A Neural Architecture for Dialectal Arabic Segmentation

Younes Samih[1], **Mohammed Attia**[2], **Mohamed Eldesouki**[3], **Hamdy Mubarak**[3], **Ahmed Abdelali**[3], **Laura Kallmeyer**[1] **and Kareem Darwish**[3]

[1]Dept. of Computational Linguistics,University of Düsseldorf, Düsseldorf, Germany
[2]Google Inc., New York City, USA
[3]Qatar Computing Research Institute, HBKU, Doha, Qatar
[1]{samih,kallmeyer}@phil.hhu.de
[2]attia@google.com
[3]{mohamohamed,hmubarak,aabdelali,kdarwish}@hbku.edu.qa

Abstract

The automated processing of Arabic dialects is challenging due to the lack of spelling standards and the scarcity of annotated data and resources in general. Segmentation of words into their constituent tokens is an important processing step for natural language processing. In this paper, we show how a segmenter can be trained on only 350 annotated tweets using neural networks without any normalization or reliance on lexical features or linguistic resources. We deal with segmentation as a sequence labeling problem at the character level. We show experimentally that our model can rival state-of-the-art methods that heavily depend on additional resources.

1 Introduction

The Arabic language has various dialects and variants that exist in a continuous spectrum. This variation is a result of multiple morpho-syntactic processes of simplification and mutation, as well as coinage and borrowing of new words in addition to semantic shifts of standard lexical items. Furthermore, there was a considerable effect of the interweave between the standard Arabic language that spread throughout the Middle East and North Africa and the indigenous languages in different countries as well as neighboring languages. With the passage of time and the juxtaposition of cultures, dialects and variants of Arabic evolved and diverged. Among the varieties of Arabic is so-called Modern Standard Arabic (MSA) which is the lingua franca of the Arab world, and is typically used in written and formal communications. On the other hand, Arabic dialects, such as Egyptian, Moroccan and Levantine, are usually spoken and used in informal communications.

The advent of the social networks and the spread of smart phones, yielded the need for dialect-aware smart systems and motivated the research in Dialectal Arabic such as dialectal Arabic identification for both text (Eldesouki et al., 2016) and speech (Khurana et al., 2016), morphological analysis (Habash et al., 2013) and machine translation (Sennrich et al., 2016; Sajjad et al., 2013).

Due to the rich morphology in Arabic and its dialects, *word segmentation* is one of the most important processing steps. Word segmentation is considered an integral part for many higher Arabic NLP tasks such as part-of-speech tagging, parsing and machine translation. For example, the Egyptian word ومكتبهاش "wmktbhA$" meaning: "and he didn't write it") includes four clitics surrounding the the verb (stem) "ktb", and is rendered after segmentation as "w+m+ktb+hA+$". The clitics in this word are the coordinate conjunction "w", the negation prefix "m", the object pronoun "hA", and the post negative suffix "$".

In this paper, we present a dialectal Egyptian segmentater that utilizes Bidirectional Long-Short-Term-Memory (BiLSTM) that is trained on limited dialectal data. The approach was motivated by the scarcity of dialectal tools and resources. The main contribution of this paper is that we build a segmenter of dialectal Egyptian using limited data without the need for specialized lexi-

Proceedings of The Third Arabic Natural Language Processing Workshop (WANLP), pages 46–54,
Valencia, Spain, April 3, 2017. ©2017 Association for Computational Linguistics

cal resources or deep linguistic knowledge that rivals state-of-the-art tools.

Challenges of Dialectal Arabic

Dialectal Arabic (DA) shares many challenges with MSA, as DA inherits the same nature of being a Semitic language with complex templatic derivational morphology. As in MSA, most of the nouns and verbs in Arabic dialects are typically derived from a determined set of roots by applying templates to the roots to generate stems. Such templates may carry information that indicate morphological features of words such POS tag, gender, and number. Further, stems may accept prefixes and/or suffixes to form words which turn DA into highly inflected language. Prefixes include coordinating conjunctions, determiner, particles, and prepositions, and suffixes include attached pronouns and gender and number markers. This results in a large number of words (or surface forms) and in turn a high-level of sparseness and increased number of unseen words during testing.

