© 2018 Cambridge University Press

Effective Multi Dialectal Arabic POS Tagging

Kareem Darwish[1], Mohammed Attia[2], Hamdy Mubarak[1], Younes Samih[1], Ahmed Abdelali[1]
Lluís Màrquez[1], Mohamed Eldesouki[1] and Laura Kallmeyer[3]
[][1]Qatar Computing Research Institute. Hamad Bin Khalifa University, Doha. Qatar.
[2]Google Inc New York, New York, NY, USA.
[3]Computational Linguistics Dept., Heinrich-Heine-University D$\ddot{u}sseldorf, 40204D\ddot{u}sseldorf.Germany.$
[1]{*kdarwish, hmubarak, ysamih, aabdelali, lmerquez, mohamoha* }*@hbku.edu.qa,*
mattia@google.com, kallmeyer@phil.uni-duesseldorf.de

(*Received 25 October 2018; revised 30 November 2018*)

Abstract

This work introduces robust multi-dialectal part of speech tagging trained on an annotated dataset of Arabic tweets in four major dialect groups: Egyptian, Levantine, Gulf, and Maghrebi. We implement two different sequence tagging approaches. The first uses Conditional Random Fields (CRF), while the second combines word and character-based representations in a Deep Neural Network with stacked layers of convolutional and recurrent networks with a CRF output layer. We successfully exploit a variety of features that help generalize our models, such as Brown clusters and stem templates. Also, we develop robust joint models that tag multi-dialectal tweets and outperform uni-dialectal taggers. We achieve a combined accuracy of 92.4% across all dialects, with per dialect results ranging between 90.2% and 95.4%. We obtained the results using a train/dev/test split of 70/10/20 for a dataset of 350 tweets per dialect.

1 Introduction

Part of Speech (POS) tagging is the task of automatically assigning syntactic category labels to tokens in text and is an important preprocessing step for higher order NLP tasks such as syntactic parsing (Jurafsky and Martin, 2009). Most Arabic POS tagging work has focused on Modern Standard Arabic (MSA), which is used in formal communication, while work on POS tagging of Dialectal Arabic (DA), which is ubiquitous on social media and informal communication, has lagged behind. DA's informality leads to the prevalence of spelling variations (often creative), transliterated foreign words, and the use of social media artifacts such as emoticons and hashtags. Since Arabic speakers typically use dialects in their daily interactions, dialects became their natural choice in online conversations, replacing or complementing MSA. Arabic dialects are broadly classified into five major dialect groups, namely Egyptian, Levantine, Maghrebi, Iraqi, and Gulf. Each dialect group comprises a number of sub-dialects. For instance the Maghrebi group covers Libyan, Moroccan, Tunisian, Algerian, and Mauritanian, creating a continuum rather than discrete dichotomous variations. Dialects often differ much in lexical choices and may exhibit syntactic and morphological differences. The larger and the more diverse the lexical, morphological, phonetic, and syntactic differences between dialects, the less mutually intelligible they become. Prior POS tagging work on such informal and dialectal social media text has been scant due in large part to the scarcity of annotated data. Available work is mostly uni-dialectal with Egyptian receiving the most attention (Duh and Kirchhoff, 2005; Habash *et al.*, 2013; Khalifa *et al.*, 2017).

This work introduces robust multi-dialectal POS tagging trained on a annotated dataset of Arabic tweets in four major Arabic dialects: Egyptian (EGY), Levantine (LEV), Gulf (GLF), and Maghrebi (MGR) (Darwish *et al.*, 2018). We employ two different approaches. The first uses Conditional Random Fields (CRF), while the second combines word and character-based representations in a Deep Neural Network (DNN) architecture. We also exploit millions of unlabeled tweets to obtain word clusters and clitic-level embeddings, and we employ a variety of linguistic features, such as clitic metatypes and stem templates. We develop robust joint models that tag multi-dialectal tweets and outperform uni-dialectal models. We achieve an average word-level accuracy of 92.4% across all dialects. The success of the joint models hinges on effective automatic dialect identification that we develop using a DNN model, which is at par or better than the current state-of-the-art, distinguishing between 5 dialects with 86.3% accuracy at tweet level. The contributions of the paper are as follows:

- We compare the use of linear CRF sequence labeling and DNN with a variety of features to leverage limited training data.
- We build joint POS tagging models that can effectively tag tweets in multiple dialects. In the process, we develop effective dialect identification that we embed within our models. We plan to publicly release our source-code and models.
- We show that state-of-the-art results can be achieved for POS tagging using a small annotated dataset.

2 Background

Some recent papers have focused on POS tagging of English social media text, particularly tweets. Gimpel *et al.* (2011) used a CRF based sequence labeler in conjunction with a variety of features, such as word distributional similarity and phonetically normalized forms. In follow-on work, they improved their POS tagging accuracy from 89.3% to 92.2% by making use of Brown clusters (Brown *et al.*, 1992), which are hierarchical clusters of words based on the contexts in which they appear. The usefulness of Brown clusters for POS tagging was also demonstrated in (Owoputi *et al.*, 2013; Stratos and Collins, 2015). Similarly we show the effectiveness of using Brown clusters in the context of our CRF and DNN models. Derczynski *et al.,* (2013) attempted to improve POS tagging of tweets by specifically targeting low frequency words, which are often misspellings or creatively spelled words.

