added to represent out-of-vocabulary (OOV) clitics. We experimented with three different values for clitic features. The first value is binary (whether it exists or not). The second is the log of clitic counts in training data. The

third is the Term Frequency-Inverse Document Frequency (TF-IDF) score for each clitic in which tweets are considered as documents.

2. **Probabilistic features**:
   - The probability of the co-occurrence of the tag and clitic $P(tag_i/clitic_i)$ and $P(clitic_i/tag_i)$
   - The probability of the previous tags bigram, trigram and four-gram $P(tag_i|tag_{i-1}), P(tag_i|tag_{i-1}, tag_{i-2})$ and $P(tag_i|tag_{i-1}, tag_{i-2}, tag_{i-3})$
   - The probability of the next tag bigram, trigram and 4-gram: $P(tag_i|tag_{i+1})$, $P(tag_i|tag_{i+1}, tag_{i+2})$ and $P(tag_i|tag_{i+1}, tag_{i+2}, tag_{i+3})$
   - The probability of the tag given the previous four and the next four clitics: $P(tag_i|clitic_{i-1}, clitic_{i-2}, clitic_{i-3}, clitic_{i-4})$
3. **Binary features**:
   - Meta-type of the clitic, whether it is a number, a foreign word, a mention, a hash tag, or URL.
   - Clitic position (initial, middle, end) of the word.
   - If a clitic is a prefix, a suffix or a stem.
   - Leading letters آأاإبت [different forms of Alif, ta], which can indicate that a clitic is a verb.
   - If the previous tag is a progressive particle or a determiner. So this will indicate if a clitic is a verb or a noun, respectively.

The values for probabilistic features are calculated using the Maximum Likelihood Estimation (MLE), while a non-zero value of 10^{-10} is assigned for unseen n-grams.

3.2. Bi-LSTM Based POS Tagger

Bi-LSTM is a special type of Recurrent Neural Network (RNN). It has proved to be a good choice for sequence modeling tasks (Ling et al., 2015) such as speech processing, POS tagging, phrased based chunking ... etc. It is also less sensitive to training data size (Plank et al., 2016). Moreover, Bi-LSTM can capture the context around source words up to very long sequences in both directions (previous and upfront) (Wang et al., 2015). It also does not need hand crafted features to work well. These characteristics make it a suitable fit for POS tagging of DA. Since there is not much training data available for DA – GA in this case – and since DA lacks standards to design powerful features, a model is needed that auto-fits its features and characteristics. Bi-LSTM structure differs from the classic RNN in that it adds a memory cell to the neural network architecture that learns to memorize information about a sequence for long periods of time. It also takes two passes of the input sequence, both forward and backward. For the sequence of clitics c_1, c_2.. c_m, where m is the number of clitics in a sequence, it manages clitic c_i by encoding information about c_1 ... c_{i-1} and c_m ... c_{i+1} sub-sequences. Bi-LSTM takes a sequence of features i.e. word representations, word embedding and any hand-crafted features. The feed-forward states of the network outputs the tag sequence for the previous clitics. The back-forward state holds tags information for the next clitics. The two states are combined using the following function:

$$l_i = tanh(L^f S_i^f + L^b S_i^b + b_l)$$

where L^f, L^b and b_l are parameters for combining the forward and backward states, S_i^f and S_i^b are the forward states and backward states respectively (Darwish et al., 2017). For implementation, we used Java Neural Network (JNN) toolkit for language modeling and part-of-speech tagging proposed by (Ling et al., 2015). In order to reach good generalization for any language processing task we need to have good word representations (Ling et al., 2015). In JNN, there are two representations: word representation and compositional character representation i.e. character-to-word (C2W). Word representation combined with C2W representation is called (CC2W+W). The input of our network is a sequence of features: clitic representations (word , C2W and CC2W+W), meta type, and/or stem template. The output will be a sequence of tag predictions in which each tag is formed by combining the forward and backward state of the network. JNN uses a tanh activation function. We include the stem template feature introduced by (Abdelali et al., 2016). The template for each clitic has been extracted and concatenated to the representation. Some clitics have no valid patterns e.g. ال [al] determiner. We also include meta-type feature which is an additional information added about the type of clitic i.e. to specify whether it is a number, an adjective number, a prefix, a suffix, a foreign, a punctuation, an Arabic letter and twitter specific types: hashtags, URLs and mentions.