In addition to the shared challenges, DA has its own peculiarities, which can be summarized as follows:

- Lack of standard orthography. Many of the words in DA do not follow a standard orthographic system (Habash et al., 2012).
- Many words do not overlap with MSA as result of language borrowing from other languages (Ibrahim, 2006), such as كافيه kAfiyh "cafe" and تاتو tAtuw "tattoo", or coinage, such as the negative particles مش mi$ "not" and بلاش balA$ "do not". Code switching is also very common in Arabic dialects (Samih et al., 2016).
- Merging multiple words together by concatenating and dropping letters such as the word مبيجلهاش mbyjlhA$ (he did not go to her), which is a concatenation of "mA byjy lhA$".
- Some affixes are altered in form from their MSA counterparts, such as the feminine second person pronoun ك k → كي ky and the second person plural pronoun تم tm → تو tw.
- Some morphological patterns that do not exist in MSA, such as the passive pattern Aitofaeal, such as اتكسر Aitokasar "it broke".
- Introduction of new particles, such is the progressive ب b meaning 'is doing' and the post negative suffix ش $, which behaves like the French "ne-pas" negation construct.
- Letter substitution and consonant mutation. For example, in dialectal Egyptian, the interdental sound of the letter ث v is often substituted by either ت t or س s as in كثير kvyr "much" → كتير ktyr and the glottal stop is reduced to a glide, such as جائز jA}iz "possible" → جايز jAyiz. Such features is deeply studied in phonology under lenition, softening of a consonant, or fortition, hardening of a consonant.
- Vowel elongation, such as راجل rAjil "man" from رجل rajul, and vowel shortening, such as دايما dayomA "always" from دايما dAyomA.
- The use of masculine plural or singular noun forms instead dual and feminine plural, dropping some articles and preposition in some syntactic constructs, and using only one form of noun and verb suffixes such as ين yn instead of ون wn and وا wA instead of ون wn respectively.
- In addition, there are the regular discourse features in informal texts, such as the use of emoticons and character repetition for emphasis, e.g. ادعووووولى AdEwwwwwwwliy "pray for me".

2 Related Work

Work on dialectal Arabic is fairly new compared to MSA. A number of research projects were devoted to dialect identification (Biadsy et al., 2009; Zbib et al., 2012; Zaidan and Callison-Burch, 2014). There are five major dialects including Egyptian, Gulf, Iraqi, Levantine and Maghribi. Few resources for these dialects are available such as the CALLHOME Egyptian Arabic Transcripts (LDC97T19), which was made available for research as early as 1997. Newly developed resources include the corpus developed by Bouamor et al. (2014), which contains 2,000 parallel sentences in multiple dialects and MSA as well as English translation.

For segmentation, Yao and Huang (2016) successfully used a bi-directional LSTM model for segmenting Chinese text. In this paper, we build on their work and extend it in two ways, namely combining bi-LSTM with CRF and applying on Arabic, which is an alphabetic language. Mohamed et al. (2012) built a segmenter based on memory-based learning. The segmenter has been trained on a small corpus of Egyptian Arabic comprising 320 comments containing 20,022 words from `www.masrawy.com` that were segmented and annotated by two native speakers. They reported a 91.90% accuracy on the task of segmentation. MADA-ARZ (Habash et al., 2013) is an Egyptian Arabic extension of the Morphological Analysis and Disambiguation of Arabic (MADA). They trained and evaluated their system on both Penn Arabic Treebank (PATB) (parts 1-3) and the Egyptian Arabic Treebank (parts 1-5) (Maamouri et al., 2014) and they reported 97.5% accuracy. MARAMIRA[1] (Pasha et al., 2014) is a new version of MADA and includes as well the functionality of MADA-ARZ which will be used in this paper for comparison. Monroe et al. (2014) used a single dialect-independent model for segmenting all Arabic dialects including MSA. They argue that their segmenter is better than other segmenters that use sophisticated linguistic analysis. They evaluated their model on three corpora, namely parts 1-3 of the Penn Arabic Treebank (PATB), Broadcast News Arabic Treebank (BN), and parts 1-8 of the BOLT Phase 1 Egyptian Arabic Treebank (ARZ) reporting an F1 score of 95.13%.