Work on POS tagging of Arabic social media text is scant. The scarcity of dialectal resources has hampered research on Dialectal Arabic (DA). Few resources were made available by programs such as TIDES, GALE and BOLT and distributed by LDC, which are not widely accessible due to their license requirements. As such, researchers used ad-hoc resources or small datasets that were curated locally and not widely accessible. Graja *et al.* (2010) created the Tunisian Dialect Corpus Interlocutor (TuDiCoI) with 893 utterances and 3,404 words from dialectal conversations between Tunisian railway staff. Bouamor *et al.,* (2014) used a collection of 2,000 sentences in Egyptian dialect as a seed to build a multi-dialectal Arabic corpus. The seed sentences were translated by native speakers into their own dialects to create a parallel multi-dialectal corpus in addition to English. Cotterell and Callison-Burch (2014) extended the work of Al-Sabbagh and Girju (2010), Zaidan and Callison-Burch (2011) to build a collection of commentaries from five Arabic newspapers and tweets that was used for automatic dialect identification. Duh and Kirchhoff (2005) used CallHome Egyptian Colloquial Arabic to build a POS tagger for Egyptian and achieved an accuracy of 69.83%. Habash *et al.* (2013) released a new adaptation for MADA (Roth *et al.*, 2008), that can also process dialectal Egyptian. Darwish *et al.* (2018) recently released a dataset of POS tagged tweets that cover four different Arabic dialect groups. We use this dataset in this paper, and we describe it in greater detail in the next section.

3 Data Description

We used the POS tagged dialectal Arabic dataset that was released by Darwish *et al.* (2018). The dataset includes 350 tweets for four major Arabic dialects that were manually segmented and POS tagged as is without applying any spelling standardization (Darwish *et al.*, 2018), such as CODA (Habash *et al.*, 2012). For example the word *mtbSlw$*[1] "do not look at him" is segmented as *m+tbS+l+w+$* and tagged as: PART+V+PREP+PRON+NEG_PART. The data is split into 5-fold partitions for cross-validation with 70/10/20 train/dev/test splits for each dialect. For comparison, we manually segmented and POS tagged an additional 350 MSA tweets, which differ in dialect but match in genre. To find MSA tweets, we used 30 very strong MSA words to filter millions of Arabic tweets. These words are mainly function words such as relative pronouns *Al*yn* and *Al*y* (who), demonstrative pronouns *h*A* and *tlk* (this, that), question words *mA*A* and *lmA*A* (what, why), and adverbs *bynmA* and *TAlmA* (while, so long as). We randomly selected 100 tweets containing each word. Then from those, we randomly selected 350 tweets as a sample of MSA tweets. The dataset size is as follows:

[1] Buckwalter transliteration is used in the paper

Dialect	Tweets	Words	Clitics
MSA	350	8,082	12,496
Egyptian (EGY)	350	7,481	11,602
Levantine (LEV)	350	7,221	11,015
Gulf (GLF)	350	6,767	10,181
Maghrebi (MGR)	350	6,400	9,408

The dataset is tagged using the Farasa POS tagset (Darwish *et al.*, 2017), which has 18 tags for MSA, 2 dialect-specific tags (PROG_PART, and NEG_PART), and 4 tweet-specific tags (HASH, EMOT, MENTION, and URL) (Darwish *et al.*, 2018). Table 1 shows examples of the dialectal and tweet-specific tags.

POS	Description	Example
NEG_PART	Negation Part.	ماقلناش (mAqlnA**$** – "we did not say")
PROG_PART	Progressive Part.	كتسالي (**k**tsAly – "he is finishing')
EMOT	Emoticon/Emoji	^_^
HASH	Hashtag	#Lebanon
MENTION	Mention	@ANimer
URL	URL	http://t.co/gbtT3

Table 1. *Dialect and tweet-specific POS tags*

Words are white-space and punctuation separated while hashtags, emotions, mentions and URL's are considered as single units without internal segmentation. Data is formatted in CoNLL format: Words are split into tokens or clitics, which are syntactic units (such as prepositions, conjunctions, determiners, pronouns and particles) that happen to attached to words. For example, the word وأحب w¿Hb "and I like" is split into two tokens where the conjunction و "and" is separated from the word. POS is provided in the data at the token level. In our annotation scheme, tokens, words, and sentences are separated by token boundary tag (TB), word boundary tag (WB), and end of sentence tag (EOS) respectively as shown in Table 2.

Figure 1 shows that dialects display more lexical diversity than MSA, with more clitics having more than one possible tag. Joining dialects increases ambiguity with 48% of tokens having more than one POS tag.