4. Experimental setup

4.1. Data

We used gold annotated dataset which is built using gold segmented GA tweets taken from Samih et al. (2017). It consists of 343 Tweets with 6,844 tokens and 10,255 clitics. we used simplified Arabic Tree Bank (ATB) 18 tag sets proposed by (Darwish et al., 2017), but we neglected the abbreviation tag, since it is unlikely to be used in DA. Moreover, we added four new tags for twitter specific data which are MENTION, URL, HASH, and EMOT for twitter mentions, hyperlinks, hash tags and emotion punctuations respectively. The total number of POS tags is 21 tags.
We did manual annotation for the data according to the following guidelines:

- Each words clitic should be labeled with one tag.
- The number of tags of a word is equal to the number of segments for that word.
- If a stem can be classified as an adverb or an adjective, we consider it an adjective.
- We only label a stem as an adverb if it always appears as an adverb.
- Any loan or foreign word written in Arabic letters (transliterated) was labeled with its original tag in the foreign language. For example, واټ از زیس "what is this" was labeled as PART, V, PRON, respectively.

The data used for experiments are 233 tweets for the training set, 33 tweets for the development set and 77 tweets for the testing set. We have another dataset which we used to test the effect of having more data that enable the systems

to learn features accurately. Since the size of the GA training data was limited and due to the high overlap between GA and MSA, we opted to augment our data with MSA data. Specifically, we used a set of 20,000 tweets, which were gold-segmented into about 890,000 segments and we tagged them using Farasa. In order to combine them with Gulf data without losing dialectal characteristics, the Gulf data was replicated to be the same size as the MSA data. We refer to this dataset as "Gulf++".

4.2. Baselines

We apply two baselines to compare the performance of our POS tagger. The first is the simple majority class baseline, where all words are labeled with the most common POS tag, namely "NOUN". The second baseline is obtained by applying the Farasa POS tagger (Darwish et al., 2017) to our GA data. Since Farasa does not cover Twitter specific tags, we assumed the tagging results for these tags to be predicted correctly.

4.3. Evaluation Metric

In order to evaluate the performance of any POS tagger, there are several evaluation measures available. In this paper, we used accuracy to measure effectiveness. Accuracy is the most widely used metrics for POS tagging (Craig Hagerman, 2012). The accuracy of the POS tagger is the ratio of correctly tagged words of a test set of all words where a correct tag is the tag that matches the true tag annotated by humans (Jurafsky and Martin, 2009). Errors analysis will be conducted using confusion matrix which indicates how many times a tag is confused with other tags (Manning et al., 2009)

5. Evaluation and Discussion

Our majority-class baseline, which assigned the "NOUN" tag to all words achieved an accuracy of 21.16%, which indicates that the POS tagging for GA is not a trivial task. The second baseline that uses the state-of-the-art Farasa POS tagger achieved 75.13% accuracy. This suggests that GA is somewhat close to MSA; a similar conclusion reached in (Samih et al., 2017). However, it is still far behind the performance on MSA, which is over 96%. This motivates the design of a dialect-specific POS tagger.

5.1. SVM Approach

Table 2 summarizes the results of experiments with SVM. Three clitic features values were tested. The best performance on the Gulf test set was achieved using the binary feature values. The accuracy was 85.8%. The best performance of the Gulf++ test set was 86.0% using TF-IDF scores. Adding meta-type information clearly increased the performance of the SVM. This was expected because these types have strong indications of the word's tag. Extending the dataset to include the MSA training example enhanced the performance of all settings. The fact that the performance benefited from having more training data to certain limits (Joachims, 2002) corroborates this observation .

The best performing system among all experiments was the system with the following setting: using the Gulf++ training set, with TF-IDF for the clitics features values and using meta-type features. The system achieved 86.0% accuracy. Further analysis of the types of errors produced by the system was carried out using a confusion matrix. The most common error was confusing verbs with nouns by 28.03%. This might have been due to the absence of short vowels and diacritics since some clitics have the same consonant letters but different pronunciations. It might also have occurred as a result of preferring the noun tag for out-of-vocabulary words because it is more common. The next most common error was confusing pronouns with noun suffixes and vice versa, which formed 24.7% of system errors. This error is common because the list of suffixes and the list of attached pronouns are similar. The third person pronoun, ـه [ha] and ـة [ta] the marker of feminine nouns is an example of this. Speakers tend to write them interchangeably in writing dialect since no strict orthographic rules are available. The third common error was confusing adjectives with nouns by 12.5%. Table 3 lists system errors rate.