3 Arabic Segmentation Model

In this section, we will provide a brief description of LSTM, and introduce the different components of our Arabic segmentation model. For all our work, we used the Keras toolkit (Chollet, 2015). The architecture of our model, shown in Figure 2 is similar to Ma and Hovy (2016), Huang et al. (2015), and Collobert et al. (2011)

3.1 Long Short-term Memory

A recurrent neural network (RNN) belongs to a family of neural networks suited for modeling sequential data. Given an input sequence $x = (x_1, ..., x_n)$, an RNN computes the output vector y_t of each word x_t by iterating the following equations from $t = 1$ to n:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t_{-1}} + b_h)$$
$$y_t = W_{hy}h_t + b_y$$

where h_t is the hidden states vector, W denotes weight matrix, b denotes bias vector and f is the activation function of the hidden layer. Theoretically RNN can learn long distance dependencies, still in practice they fail due the vanishing/exploding gradient (Bengio et al., 1994). To solve this problem, Hochreiter and Schmidhuber (1997) introduced the long short-term memory RNN (LSTM). The idea consists in augmenting the RNN with memory cells to overcome difficulties with training and efficiently cope with long distance dependencies. The output of the LSTM hidden layer h_t given input x_t is computed via the following intermediate calculations: (Graves, 2013):

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i)$$
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f)$$
$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$
$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o)$$
$$h_t = o_t \tanh(c_t)$$

where σ is the logistic sigmoid function, and i, f, o and c are respectively the input gate, forget gate, output gate and cell activation vectors. More interpretation about this architecture can be found in (Lipton et al., 2015). Figure 1 illustrates a single LSTM memory cell (Graves and Schmidhuber, 2005)

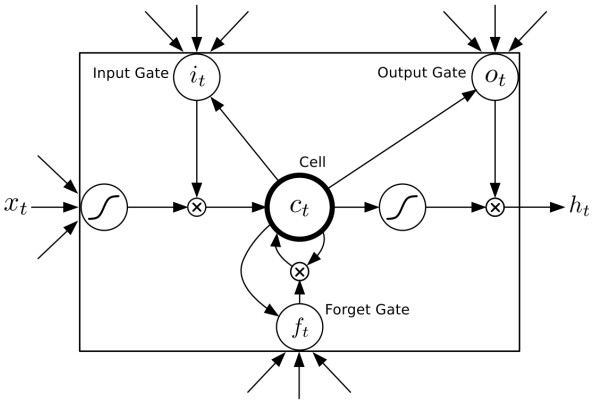

Figure 1: A Long Short-Term Memory Cell.

3.2 Bi-directional LSTM

Bi-LSTM networks (Schuster and Paliwal, 1997) are extensions to the single LSTM networks. They

[1]MADAMIRA release 20160516 2.1

are capable of learning long-term dependencies and maintain contextual features from the past states and future states. As shown in Figure 2, they comprise two separate hidden layers that feed forwards to the same output layer. A BiLSTM calculates the forward hidden sequence $\overrightarrow{h}$, the backward hidden sequence $\overleftarrow{h}$ and the output sequence y by iterating over the following equations:

$$\overrightarrow{h_t} = \sigma(W_{x\overrightarrow{h}}x_t + W_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{h}_{t-1} + b_{\overrightarrow{h}})$$
$$\overleftarrow{h_t} = \sigma(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t-1} + b_{\overleftarrow{h}})$$
$$y_t = W_{\overrightarrow{hy}}\overrightarrow{h_t} + W_{\overleftarrow{hy}}\overleftarrow{h_t} + b_y$$

More interpretations about these formulas are found in Graves et al. (2013a).