4 Learning: Methods and Features

4.1 Learning Methods

We used two learning methods with different paradigms, namely: CRFs and DNNs. **Linear Chain CRF:** The effectiveness of CRFs (Lafferty *et al.*, 2001) was shown

Index	Token	POS
0	و (w) “’and’	CONJ
0	TB	TB
0	أحب (¿Hb) “I like”	V
0	WB	WB
1	أسمع (¿smE) “I listen”	V
1	WB	WB
..	..	..
n	EOS	EOS

Table 2. *Data format for segmentation and POS tagging*

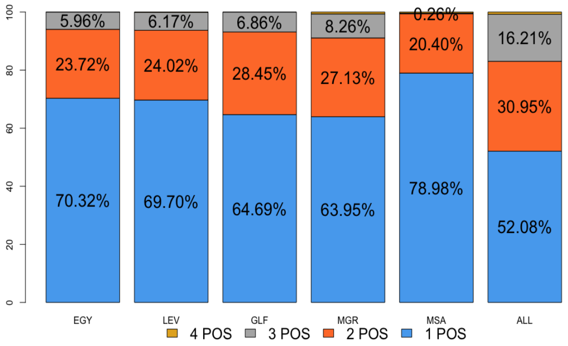

Fig. 1. Ambiguity in POS tags: % of clitics with 1 or more tags per dialects and in combination.

by Darwish *et al.,* (2018) in POS tagging on this dataset. We replicated their setup, which uses CRF++ implementation of a CRF sequence labeler with L2 regularization and default value of 10 for the generalization parameter “C”.[2] CRFs combine state-level and transition features and are simple, well-understood, and usually provide efficient models with close to state-of-the-art results.

Deep Neural Network: We used the DNN model depicted in Figure 2, which is well suited for sequence tagging. It is a variant of the bi-LSTM-CRF architecture proposed by Reimers and Gurevych (2017); Ma and Hovy (2016); Lample *et al.* (2016); Huang *et al.* (2015).[3] It combines a double representation of the input words by using word embeddings and a character-based representation (with CNNs). The input sequence is processed with bi-LSTMs, and the output layer is a linear chain CRF. The model uses:

Clitic-level embeddings allow the learning algorithms to use large unlabeled data to generalize beyond the seen training data. We explore randomly initialized em-

[2] `https://github.com/taku910/crfpp`

[3] Our implementation is mostly inspired by the work of Reimers and Gurevych (2017).

beddings based on the seen training data and pre-trained embedding.

Character-level CNNs have proved effective for various NLP tasks due to their ability to extract sub-word information (ex. prefixes or suffixes) and to encode character-level representations of words (Collobert *et al.*, 2011; Chiu and Nichols, 2016; dos Santos and Guimarães, 2015).

Bi-LSTM Recurrent neural networks (RNN) are well suited for modeling sequential data, achieving ground-breaking results in many NLP tasks (e.g., machine translation). Bi-LSTMs (Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) are capable of learning long-term dependencies and maintaining contextual features from both past and future states while avoiding the vanishing/exploding gradients problem. They consist of two separate bidirectional hidden layers that feed forward to the same output layer.

CRF is used jointly with bi-LSTMs to avoid the output label independence assumptions of bi-LSTMs and to impose sequence labeling constraints as in Lample *et al.* (2016).

The architecture of our model, shown in Figure 2, is applied to the example word "mA+dxl+tw+$" (you did not enter) to predict its POS tags. For each clitic, the CNN computes the character-level representation with character embeddings as inputs. Then the character-level representation vector is concatenated with both clitic embeddings vector and feature embedding vectors to feed into the bi-LSTM layer. Finally, an affine transformation followed by a CRF is applied over the hidden representation of the bi-LSTM to obtain the probability distribution over all the POS labels. Training is performed using stochastic gradient descent with momentum of 0.9 and batch size equal to 5. Given the relatively small POS datasets, we employ dropout (Hinton *et al.*, 2012) and early-stopping (Caruana *et al.*, 2000) to mitigate overfitting. We tuned our hyper-parameters on the development dataset using random search. We used the following hyper-parameters:

Layer	Hyper-Parameters	Value
Characters CNN	window size	3
	number of filters	40
Bi-LSTM	state size	200
	initial state	0.0
Dropout	dropout rate	0.5
Characters Emb.	dimension	100
Clitics Emb.	dimension	300
	batch size	5
	learning rate	0.01
	decay rate	0.05

4.2 Features

For both learning methods, we experimented with three features: two features that were shown to be effective for dialectal POS tagging by Darwish *et al.* (2018), namely token metatypes and stem templates; and the third is Brown clusters.

Metatypes (MT) include 10 types of tokens that are heuristically determined, namely: #Hashtag; @Mention; URL; Emoticon/emoji (dictionary-based); Retweet (“RT”); Foreign (all Latin characters); Number (numerals or spelled out numbers); Punctuation; Arabic (all Arabic letters); and Other.

Stem templates (ST) are morphological patterns that are used to derive stems from roots, such as the stem “lAEb” (player) which is derived from the root “lEb” using the stem template “CACC”. We used Farasa (Abdelali *et al.*, 2016) to determine stem templates.

