LSTM-CNN Deep Learning Model for Sentiment Analysis of Dialectal Arabic

Kathrein Abu Kwaik[1(✉)], Motaz Saad[2], Stergios Chatzikyriakidis[1], and Simon Dobnik[1]

[1] CLASP, Department of Philosophy, Linguistics and Theory of Science, Box 200, 405 30 Gothenburg, Sweden
{kathrein.abu.kwaik,stergios.chatzikyriakidis,simon.dobnik}@gu.se
[2] The Islamic University of Gaza, Gaza, Palestine
motaz.saad@gmail.com

Abstract. In this paper we investigate the use of Deep Learning (DL) methods for Dialectal Arabic Sentiment Analysis. We propose a DL model that combines long-short term memory (LSTM) with convolutional neural networks (CNN). The proposed model performs better than the two baselines. More specifically, the model achieves an accuracy between 81% and 93% for binary classification and 66% to 76% accuracy for three-way classification. The model is currently the state of the art in applying DL methods to Sentiment Analysis in dialectal Arabic.

Keywords: Sentiment Analysis · Arabic dialects · Deep Learning · LSTM · CNN

1 Introduction

With the emergence of social media, large amounts of valuable data become available online and easy to access. Social media users discuss everything they care about through blog posts or tweets, share their opinions and show interest freely; while they do not actually do it in person. We read about political debates, social problems, questions about a particular product, etc. Companies also use social networks to promote their products and services, and explore people's opinions to improve their products and services, thereby generating a huge amount of data. In this context, the need for an analytical tool that can process the users data and classify them in terms of sentiment polarities is increased and become a necessity.

Sentiment analysis (SA) or Opinion Mining (OM) is the task of determining and detecting the polarity/opinion in a given piece of text and classifying it into positive, negative or neutral and in some fine grained cases also a mixed class. English and other European languages have been explored in the majority SA tools and research; recent efforts extend the focus to other low-resources languages such as Arabic and dialectal Arabic.

© Springer Nature Switzerland AG 2019
K. Smaïli (Ed.): ICALP 2019, CCIS 1108, pp. 108–121, 2019.
https://doi.org/10.1007/978-3-030-32959-4_8

Arabic is one of the five most spoken languages in the world, spoken by more than 422 million native speakers[1]. The situation in Arabic is a classic case of diglossia, whereby the written formal language differs substantially from the spoken vernacular [1,2]. Modern standard Arabic (MSA) is heavily based on Classical Arabic and constitutes the official written language used in government affairs, news, broadcast media, books and education. MSA acts as the lingua franca amongst Arabic native speakers [3]. However, the spoken language (collectively referred to as Dialectal Arabic) widely varies across the Arab world. Moreover, there is neither standard written orthography nor formal grammar for these dialects.

To predict the sentiment of an Arabic piece of text, the majority of the works rely on Machine Learning (ML) algorithms like Linear Support Vector Classification (LinearSVC), Multinomial Naive Bayes (MNB) and others [4–9]. Even though these classifiers are very easy to implement and achieve good results, they require a lot of feature engineering before applying the data to the classifiers. Therefore, work in Arabic sentiment analysis still depends heavily on the morphological and syntactic aspects of the language, such as POS tagging, word stemming, the sentiment lexicons and other hand-crafted features. It was in these areas that there have been several improvements in detecting sentiment.

After the remarkable improvement brought about by Deep Learning (DL) over the traditional ML approaches, researchers tend to investigate and explore the performance of the deep neural networks in analysing different kinds of Arabic texts and extract features for some NLP tasks such as: Language Identification, Text Summarising, Sentiment Analysis and so on [10–12].

In this paper we introduce a deep neural network which combines Bi-directional Long-Short Term Memory Networks (Bi-LSTM) with Convolutional Neural Networks (CNN) to predict the polarity of a text and classify it as either having positive or negative polarity. We exploit some available Arabic sentiment datasets: LABR [13], ASTD [14] and Shami-Senti [15] with different sizes and different dialects. Our system outperforms the state-of-the-art deep learning models for particular datasets like ASTD [16] with improvements on smaller datasets.

The paper is organised as follows: Sect. 2 gives a brief review of existing work that uses deep learning for Arabic Sentiment Analysis. In Sect. 3 we briefly discuss the deep learning architectures and we experiment two baselines: a simple LSTM model and the Kaggle model which uses a combination of LSTM and CNN layers, In Sect. 4, we propose our model and show that it outperforms both baselines, and achieves state-of-the-art results for DL models. Finally, In Sect. 5 we conclude and discuss directions for future work.