3.3 Conditional Random Fields (CRF)

Over the last recent years, BiLSTMs have achieved many ground-breaking results in many NLP tasks because of their ability to cope with long distance dependencies and exploit contextual features from the past and future states. Still when they are used for some specific sequence classification tasks, (such as segmentation and named entity detection), where there is a strict dependence between the output labels, they fail to generalize perfectly. During the training phase of the BiLSTM networks, the resulting probability distribution of each time step is independent from each other. To overcome the independence assumptions imposed by the BiLSTM and exploit these kind of labeling constraints in our Arabic segmentation system, we model label sequence logic jointly using Conditional Random Fields (CRF) (Lafferty et al., 2001). CRF, a sequence labeling algorithm, predicts labels for a whole sequence rather than for the parts in isolation as shown in Equation 1. Here, s_1 to s_m represent the labels of tokens x_1 to x_m respectively, where m is the number of tokens in a given sequence. After we have this probability value for every possible combination of labels, the actual sequence of labels for this set of tokens will be the one with the highest probability.

$$p(s_1...s_m|x_1...x_m) \quad (1)$$

$$p(\vec{s}|\vec{x};\vec{w}) = \frac{exp(\vec{w}.\vec{\Phi}(\vec{x},\vec{s}))}{\sum_{\vec{s'}\epsilon S^m} exp(\vec{w}.\vec{\Phi}(\vec{x},\vec{s'}))} \quad (2)$$

Equation 2 shows the formula for calculating the probability value from Equation 1. Here, S is the set of labels. In our case S *={B, M, E, S, WB}*, where B is the beginning of a token, M is the middle of a token, E is the end of a token, S is a single character token, and WB is the word boundary. $\vec{w}$ is the weight vector for weighting the feature vector $\vec{\Phi}$. Training and decoding are performed by the Viterbi algorithm.

Note that replacing the softmax with CRF at the output layer in neural networks has proved to be very fruitful in many sequence labeling tasks (Ma and Hovy, 2016; Huang et al., 2015; Lample et al., 2016; Samih et al., 2016)

3.4 Pre-trained characters embeddings

A very important element of the recent success of many NLP applications, is the use of character-level representations in deep neural networks. This has shown to be effective for numerous NLP tasks (Collobert et al., 2011; dos Santos et al., 2015) as it can capture word morphology and reduce out-of-vocabulary. This approach has also been especially useful for handling languages with rich morphology and large character sets (Kim et al., 2016). We use pre-trained character embeddings to initialize our look-up table. Characters with no pre-trained embeddings are randomly initialized with uniformly sampled embeddings. To use these embeddings in our model, we simply replace the one hot encoding character representation with its corresponding 200-dimensional vector. Table 1 shows the statistics of data we used to train our character embeddings.

Genre	**Tokens**
Facebook posts	8,241,244
Tweets	2,813,016
News comments	95,241,480
MSA news texts	276,965,735
total	**383,261,475**

Table 1: character embeddings training data statistics

3.5 BiLSTM-CRF for Arabic Segmentation

In our model we consider Arabic segmentation as character-based sequence classification problem. Each character is labeled as one of the five labels B, M, E, S, WB that designate the segmentation decision boundary. B, M, E, WB represent Beginning, Middle, End of a multi-character segment, Single character segment, and Word Bound-