Brown clusters (BC) Brown clustering is a hierarchical clustering of words based on their context (Brown *et al.*, 1992) and produces a kind of word embeddings. Similar words, particularly those with the same POS tag, tend to appear in similar contexts. BCs can be learned from large unlabeled texts and have been shown to improve POS tagging, specially for small training sets (Owoputi *et al.*, 2013; Stratos and Collins, 2015). To obtain BCs, we first collected a set of 5 million tweets for each dialect by filtering tweets based on Twitter users’ stated locations. For instance, to obtain Maghrebi tweets, we collected those that matched geographical locations such as Morocco, Casablanca, Algeria, Tunisia using their Arabic, English, and French names. Similarly, we filtered tweets that match locations in which the other dialects are spoken. Though geographical filtration does not guarantee a specific dialect, we assumed that a substantial part of tweets would be in the

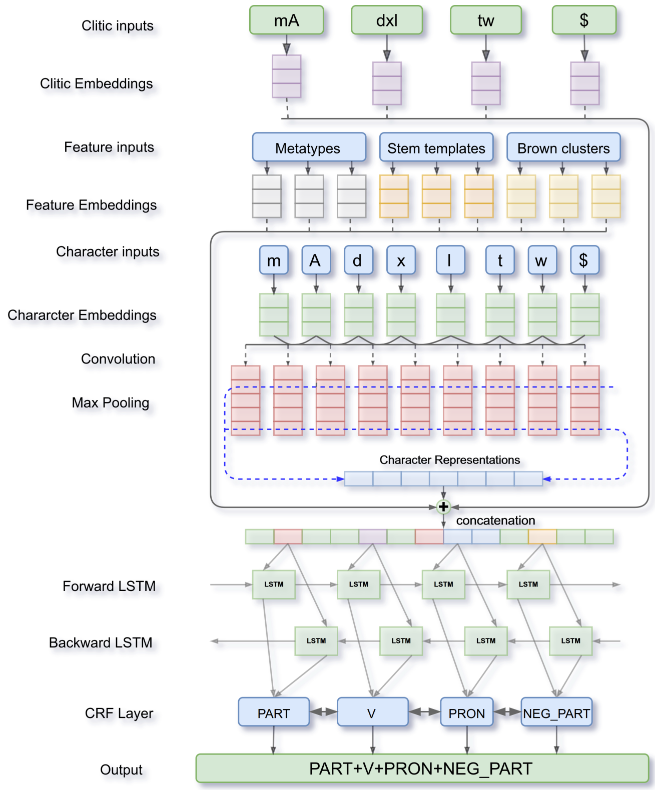

Fig. 2. DNN architecture applied on the word *mdxltw$* “you did not enter”.

desired dialect. For MSA, we filtered tweets using the aforementioned strong MSA words (pronouns and particles). We segmented MSA tweets using Farasa (Abdelali *et al.*, 2016) and the dialectal tweets using the DNN segmenter of Samih *et al.* (2017). We obtained BC’s using the implementation of Liang (2005). To illustrate the effectiveness of BCs, the top tokens in the cluster that includes the emoticon “;)” include “:d”, “xd”, and “:-d”. We used this dataset to pre-train embeddings.

5 Experiments and Results

5.1 CRF

We conducted two different sets of experiments. In the first set, we trained uni-dialectal models that are trained and tested on the same dialect. For our baseline ***(BL)*** model, given a sequence of clitics $c_n ... c_{-2}, c_{-1}, c_0, c_1, c_2 ... c_m$, where we assumed perfect segmentation, we used a combination of clitic unigram features $\{c_{-1};c_0;c_1\}$ and bigram features $\{c_{-2}^{-1};c_{-1}^{0};c_0^{1};c_1^{2};\}$ (Darwish *et al.*, 2018). We also experimented with stem templates ***(ST)***, clitic metatypes ***(MT)***, and Brown clusters ***(BC)***. For BCs, we varied the number of clusters to be 50, 100, 200, or 400, and we used cluster paths as features. We conducted side experiments where we used

	Unseen			Seen		
	+ST +MT	+BC	+ST +MT +BC	+ST +MT	+BC	+ST +MT +BC
MSA	83.5	89.4	91.5	94.2	95.8	96.4
EGY	79.7	84.7	88.6	94.9	96.2	96.3
LEV	74.5	79.4	79.4	90.8	91.8	92.1
GLF	71.7	82.5	83.1	91.2	93.3	93.7
MGR	74.7	80.9	82.9	91.3	91.8	92.4
AVG	76.8	83.4	85.1	92.5	93.8	94.2

Table 3. *Effect of ST+MT compared to BC_{200}*

different prefix lengths of cluster paths, but we did not observe any improvements. Table 4 shows the effect of using each of the different features individually or collectively. All the results reported in the paper are at word-level (not clitic-level). Since words are composed of one or more clitics, per clitic results are almost always higher than word-level results. To illustrate the difference, consider the utterance: *m+Akl+t+$ Al+>kl* (I did not eat the food) where the correct POS tags would be: PART+V+PRON+NEG DET+NOUN. If the system erroneously generated PART+V+PRON+NEG DET+V, accuracy at word level would be 50% (first word-level composite tag is correct, while the second is not) while clitic-level accuracy would be 83.3% (5 out of 6).