Dataset	Features	Accuracy
Gulf	binary	81.85
Gulf	Binary + meta-type	85.8
Gulf++	Binary + meta-type	85.8
Gulf	log	81.85
Gulf	log + meta-type	85.2
Gulf++	log + meta-type	85.7
Gulf	TF-IDF	78.1
Gulf	TF-IDF + meta-type	80.4
Gulf++	TF-IDF + meta-type	86.0

Table 2: SVM experiments results

Error Type	Percentage
V –> NOUN	28.03%
ADJ –>NOUN	12.5%
PRON –> NSUFF	13.3%
NSUFF –> PRON	11.4%
PART –> ADJ	8.9%
PART –>NOUN	3.03%
HASH –> NOUN	3.03%
NOUN –>V	2.7%
PROG_PART –>PREP	2.3%

Table 3: Most common errors for the best SVM system

5.2. Bi-LSTM Approach

Different representations were put into an experiment, namely: compositional character representation (C2W), word representation and a combination of both CC2W+W. In general, CC2W+W proved to have the highest accuracy among all representations. Word representation is next in accuracy, followed by C2W. In essence, C2W representation achieved its best performance when no feature was used and using word level features caused its accuracy to drop. It seems that characters had a stronger relation than word level relations. Conversely, word representation achieved higher accuracy when both word level features, meta-type and template, were used. Finally, CC2W+W representation benefited from all word level features. It seems that CC2W+W was best when the training set was large.

Features	Gulf			Gulf++		
	C2W	Word	CC2W+W	C2W	Word	CC2W+W
None	86.7	85.9	88.5	88.5	86.8	89.6
Meta-Type	86.3	87.5	88.9	88.6	88.3	90.5
Template	85.6	88.7	88.6	88.6	89.0	90.7
Meta-Type+Template	85.4	89.7	89.1	88.7	90.6	91.2

Table 4: Bi-LSTM experiments results

System	Meta-Type	Template	Accuracy
SVM (Gulf+Binary)	Yes	No	85.8
SVM (Gulf++, TF-IDF)	Yes	No	86.0
Bi-LSTM (word representation, Gulf)	Yes	Yes	89.7
Bi-LSTM (word representation, Gulf++)	Yes	Yes	90.6
Bi-LSTM (CC2W+W (Gulf++))	Yes	Yes	**91.2**

Table 5: Best performing systems in SVM and Bi-LSTM experiments

Error Type	Percentage
V –>NOUN	16.6%
ADJ –>NOUN	16.0%
PRON –>NSUFF	12.4%
NSUFF –>PRON	9.4%
NOUN –>V	7.7%
NOUN –>ADJ	5.9%
PART –>NOUN	4.7%
NOUN –>PART	2.3%
PREP –>PART	2.3%

Table 6: Most common errors for the best Bi-LSTM system

When the training set was small the word representation fared better (see Table 4).

Experimenting with features shows that the highest accuracy values were achieved when the meta-type features and the template feature were combined. When meta-type features were added, CC2W+W representation achieved the highest accuracy scores, followed by word representation (see Table 4). This was true because the meta-type was a set of features that was meaningful to the clitic itself, not to its characters sequence. The template feature was also more meaningful for the clitic level than for the character level. The template feature helped to overcome one of the most common errors made by Arabic POS taggers which is confusing adjectives with nouns. Hence there is an improvement of adjective tag accuracy from 67.4% to 71.6% for CC2W+W representation on Gulf data and for noun tag accuracy from 82.2% to 82.9%. The improvement was due to the fact that adjectives and nouns have consistent templatic forms.

Adding more data to the training set improved the results in all settings with and without features, different representations. For example the system benefited from enriching the training data on meta-types features because it had the chance to observe more meta-types features and learn their effects on tag prediction. This observation was supported in part by the fact that per-tag accuracies for our best Bi-LSTM system were 100% for MENTION, HASH, NUM and URL tags, in which all had corresponding meta-type features.

The highest system was achieved with the following settings: CC2W+W representation, meta-type and template features and on Gulf++ dataset. The system accuracy was 91.2%, out-of-vocabulary accuracy was 73.5%. And the scores for precision, recall and F-score are 83.9%, 91.2% and 87.4% respectively. The errors trends were the same as the SVM errors. The most common error was confusing verbs with nouns with 16.6%. The second was confusing adjectives with nouns with 16.0%, followed by confusing pronouns with nouns suffixes, 12.4%. Table 6 gives a summary of the systems confusion matrix.