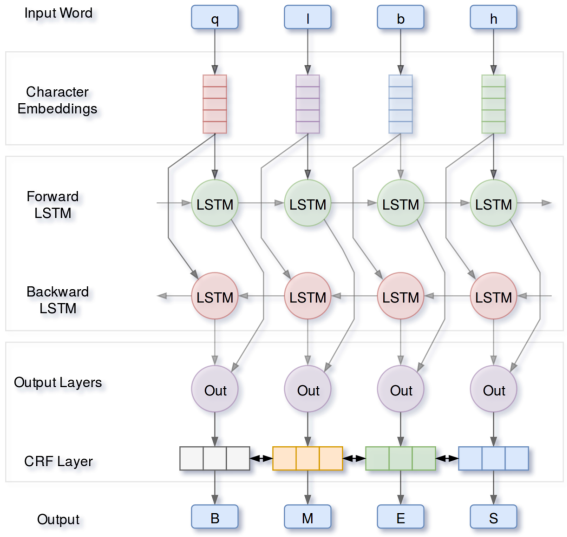

Figure 2: Architecture of our proposed neural network Arabic segmentation model applied to an example word. Here the model takes the word *qlbh*, "his heart" as its current input and predicts its correct segmentation. The first layer performs a look up of the characters embedding and stacks them to build a matrix. This latter is then used as the input to the Bi-directional LSTM. On the last layer, an affine transformation function followed by a CRF computes the probability distribution over all labels

ary respectively.

The architecture of our segmentation model, shown in Figure 2, is straightforward. It comprises the following three layers:

- Input layer: it contains character embeddings.
- Hidden layer: BiLSTM maps character representations to hidden sequences.
- Output layer: CRF computes the probability distribution over all labels.

At the input layer a look-up table is initialized by pre-trained embeddings mapping each character in the input to d-dimensional vector. At the hidden layer, the output from the character embeddings is used as the input to the BiLSTM layer to obtain fixed-dimensional representations for each character. At the output layer, a CRF is applied over the hidden representation of the BiLSTM to obtain the probability distribution over all the labels. Training is performed using stochastic gradient (SGD) descent with momentum 0.9 and batch size 50, optimizing the cross entropy objective function.

3.6 Regularization

Dropout Due to the relatively small size the training data set and development data set, overfitting poses a considerable challenge for our Dialectal Arabic segmentation system. To make sure that our model learns significant representations, we resort to dropout (Hinton et al., 2012) to mitigate overfitting. The basic idea of dropout consists in randomly omitting a certain percentage of the neurons in each hidden layer for each presentation of the samples during training. This encourages each neuron to depend less on other neurons to learn the right segmentation decision boundaries. We apply dropout masks to the character embedding layer before inputting to the BiLSTM and to its output vector. In our experiments we find that dropout with a rate fixed at 0.5 decreases overfitting and improves the overall performance of our system.

Early Stopping We also employ early stopping (Caruana et al., 2000; Graves et al., 2013b) to mitigate overfitting by monitoring the model's performance on development set.

4 Dataset

We used the dataset described in (Darwish et al., 2014). The data was used in a dialect identification task to distinguish between dialectal Egyptian and MSA. It contains 350 tweets with more than 8,000 words including 3,000 unique words written in Egyptian dialect. The tweets have much dialectal content covering most of dialectal Egyptian phonological, morphological, and syntactic phenomena. It also includes Twitter-specific aspects of the text, such as #hashtags, @mentions, emoticons and URLs.

We manually annotated each word in this corpus to provide: CODA-compliant writing (Habash et al., 2012), segmentation, stem, lemma, and POS, also the corresponding MSA word, MSA segmentation, and MSA POS. We make the dataset[2] available to researchers to reproduce the results and help in other tasks such as CODA'fication of dialectal text, dialectal POS tagging and dialect to MSA conversion. Table 2 shows an annotation ex-

[2]Dataset is available at `http://alt.qcri.org/resources/da_resources`

ample of the word بيقولك "byqwlk" (he is saying to you).

Field	Annotation
Orig. word	بيقولك "byqwlk"
CODA	بيقول لك "byqwl lk"
Segmentation	ب+يقول ل+ك "b+yqwl l+k"
POS	PROG_PART+V PREP+PRON
Stem	يقول ل "yqwl l"
lemma	قال ل "qAl l"
MSA	يقول لك "yqwl lk"
MSA Segm.	يقول ل+ك "yqwl l+k"
MSA POS	V PREP+PRON

Table 2: Annotation Example

For the purpose of this paper, we skip CODA'fication, and conduct segmentation on the original words to increase the robustness of the system. Therefore, the segmentation of the example in Table 2 is given as بـ+يقو+لـ+ك b+yqw+l+k. We need also to note that, by design, the perfective prefixes are not separated from verbs in the current work.