As the results show, stem templates, metatypes, and Brown clusters improve upon the baseline model. Combining all features improves POS tagging accuracy with ST+MT+BC_{200} providing the best results (92.0 versus a baseline of 85.4). Though we tried to faithfully reproduce the results of Darwish *et al.* (2018), which constitute our baseline, our results were consistently higher by roughly 2%.

We also compared the effect of ST+MT together with BCs to using BCs alone on clitics that were seen or unseen during training. As Table 3 shows, BCs were far more effective than the combination of MT and ST in generalizing to unseen clitics. Using all features, yielded the best overall results. The effect was less pronounced for seen clitics.

In the second set, we trained joint models using the training data for all dialects and we tested on individual dialects. We wanted to determine if dialectal POS tagging for one dialect can benefit from the data of another dialect. We used all the aforementioned features and we varied the number of BCs. Table 5 (a) reports the results of joint training. As shown, joint training yields lower results than uni-dialectal models.

Dialect ID The lower results of the joint model prompted us to experiment with dialect ID as a feature. The rationale for this is that uni-dialectal models fundamentally assume that dialect IDs are known. We obtained dialect IDs using two methods. First, we used the gold dialect IDs. Second, we trained an automatic dialect classifier using the training and dev parts of each of the folds and we tested

	BL	+ST	+MT	+BC$_n$				+ST+MT+BC$_n$			
				n=50	100	200	400	n=50	100	200	400
MSA	85.4	88.3	88.7	91.2	92.0	92.4	92.7	93.9	94.2	**94.7**	94.3
EGY	90.9	92.4	92.0	92.4	93.1	93.9	93.7	94.1	94.3	**94.7**	**94.7**
LEV	84.3	86.3	87.3	86.3	86.7	87.2	88.0	89.3	89.4	**89.4**	**89.4**
GLF	83.9	86.6	85.3	87.8	88.6	89.7	91.0	90.6	90.7	**91.3**	89.6
MGR	82.6	84.6	86.8	85.9	86.6	87.3	88.4	88.9	89.1	**89.7**	89.4
Average	85.4	87.6	88.0	88.7	89.4	90.1	90.8	91.4	91.6	**92.0**	91.5

Table 4. *Training and testing on the same dialect with different features with n brown clusters.*

	no dialect ID (a)				Automatic dialect ID (b)				Gold dialect ID (c)			
# of BCs	50	100	200	400	50	100	200	400	50	100	200	400
MSA	93.3	93.3	**93.6**	93.2	94.2	94.2	**94.5**	94.4	94.5	94.4	**94.8**	94.6
EGY	94.0	94.5	**94.7**	94.4	94.9	95.4	**95.4**	95.2	95.2	95.7	**95.8**	95.5
LEV	89.6	89.3	**89.8**	89.4	90.1	90.2	**90.6**	90.4	90.7	90.8	90.7	**91.0**
GLF	89.5	89.8	**90.3**	89.8	90.6	91.1	91.3	**91.4**	91.3	91.8	**92.2**	92.1
MGR	88.3	88.6	**88.7**	**88.7**	89.9	90.2	90.2	**90.5**	90.2	90.5	90.7	**90.8**
Average	90.9	91.1	**91.4**	91.1	92.0	92.2	**92.4**	**92.4**	92.4	92.6	**92.8**	**92.8**

Table 5. *Joint training on all dialects and testing on individual dialects with all features and dialect ID.*

on the test part of the fold. We trained the fastText DNN classifier (Bojanowski *et al.*, 2016) using character 2, 3, 4, and 5 grams as inputs, a 40 dimensional embedding vector for each input, a learning rate of 0.1, and 50 training epochs. The resultant classifier achieved an average accuracy of 86.3% across all folds. Arabic dialect detection is non-trivial due to high lexical overlap between dialects. Our dialect identifier is competitive with state-of-the-art classifiers, where reported results for MSA vs. Egyptian range between 88.5% (Elfardy and Diab, 2013) and 94.4% (Darwish *et al.*, 2014) with even lower results (74%) for multi-dialect identification (Malmasi *et al.*, 2015).

We added the dialect ID as three features: dialect ID, combination of dialect ID and clitic, and combination of dialect ID and Brown clusters. Table 5 reports on the results of using either gold dialect IDs, which provides the maximum attainable gain from using dialect IDs, and automatic dialect ID. As can be seen, using dialect ID, either gold or automatic, as a feature yielded results that surpassed all our previous results. Using gold dialect IDs gave 0.4% higher than using the automatic dialect IDs. As seen before, using either 200 or 400 Brown clusters yielded the best overall results.

Significance Test We ran a paired two-tailed t-test and a Wilcoxon signed-rank test to ascertain if differences between results are statistically significant or not. While the t-test is a parametric test, the Wilcoxon test is not. When comparing two setups, we compared the results they produce for every fold for every dialect (25 different values) in a paired manner. We ran n x n comparisons between: baseline; baseline with each individual feature (ST, MT, BC); baseline with all features; and joint system with all features with no dialect ID, automatic dialect ID, and gold dialect ID. When BCs were used, we looked at systems using 200 BCs.