[1] http://www.unesco.org/new/en/unesco/events/prizes-and-celebrations/celebrations/international-days/world-arabic-language-day-2013/.

2 Related Work

Sentiment Analysis is usually considered a supervised classification task, where the texts are classified into two or more sentiments classes by providing a dataset with the text and the sentiment label. The common approach in SA is the use of ML through language modelling and feature engineering.

Most of the SA techniques in Arabic use words and character n-gram features with different representation settings and different classifiers [13,17–19]. In some cases, ensemble classifiers are used [20,21]. Moreover, sentiment lexicons are additional and valuable sources of features that have been used to enrich features for SA [22–25].

In [26], the authors introduce a subjectivity and sentiment analysis system for Arabic tweets by extracting different sets of features such as the form of the words (Stem, Lemma), POS tagging, the presence of the sentiment adjective and the Arabic form of the tweet (MSA or DA), in addition to other Twitter-specific features such as the userID (person, organization) and the gender of the user. In [8] a language model is built and different machine learning classifiers are used to handle tweets in MSA and Jordanian.

An early deep learning framework for Sentiment Analysis for Arabic is proposed in [27]. The authors explore several network architectures based on Deep Belief networks, Deep Auto Encoder and the Recursive Auto Encoder. The authors there do not mention the range of labels of the polarity classification. They use The Linguistic Data Consortium Arabic Tree Bank (LDC ATB) dataset and show that the model outperforms the state of the art models on the same dataset by around 9% in terms of F-score. They get an accuracy of 74.5%.

Baly et al. [28] build a deep learning model to detect the polarities of tweets in a 5-scale classification that ranges from very negative to very positive. They retrieve tweets from 12 Arab countries in 4 regions (the Arab Gulf, the Levant, Egypt and North Africa). They collect 470 K tweets. Their deep learning model consists of an embedding layer followed by an LSTM layer. Pre-trained word embeddings are applied using the skip-gram model from Word2Vec. The authors investigate the performance of their model on different morphological forms (lemma and stem). They achieve an accuracy of 70% for the Egyptian tweets and lemma embeddings while for UAE tweets they get 63.7% accuracy.

Soumeur et al. [29] investigate the Sentiment Analysis in the Algerian users' comments on various Facebook brand pages of companies in Algeria. They collect 100 K comments written in Algerian, but they only annotate 25 K comments as positive, negative or neutral. They apply a CNN as a feature extractor and transformation network. Their model consists of three type of layers, three CNN layers each with 50 filters and 3 kernel size, followed by pooling layers and the fully connected layers to predict the sentiment of the comment. Their model achieves an 89.5% accuracy.

SEDAT, a sentiment and emotion analyser model, was built in [30] using Arabic tweets. Word and document embeddings in addition to a set of semantic features are used. All the extracted features into CNN-LSTM networks followed by a fully connected layer are applied. The data has been obtained from the

public datasets for SemEval2018 (Task 1: Affect in Tweets), which has a size of nearly 7 K tweets. The authors further calculate Spearman's correlation coefficient over the baseline models which they outperform with 0.01–0.02 points of difference.

Recently, an ensemble deep learning model was proposed in [16]. There, the authors combine CNN and LSTM models to predict the sentiment of Arabic tweets exploiting Arabic Sentiment Tweets Dataset (ASTD). The model outperforms the state-of-the-art deep learning models F1-score of 53.6%, as they achieve an accuracy of 65% and an F1-score of 64.46%.

3 Deep Learning Baselines for Sentiment Analysis

In this section we present two baseline DL systems for dialectal sentiment analysis. But first, we will talk about the word representation and Deep learning network architectures briefly, in the following subsections.

3.1 Word Representation

Although word embedding vectors are easy to train, there are many pre-trained word vectors that were trained on a large amount of textual data. In this work we use Aravec, which is Arabic pre-trained word embeddings [31]. The Aravec are pre-trained using large data from multiple source like Twitter and Wikipedia and implemented by Word2Vec [32]. Each sample/sentence is replaced by a 2D vector representation of dimension $n \times d$, where n is the number of words in the sentence and d is the length of the embedding vector. After many trials we decided to apply the Aravec-CBOW model of dimension $d = 300$.

3.2 LSTM Network

The traditional Continuous Bag of Word model (CBOW) allows to encode arbitrary length of sequence inputs as fixed-size vectors, but this disregards the order of the features in the sequence [32]. In contrast, Recurrent Neural networks (RNN) represent arbitrary-sized sequences in a fixed-size vector as CBOW, while they pay attention to the structure of the input sequence. Special RNNs with gated architecture such as LSTMs have proven very powerful in capturing statistical regularities in sequential inputs [33].