5 Experiments and Results

We split the data described in section 4 into 75 sentences for testing, 75 for development and the remaining 200 for training.

The concept We followed in LSTM sequence labeling is that segmentation is one-to-one mapping at the character level where each character is annotated as either beginning a segment (B), continues a previous segment (M), ends a segment (E), or is a segment by itself (S). After the labeling is complete we merge the characters and labels together, for example بيقولوا byqwlwA is labeled as "SBMMEBE", which means that the word is segmented as b+yqwl+wA. We compar results of our two LSTM models (BiLSTM and BiLSTM-CRF) with Farasa (Abdelali et al., 2016), an open source segementer for MSA[3], and MADAMIRA for Egyptian dialect. Table 3 shows accuracy for Farasa, MADAMIRA, and both of our models.

The results show that for this small test-set BiLSTM-CRF (92.65%) performs better than

[3]Available for download from: http://alt.qcri.org/tools/farasa/

System	Accuracy
Farasa (Baseline[4])	88.34 %
MADAMIRA	92.47 %
BiLSTM	86.27 %
BiLSTM-CRF	**92.65 %**

Table 3: F_1 and accuracy results on the test data. We consider Farasa our baseline. This table compares between Farasa, BiLSTM models with MADAMIRA

MADAMIRA (92.47%) by only 0.18% which is not statistically significant. The advantage of our system is that, unlike MADAMIRA which relies on a hand-crafted lexicon, our system generalizes well on unseen data. To illustrate this point, the test set has 1,449 words, and 586 of them (40%) are not seen in the training set. This shows how well the system is robust with OOV words.

6 Analysis

MADAMIRA error analysis:

When analyzing the errors (109 errors) in MADAMIRA, we found that they are most likely due to lexical coverage or the performance of morphological processing and variability.

- OOV words: e.g. الوايرلس AlwAyrls "the wireless", الهاشتاج AlHA$tAj "the hashtag".
- Spelling variation: e.g. الغطى AlgTY "the cover", لأهلى l>hly "to Ahly".
- Morphological inflection (imperative): e.g. شدي $dy "pull", فوقوا fwqwA "wake up".
- Segmentation ambiguity: e.g. ليه lyh meaning "why" or "to him", مالنا mAlnA meaning "our money" or "what we have".
- Combinations not known to MADAMIRA: e.g. متقفلوهاش mtqflwhA$ "don't close it", أوصفلكوا >wSflkwA "I describe to you".
- Different annotation convention: e.g. عشان E$An "because" and النهارده AlnhArdh "today" are one token in our gold data but analyzed as two tokens in MADAMIRA.

BiLSTIM Error analysis:

The errors in this system (199 errors) are broadly classified into three categories:

- Confusing prefixes and suffixes with stem's constituent letters: e.g. لطيفه lTyfh "nice", عالمي EAlmy "international".
- Not identifying segments: e.g. يارب yArb "O my Lord", قدامي qd Amy "in front of me".
- The majority of errors (108 instances) are bad sequences coming from invalid label combination, like having an E or M without a preceding B, or M without a following E. It seems that this label sequence logic is not yet fully absorbed by the system, maybe due to the small amount of training data.

BiLSTIM-CRF Error analysis:

This model successfully avoids the invalid sequence combinations found in BiLSTM. As pointed out by (Lample et al., 2016), BiLSTM makes independent classification decisions which does not work well when there are interdependence across labels (e.g., E or M must be preceded by B, and M must be followed by E). Segmentation is one such task, where independence assumption is wrong, and this is why CRF works better than the softmax in modeling tagging decisions jointly, correctly capturing the sequence logic.

The number of errors in BiLSTIM-CRF is reduced to 101 and the number of label sequences not found in the gold standard is reduced to just 14, yet with all of them obeying the valid sequence rules. The remaining errors are different from the errors generated by BiLSTM, but they are similar in that the mistokenization happens due to the system's inability to decide whether a substring (which out of context can be a valid token) is an independent token or part of a word, e.g. بخير bikhir "is well', ماشي mA$iy "OK".