Aside from the comparison between the baseline with +ST and the baseline +MT where the p-values were 0.745 and 0.711 for the t-test and Wilcoxon test respectively, all other differences were statistically significant with p-values ≤ 0.01 for both tests. This shows that the BCs feature yields better results than either of the two other features. Combining features leads to statistically significant improvements. Joint training without dialect IDs degrades results significantly, while joint training with automatic or gold dialect IDs improves results significantly. Lastly, improving automatic dialect identification is likely to lead to statistically significant improvements.

5.2 DNN

We conducted two sets of experiments using DNN. First, we trained uni-dialectal models. We conducted four experiments with different layers stacked on top of each other, making use of linguistic features, word embeddings, and unsupervised clustering. The experiments were as follows:

Baseline (BL) were we used clitics only with randomly-initialized embeddings. In this setup we use bi-LSTM with Chain CRF classifier.

Baseline+Chars we added randomly-initialized embeddings and character representations. We add a one-dimensional CNN layer for characters. These two layers and the subsequent layers are merged (concatenated) together before being passed on to the Bi-LSTM and the CRF classifier.

BL+Chars+Embed we used pre-trained embeddings for clitics and characters, which were trained on the aforementioned tweets corpus.

BL+Chars+Embed+Features we used clitics and characters with pre-trained embeddings and all features (ST+MT+BC_{200}).

The results in Table 6 show that the DNN model receives significant boosts from adding: a) a CNN characters layer (79.3% to 87.3%); b) a pre-trained embeddings layer (87.3% to 90.7%); and c) the features (90.5% to 91.7%). We conducted side experiments to ascertain the relative effectiveness of the different features. We found that the model achieves the most gain from the Brown clusters (+0.9%) and does not get any significant gain from either metatypes (-0.2%) or stem templates (+0.1%). We assume that the reason for that is that character-specific features are already encoded in the CNN characters layer. One interesting observation is that though pre-trained embedding and Brown clusters are similar in the sense that both try to learn from unlabeled data, their combination is better than either of them alone.

In the second set of experiments, we trained joint models and tested on individual dialects. We used the best uni-dialectal configuration, which uses clitic and character inputs with pre-trained embeddings that represents all the dialects and all features. The results in Table 7 (a) show that the model does not benefit from

	BL	+Chars	+Chars +Embed	+Chars +Embed +ST+MT +BC$_{200}$
MSA	82.8	90.6	92.8	94.3
EGY	84.6	90.2	94.2	94.9
LEV	76.6	84.2	88.2	89.8
GLF	76.6	85.8	89.7	90.9
MGR	75.8	85.5	88.4	88.7
Average	79.3	87.3	90.7	91.7

Table 6. *DNN: Training and testing on the same dialect with different features*

Dialect ID	(a) None	(b) Gold
MSA	94.7	94.0
EGY	94.3	94.7
LEV	89.8	90.1
GLF	89.9	90.4
MGR	88.2	88.4
Average	91.4	91.5

Table 7. *Joint DNN training w/ & w/o dialect IDs (Features: Chars+Embed+ST+MT+BC$_{200}$)*

joint training with results dropping by 0.3%. This is consistent with results that we observed for CRFs (Table 5 (a)). Thus, we also experimented with providing the dialect ID as a feature to our DNN. Table 7 (b) shows the results of using gold dialect IDs. The DNN model does not benefit significantly from dialect IDs and the results of the joint model (with and without dialect ID) is comparable to the best uni-dialectal model.

For significance testing, again we used the t-test and the Wilcoxon test to compare all the setups in Tables 6 and 7. The best uni-dialectal setup (Chars+Embed+ST+MT+BC$_{200}$) and the joint training setups with and without dialect ID were all statistically indistinguishable with p-values ≥ 0.05 using both tests. All other differences were statistically significant with p-values ≤ 0.01 for both tests. This indicates that: a) adding a character-level CNN and pre-trained embeddings yielded statistically significant improvements; b) adding features led to significant improvement; c) the drop of the joint model, with and without dialect IDs, compared to the best uni-dialectal model was not statistically significant; and d) adding the dialect ID did not lead to significant improvement.

No. of possible	Dialect ID	
tags per clitic	None	Auto.
1	98.4	98.4
2	86.4	89.0
3	92.2	94.4
4	91.4	92.0

Table 8. *Effect of dialect ID on improving tagging of clitics with different number of possible tags*

5.3 Discussion

We compared the results of using the CRF and DNN setups. We specifically compared:

- CRF baseline (85.4% – Table 4) and DNN with character representations and pre-trained embeddings (90.7% – Table 6). Both uni-dialectal models involve no feature engineering. Results show that the DNN setup outperforms the CRF baseline, as the DNN is able to learn features automatically. The difference is statistically significant with p-values ≤ 0.01 for both the significance tests.

- CRF using all features (+ST+MT+BC$_{200}$) (92.0% – Table 4) and DNN using all features (+Chars+Embed+ST+MT+BC$_{200}$) (91.7% – Table 6). Though the CRF results are slightly higher (+0.3%), the difference is not significant with p-values ≥ 0.05 using both significance tests.

- joint learning with automatic dialect ID with CRF (92.4% – Table 5 (b)) and DNN (91.5% – Table 7 (b)). CRF yields statistically significantly better results than DNN for joint training with +0.9% improvement (absolute).