5.3. Discussion

To summarize the previous analyses, SVM is fast and achieves good result with a basic set of features, while Bi-LSTM is slower but can achieve higher accuracy levels without using any features. Still the effect of a good combination of word representation and features combination can enhance the results to a great extent. This fact is corroborated by the findings of Darwish et al. (2017), Plank et al. (2016), and Ling et al. (2015) in which all Bi-LSTM taggers benefited from features. Unlike the results of (Darwish et al., 2017) in which the SVM tagger outperformed the Bi-LSTM tagger by 0.1%. Our results show that Bi-LSTM outperforms SVM. This may be due to the fact that the Gulf dialect has no well-known grammatical standards or orthographic rules. Knowing the standards of MSA helps with feature engineering for those standards and characteristics, which helps in improving the accuracy of SVM. In the same vein, Bi-LSTM can capture non-lexical relations and dialectal trends and model them well without the need for highly dialectal features. Generally, both systems outperform Farasa, which supports our hypothesis that there should be specially designed tools to manage DA.

Table 5 summarises the results of the best systems among all experiments. Although the clitic level features were shared by MSA and Gulf, they improved the performance. Adding more detailed features can enhance the results.

6. Conclusion

In this work, we test the performance of a state-of-the-art MSA POS tagger on Gulf Arabic. The tagger achieved an

accuracy of 75% only. This motivated the design of a GA POS taggers using two approaches of POS tagging.

First, we designed an SVM tagger. A set of features was put into the test; their effect on accuracy was reported. Second, we examined a Bi-LSTM POS tagger. In both approaches we tested for the effect on accuracy of adding more data to the training set. A dataset for GA POS tagging task was prepared for use and made accessible to the research community. The best performance of Bi-LSTM is 91.2% using CC2W+W representation and meta-types and template features. On the other hand, the best performance of SVM is 85.96% by setting the clitic feature value to TFIDF and using meta-types features. Both systems achieved their highest accuracy when trained on the Gulf++ dataset.

However, Bi-LSTM outperforms SVM in most of its settings. The accuracy of Farasa on our Gulf dataset (75%) indicates that Gulf Arabic is close to MSA to some extent. The accuracy boost we could achieve supported our assumption that we need specifically designed and trained tools for DA.

Our future work includes investigating and adding more dialect data to the training set rather than MSA data; consulting dialect linguistic resources to engineer more informative features for SVM and Bi-LSTM.

7. Acknowledgements

We would like to thank Prof. Awwad Alahmadi for his insightful discussion and examples regarding Arabic linguistics and for taking care of the IPA transcriptions.

8. References

Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). Farasa: A Fast and Furious Segmenter for Arabic. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 2016:11–16.

Abuata, B. and Al-Omari, A. (2015). A rule-based stemmer for Arabic Gulf dialect. *Journal of King Saud University - Computer and Information Sciences*, 27(2):104–112.

Al-sabbagh, R. and Girju, R. (2012). A Supervised POS Tagger for Written Arabic Social Networking Corpora. *Proceedings of KONVENS 2012 (Main track: oral presentations), Vienna, September 19*, 2012:39–52.

Boudlal, A., Belahbib, R., Lakhouaja, A., Mazroui, A., Meziane, A., and Bebah, M. (2011). A markovian approach for Arabic root extraction. *International Arab Journal of Information Technology*, 8(1):91–98.

Craig Hagerman. (2012). Evaluating the Performance of Automated Part-of-Speech Taggers on an L2 Corpus. pages 29–42.

Darwish, K. and Magdy, W. (2014). Arabic Information Retrieval. *Foundations and Trends® in Information Retrieval*, 7(4):239–342.

Darwish, K., Magdy, W., and Mourad, A. (2012). Language processing for arabic microblog retrieval. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2427–2430. ACM.

Darwish, K., Abdelali, A., and Mubarak, H. (2014). Using Stem-Templates to improve Arabic POS and Gender/Number Tagging. *International Conference on Language Resources and Evaluation (LREC-2014)*, pages 2926–2931.