LSTM is the first network that introduces the gating mechanism and is designed to capture the long-distance dependencies and solve the problem of vanishing gradients [34]. While the LSTM is a feed-forward network that reads the sequence from left to right, the Bidirectional LSTM (Bi-LSTM) connects two layers from opposite directions (forward and backward) over the same output. The output layer receives information from both the preceding sequence (backwards) and following sequence (forward) states simultaneously. It is thus very useful when the context of the input is needed, for example when the negation term appears after a positive term [35].

3.3 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are feature extractor networks that are able to detect indicative local predictors in a large structure [36]. They are designed to combine these predictors to produce a fixed sized vector representation that captures and extracts the most informative local aspects for the prediction task.

For text classification, we use a 1D-CNN, a well known CNN architecture for dealing with sequences. It uses a convolution layer with a window size k that is able to identify the indicative k-grams in the input text, and then act as an n-gram detector [33]. Every CNN applies a nonlinear function called a filter, which transforms a window of size k into scalar values. After applying a multi-filter, the CNN produces m vectors, where each vector corresponds to a filter. Thus, a pooling task is required to combine all of the m vectors into a single m dimension vector. Generally, CNNs focus more on the informative features and disregard their locations in the input text [35,37].

3.4 Datasets

We use the following corpora in our experiments (the characteristics of these corpora are presented in Table 1):

- LABR [13]: it is one of the largest SA datasets to date for Arabic. The data are extracted from a book review website and consist of over 63 k book reviews written mostly in MSA with some dialectal phrases. We use the binary balanced and unbalanced subsets of LABR, in addition to the three-way classification subsets. In LABR, user ratings are used in order to classify sentences. Ratings of 4 and 5 stars are taken by the authors as positive, ratings of 1 and 2 stars are taken as negative and 3 star ratings are taken as neutral. In the binary classification case, 3 star ratings are removed, keeping only the positive and negative labels.
- ASTD [14]: it is an Arabic SA corpus collected from Twitter and focusing on the Egyptian dialect. It consists of approximately 10 k tweets which are classified as objective, subjective positive, subjective negative, and subjective mixed.
- Shami-Senti [15]: a Levantine SA corpus. It contains approximately 2 k posts from social media sites in general topics, classified as Positive, Negative and Neutral from the four main countries where Levantine is spoken: Palestine, Syria, Lebanon and Jordan.

Data Preparation. We apply the following pre-processing steps on all corpora:

1. Remove special characters, punctuation marks and all diacritics;
2. Remove all digits including dates;
3. Remove all repeated characters and keep only two repeated characters, using the algorithm from [15];
4. Remove any non-Arabic characters.

Table 1. The number of instances per category in the corpora used in our experiments

Corpus	NEG	POS	Neutral
Shami-Senti	935	1,064	243
LABR 3 balanced	6,580	6,578	6,580
LABR 2 balanced	6,578	6,580	
LABR 2 Un-Balanced	8,222	42,832	
ASTD	1,496	665	738

We replace every instance sentence with its corresponding word embedding vector from a pre-trained AraVec model [31]. In case of words that occur in the text but do not have an embedding in the pre-trained model, we look for the most similar words, and use them in order to get the corresponding word embeddings vector. More specifically, we look that the distance between the input word and the word in the Aravec model does not exceed two characters either from the beginning or the end of the word. The maximum length of every sentence is fixed to 70 words, thus we apply post-padding with zeros to ensure that all input sentences have the same length.

3.5 LSTM Baseline

This section describes our LSTM baseline. The Keras library has been used for the implementation of all experiments [38]. After many trials we used the Keras checkpoint function to save the best weight model. The checkpoint function automatically stops training when the validation loss starts increasing. After several experiments we decide to use the Adam optimiser with categorical cross entropy loss function for the multi-classification task, and RMSprop[2] for binary classification. The parameters we used in the baseline model are selected after running a number of experiments playing around with the different parameters as shown in Table 2. After many experiments, the parameters that lead to the best result for the baseline are highlighted in bold in the Table 2.

The first experiment we conduct uses a simple LSTM network which consists of an Embedding Layer with pre-trained word embedding followed by two LSTM layer with 128 and 64 output units respectively, followed by a fully connected Relu activation layer with 100 output units and a 0.5 dropout layer. Finally, a dense Sigmoid layer to predict the labels is used.