- We were hoping that by merely performing joint training, the results would improve overall. That was not the case. As seen in Figure 1, joining the tweets from different dialects increases ambiguity with 48% of clitics having at least 2 different possible POS tags compared to 30-36% for individual dialects. However, as Table 8 shows, adding dialect ID for the CRF setup improved tagging for clitics with multiple possible POS tags.

- Lastly, we looked at the effectiveness of our best CRF and DNN systems in handling words that were unseen during training. The CRF system was correct 85.4% of the time compared 86.0% for the DNN system. The difference is rather small with no clear advantage for either one.

Tables 9 and 10 show the most common word- and clitic-level errors for both

CRF and DNN approaches with joint training and how often they appear. The error distributions of both approaches seem to closely match, with verbs-noun and noun-adjective confusion being the most common. As the examples in Tables 9 and 10 show, many errors stem from: a) dialectal or foreign words (marked with ϕ in tables); b) ambiguous words that can assume either POS tag given different contexts (marked with ξ); c) letter substitutions between different forms of *alef*, *ta marbouta* and *ha*, and *alef maqsoura* and *ya* (marked with ψ); and d) non-conventional dialectal spellings (marked with λ). Other errors include named entities and misspelled words.

Error Type	CRF	DNN	Examples
ADJ+NSUFF ↔NOUN+NSUFF	9%	9%	شخصي+ة ($xSy+p) “personality/personal” ξ
V+PRON ↔NOUN+PRON	9	9	حكي+تي (Hky+ty) “my story/you said” ξ,λ
NOUN+NSUFF ↔NOUN+PRON	7	9	صبي+ه (Sby+h) “girl/his son” ξ,ψ
DET+ADJ ↔DET+NOUN	7	7	ال+سياسي (Al+syAsy) “the politic(al—ian)” ξ
DET+ADJ+NSUFF ↔DET+NOUN+NSUFF	6	6	ال+مصري+ين (Al+mSry+yn) “the Egyptians” ξ

Table 9. *Most common word-level errors*

Error Type	CRF	DNN	Examples
V↔NOUN	23%	26%	نوض (nwD) “get up” ϕ
ADJ↔NOUN	21	21	جميل (jmyl) “favor/beautiful” ξ
PART↔NOUN	8	7	هاي (hAy) “Hi” ϕ
ADJ↔V	6	5	أسعد (>sEd) “happiest/brought joy” ξ
ADV↔PART	5	4	بس (bs) “only/enough” ξ

Table 10. *Most common clitic-level errors*

5.4 Comparison to Existing Tools

We compared our results against two state-of-the-art Arabic POS taggers, namely Farasa (Darwish *et al.*, 2017) and MADAMIRA (Pasha *et al.*, 2014). Since neither were tuned for tweets, we edited their outputs to assign tweet-specific tags as not to penalize them for missing hashtags, mentions, URLs, and emoticons. Farasa is tuned for MSA only. Its accuracy on MSA tweets was 89.3%, which is much lower than its results on news (96.2%) (Darwish *et al.*, 2017) and also significantly lower than the results obtained by our CRF with joint training (94.5%). This suggests that the tweets genre is rather different from news stories, which behooves the need for training taggers specifically for tweets. As for MADAMIRA, which is tuned for MSA and EGY, achieved 88.0% and 90.5% on MSA and EGY tweets, respectively. Again, these results are much lower than MADAMIRA on news stories (95.3%) (Darwish *et al.*, 2017) and much lower than our CRF (94.5% and 95.4%). Training on limited in-domain data can yield better results than a POS tagger that is trained on lots of out-of-domain data. It is noteworthy that both Farasa and MADAMIRA could have benefited from gold segmentations. However, altering either system to supply gold segmentations is beyond the scope of this work.

6 Conclusion

In this paper, we present state-of-the-art POS tagging of multi-dialectal Arabic tweets using CRF and DNN approaches, and based on a small tagged dataset of Arabic tweets. The dataset covers MSA and the four major dialect groups. We explore uni-dialectal and joint models. While uni-dialectal CRF and DNN models yield statistically indistinguishable results, our CRF model outperforms DNN for joint training. We show that using a mere 350 tweets per dialect can lead to word-level accuracy of 92.4% on average across dialects. The CRF model obtains the best results when using: linguistic features (stem templates and clitic metatypes), word clusters from a large unlabeled tweet corpus, and automatic dialect identification. These three elements boost the accuracy from 85.4% to 92.4% (i.e., a relative error reduction of 48%). For the DNN approach, we combine clitic and character-level inputs with the features we used for CRF to obtain the best DNN results. Though both serve similar purposes, combining pre-trained embeddings and word clusters yields the best results. Linguistic features and dialect identification marginally affect DNN results.

References

Abdelali Ahmed, Darwish Kareem, Durrani Nadir, and Mubarak Hamdy. 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. Association for Computational Linguistics, San Diego, California.

Al-Sabbagh Rania and Girju Roxana. 2010. Mining the web for the induction of a dialectical arabic lexicon. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC 2010)*, pages 288–293.