Darwish, K., Mubarak, H., and Abdelali, A. (2017). Arabic POS Tagging : Don ' t Abandon Feature Engineering Just Yet. *WANLP 2017 (co-located with EACL 2017)*, pages 130–137.

Diab, M. and Habash, N. (2007). Arabic dialect processing tutorial. *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Tutorial Abstracts on XX - NAACL '07*, (April):5–6.

Diab, M., Habash, N., Rambow, O., Altantawy, M., and Benajiba, Y. (2010). COLABA: Arabic dialect annotation and processing. *LREC Workshop on Semitic Language Processing*, (January 2016):66–74.

Diab, M. (2009). Second Generation AMIRA Tools for Arabic Processing : Fast and Robust Tokenization , POS tagging , and Base Phrase Chunking. *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, pages 285–288.

Duh, K. and Kirchhoff, K. (2005). POS Tagging of Dialectal Arabic : A Minimally Supervised Approach. *Computational Linguistics*, (June):55–62.

Giménez, J. and Màrquez, L. (2003). Fast and accurate part-of-speech tagging: The SVM approach revisited. *Ranlp*, pages 153–163.

Habash, N. and Rambow, O. (2006). MAGEAD: a morphological analyzer and generator for the Arabic dialects. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (Coling-ACL)*, M:681–688.

Habash, N., Diab, M., and Rambow, O. (2012a). Conventional Orthography for Dialectal Arabic. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, (January 2012):711–718.

Habash, N., Eskander, R., and Hawwari, A. (2012b). A Morphological Analyzer for Egyptian Arabic. *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology SIGMORPHON2012*, pages 1–9.

Habash, N. Y. (2010). *Introduction to Arabic Natural Language Processing*, volume 3. Morgan & Claypool Puplisher.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*, volume 29. SPRINGER SCIENCE+BUSINESS MEDIA. LLC.

Joachims, T. (2008). SVM-Multiclass: Multi-Class Support Vector Machine.

Jurafsky, D. and Martin, J. H. (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. *Speech and Language Processing An Introduction to Natural Language Processing Computational Linguistics and Speech Recognition*, 21:0–934.

Katz, G. and Diab, M. (2011). Introduction to the Special Issue on Arabic Computational Linguistics. *ACM Transactions on Asian Language Information Processing*, 10(1):1–4.

Khalifa, S., Habash, N., Abdulrahim, D., and Hassan, S. (2016). A Large Scale Corpus of Gulf Arabic. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4282–4289.

Khoja, S. (2001). APT : Arabic Part-Of-speech Tagger. *Proceedings of the Student Workshop at NAACL*, pages 20—-25.

Ling, W., Luis, T., Marujo, L. L., Astudillo, R. F., Amir, S., Dyer, C., Black, A. W., Trancoso, I., Fermandez, R., Amir, S., Marujo, L. L., and Luís, T. (2015). Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (September):1520–1530.

Manning, C. D., Ragahvan, P., and Schutze, H. (2009). An Introduction to Information Retrieval. *Information Retrieval*, (c):1–18.

Pasha, A., Al-badrashiny, M., Diab, M., Kholy, A. E., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. M. (2014). MADAMIRA : A Fast , Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. *Proceedings of the 9th Language Resources and Evaluation Conference (LREC'14)*, pages 1094–1101.

Plank, B., Søgaard, A., and Goldberg, Y. (2016). Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-16)*, pages 412–418.

Rambow, O., Habash, N., Rambow, O., and Roth, R. (2009). MADA + TOKAN : A toolkit for Arabic tokenization , diacritization , morphological disambiguation , POS Tagging, Stemming and Lemmatization. (January 2017).

Samih, Y., Eldesouki, M., Attia, M., Darwish, K., Abdelali, A., Mubarak, H., and Kallmeyer, L. (2017). Learning from Relatives : Unified Dialectal Arabic Segmentation. (CoNLL):1–10.

Wang, P., Qian, Y., Soong, F. K., He, L., and Zhao, H. (2015). A Unified Tagging Solution: Bidirectional LSTM Recurrent Neural Network with Word Embedding. *arxiv: EMNLP-reject*.

Wilks, Y. (1996). Natural language processing. *Commun. ACM*, 39(1):60–62.

Zeroual, I., Lakhouaja, A., and Belahbib, R. (2017). Towards a standard Part of Speech tagset for the Arabic language. *Journal of King Saud University - Computer and Information Sciences*, 29(2):171–178.