7 Conclusion

Using BiLSTM-CRF, we show that we can build an effective segmenter using limited dialectal Egyptian Arabic labeled data without relying on lexicons, morphological analyzer or linguistic knowledge. The CRF optimizer for LSTM successfully captures label sequence logic and avoids invalid label combinations. The results obtained are comparable to a state-of-the-art system, namely MADAMIRA, or even better. Admittedly, the small test set used in this work might not allow us to generalize the claim, and we plan to run more expansive tests. Nonetheless, given that there are no standard dataset available for this task, objective comparison of different systems remains elusive. A number of improvements can possibly enhance the accuracy of our system further, including exploiting large resources available for MSA. Despite the differences dialects and MSA, there is significant lexical overlap between MSA and dialects. This is demonstrated by the accuracy of Farasa which was built to handle MSA exclusively, yet achieving 88.34% accuracy on the dialectal data. Thus, combining MSA and dialectal data in training or performing domain adaptation stands to enhance segmentation. Additionally, we plan to carry these achievements further to explore other dialects.

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. Association for Computational Linguistics, San Diego, California.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Fadi Biadsy, Julia Hirschberg, and Nizar Habash. 2009. Spoken arabic dialect identification using phonotactic modeling. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, Semitic '09, pages 53–61, Stroudsburg, PA, USA. Association for Computational Linguistics.

Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. A multidialectal parallel corpus of arabic. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Rich Caruana, Steve Lawrence, and Lee Giles. 2000. Overfitting in neural nets: Backpropagation, conju-

gate gradient, and early stopping. In *NIPS*, pages 402–408.

François Chollet. 2015. keras. https://github.com/fchollet/keras.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

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

Cıcero dos Santos, Victor Guimaraes, RJ Niterói, and Rio de Janeiro. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of NEWS 2015 The Fifth Named Entities Workshop*, page 25.

Mohamed Eldesouki, Fahim Dalvi, Hassan Sajjad, and Kareem Darwish. 2016. Qcri@ dsl 2016: Spoken arabic dialect identification using textual. *VarDial 3*, page 221.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610.

Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. 2013a. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013b. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

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

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.

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

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

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

Zeinab Ibrahim. 2006. Borrowing in modern standard arabic. *Innovation and Continuity in Language and Communication of Different Language Cultures 9. Edited by Rudolf Muhr*, pages 235–260.

Sameer Khurana, Ahmed Ali, and Steve Renals. 2016. Multi-view dimensionality reduction for dialect identification of arabic broadcast speech. *arXiv preprint arXiv:1609.05650*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzell. 2015. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019.

Xuezhe Ma and Eduard Hovy. 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, August. Association for Computational Linguistics.

Mohamed Maamouri, Ann Bies, Seth Kulick, Michael Ciul, Nizar Habash, and Ramy Eskander. 2014. Developing an egyptian arabic treebank: Impact of dialectal morphology on annotation and tool development. In *LREC*, pages 2348–2354.

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 *ACL (2)*, pages 206–211.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan M Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. *Proc. LREC*.

Hassan Sajjad, Kareem Darwish, and Yonatan Belinkov. 2013. Translating dialectal Arabic to English. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL '13, pages 1–6, Sofia, Bulgaria.

Younes Samih, Suraj Maharjan, Mohammed Attia, Laura Kallmeyer, and Thamar Solorio. 2016. Multilingual code-switching identification via lstm recurrent neural networks. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching,*, pages 50–59, Austin, TX.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Yushi Yao and Zheng Huang. 2016. Bi-directional lstm recurrent neural network for chinese word segmentation. In *International Conference on Neural Information Processing*, pages 345–353. Springer.

Omar F Zaidan and Chris Callison-Burch. 2014. Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. 2012. Machine translation of arabic dialects. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 49–59, Stroudsburg, PA, USA. Association for Computational Linguistics.