Bojanowski Piotr, Grave Edouard, Joulin Armand, and Mikolov Tomas. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606.*

Bouamor Houda, Habash Nizar, and Oflazer Kemal. 2014. A multidialectal parallel corpus of arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 1240–1245.

Brown Peter F, Desouza Peter V, Mercer Robert L ,Pietra Vincent J Della , and Lai Jenifer C . 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Caruana Rich, Lawrence Steve, and Giles Lee. 2000. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *NIPS*, pages 402–408.

Chiu Jason and Nichols Eric. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Collobert R., Weston J., Bottou L., Karlen M., Kavukcuoglu K., and Kuksa P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Cotterell Ryan and Callison-Burch Chris. 2014. A multi-dialect, multi-genre corpus of informal written arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 241–245.

Darwish Kareem, Mubarak Hamdy, Abdelali Ahmed, and Eldesouki Mohamed. 2017. Arabic pos tagging: Don't abandon feature engineering just yet. *WANLP 2017 (co-located with EACL 2017)*, page 130.

Darwish Kareem, Mubarak Hamdy, Abdelali Ahmed, Eldesouki Mohamed, Samih Younes, Alharbi Randah, Attia Mohammed, Magdy Walid, and Kallmeyer Laura. 2018. Multidialect arabic POS tagging: A CRF approach. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.*

Darwish Kareem, Sajjad Hassan, and Mubarak Hamdy. 2014. Verifiably effective arabic dialect identification. In *EMNLP*, pages 1465–1468.

Derczynski Leon, Ritter Alan, Clark Sam, and Bontcheva Kalina. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, pages 198–206.

Duh Kevin and Kirchhoff Katrin. 2005. Pos tagging of dialectal arabic: a minimally supervised approach. In *Proceedings of the acl workshop on computational approaches to semitic languages*, pages 55–62. Association for Computational Linguistics.

Elfardy Heba and Diab Mona T. 2013. Sentence level dialect identification in arabic. In *ACL (2)*, pages 456–461.

Gimpel Kevin, Schneider Nathan, O'Connor Brendan, Das Dipanjan, Mills Daniel, Eisenstein Jacob, Heilman Michael, Yogatama Dani, Flanigan Jeffrey, and Smith Noah A . 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

Graja Marwa, Jaoua Maher, and Hadrich Belguith Lamia . 2010. Lexical study of a spoken dialogue corpus in tunisian dialect. In *The international arab conference on information technology (ACIT), benghazi–libya.*

Habash Nizar, Diab Mona T, and Rambow Owen. 2012. Conventional orthography for dialectal arabic. In *LREC*, pages 711–718.

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

Hinton Geoffrey E, Srivastava Nitish, Krizhevsky Alex, Sutskever Ilya, and Salakhutdinov Ruslan R . 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580.*

Hochreiter Sepp and Schmidhuber Jürgen. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Huang Zhiheng, Xu Wei, and Yu Kai. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Jurafsky Daniel and Martin James H. 2009. *Speech and Language Processing*, 2nd edition. Pearson Prentice Hall, New Jersey. ISBN 978-0-13-187321-6.

Khalifa Salam, Hassan Sara, and Habash Nizar. 2017. A morphological analyzer for gulf arabic verbs. *WANLP 2017 (co-located with EACL 2017)*, page 35.

Lafferty John, McCallum Andrew, and Pereira Fernando CN. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pages 282–289.

Lample Guillaume, Ballesteros Miguel, Subramanian Sandeep, Kawakami Kazuya, and Dyer Chris. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360.*

Liang Percy. 2005. *Semi-supervised learning for natural language.* Ph.D. thesis, Massachusetts Institute of Technology.

Ma Xuezhe and Hovy Eduard. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Malmasi Shervin, Refaee Eshrag, and Dras Mark. 2015. Arabic dialect identification using a parallel multidialectal corpus. In *International Conference of the Pacific Association for Computational Linguistics*, pages 35–53. Springer.

Owoputi Olutobi, O'Connor Brendan, Dyer Chris, Gimpel Kevin, Schneider Nathan, and Smith Noah A. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT 2013*, pages 380–390. Association for Computational Linguistics.

Pasha Arfath, Al-Badrashiny Mohamed, Diab Mona T, El Kholy Ahmed, Eskander Ramy, Habash Nizar, Pooleery Manoj, Rambow Owen, and Roth Ryan. 2014. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 1094–1101.

Reimers Nils and Gurevych Iryna. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 338–348, Copenhagen, Denmark.

Ryan Roth, Rambow Owen, Habash Nizar, Diab Mona, and Rudin Cynthia. 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of the Conference of American Association for Computational Linguistics (ACL08).*

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

dos Santos Cicero and Guimarães Victor . 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entity Workshop*, pages 25–33, Beijing, China. Association for Computational Linguistics.

Schuster Mike and Paliwal Kuldip K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Stratos Karl and Collins Michael. 2015. Simple semi-supervised pos tagging. In *VS@ HLT-NAACL*, pages 79–87.

Zaidan Omar F and Callison-Burch Chris. 2011. The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In *Proceedings of* *the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 37–41. Association for Computational Linguistics.