Table 3 shows the results for various LSTM-BiLSTM models with different combinations. The LSTM → LSTM experiment is the baseline model described above, while in the BiLSTM → LSTM experiment, we change the first layer with a BiLSTM layer. Finally, we try both BiLSTM on the data (BiLSTM → BiLSTM). The model seems to be overfitting the data with the accuracy being very low (less than the 50%). When we apply the baseline model (LSTM → LSTM) on ASTD and ShamiSenti corpora we get a 53% accuracy for both.

[2] https://keras.io/optimizers/.

Table 2. General parameters of deep learning models

Parameter	Value
Dataset split	80% train, 10% development, 10% test
Max number of features	[7K, 10K, **15K**, 25 K, 40K]
Embedding size	[100, **300**]
Embedding model	**CBOW**, Skip-gram
Embedding trainable	**True**, False
Max sample length	[50, **70**, 100]
Filter	[23, 64, 128]
Kernel size	[1, 2, 3, 4, 5, 6]
Pool size	[1, 2, 3, 4, 5]
Batch size	[32, **50**, 100, 128, 256]
Max epoch	10, 50, **100**, 1000
Dropout	0.2, **0.5**, 0.7
Optimiser	**Adam**, **RMSprop**, SGD
Activation function	Softmax, **Sigmoid**, **Relu**
LABR split (train/validation/test)	[70, 10, 20]
ASTD and Shami-Senti split (train/validation/test)	[80, 10, 10]

Table 3. Accuracy of networks with two sequential LSTM/BiLSTM layers for three-way classification

Dataset	Experiment name	Accuracy
LABR 3	LSTM → LSTM (baseline model)	41.9%
LABR 3	BiLSTM → LSTM	42.3%
LABR 3	BiLSTM → BiLSTM	40.6%
ASTD	LSTM → LSTM	53%
Shami-Senti	LSTM → LSTM	53%

Given the low accuracy on the three class task, we investigate the task of binary sentiment classification using BiLSTM → LSTM model from the second experiment on all of datasets (LABR, ASTD, Shami-Senti) as it produces the highest accuracy among all the previous experiments. In the binary task, we employ RMSprop as an optimiser with binary cross entropy loss function. Table 4 shows the results.

Table 4. Accuracy of the BiLSTM → LSTM model with binary classification task on our corpora

Corpus	Test
LABR 2 balanced	55.34%
LABR 2 un-balanced	**81**%
ASTD	**68.5**%
Shami-Senti	54.5 %

The system achieves an unexpected result on the ASTD and LABR 2 unbalanced datasets of 68.5% and 81% accuracy respectively. Table 5 shows the confusion matrix for both of these datasets. Since in the ASTD corpus the negative samples are approximately two-thirds the positive ones, the model tends to predict the negative class as an output label more often than the positive label. Similarly, in the LABR 2 unbalanced the model is biased towards the majority class, i.e. the positive class.

Table 5. Confusion matrix for the BiLSTM → LSTM model for ASTD and LABR 2 unbalanced corpora.

ASTD corpus			
		Predicted	
		Positive	Negative
Actual	Positive	11	**45**
	Negative	23	**136**

LABR 2 unbalanced			
		Predicted	
		Positive	Negative
Actual	Positive	**8036**	505
	Negative	**1555**	114

3.6 Kaggle Baseline

As a next step, we implement the winner model from the Kaggle sentiment analysis competition which was build for English sentiment analysis and has achieved an accuracy of 96%.[3] They used the Amazon Fine Food Reviews dataset, which includes 568,454 reviews, each review has a score from 1 to 5. The model is illustrated in Fig. 1 and consists of a CNN layer with max pooling of size 2 and a dropout layer to exclude some features, followed by one LSTM layer, and at the end, a fully connected layer to predict one output class among 3 sentiment classes (Positive, Negative and Neutral).

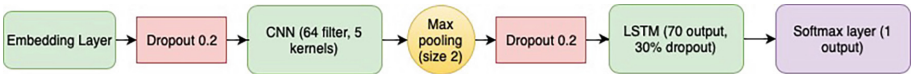

Fig. 1. Kaggle winner model

We train the model using LABR, ASTD and Shami-Senti and apply both three-way and binary classification. The results are shown in Table 6. We get a high accuracy for the LABR 2 unbalanced corpus and the ASTD corpus, 80.6% and 70.7% respectively. Taking a look at the confusion matrix in Table 7, we see that the model does not learn well. Being biased towards the majority class every time, it is clear that the model is over-fitting the training data.

[3] https://www.kaggle.com/monsterspy/conv-lstm-sentiment-analysis-keras-acc-0-96.

Table 6. Accuracy of the Kaggle model on three-way and binary sentiment classification

Corpus	Three-way classification	Binary classification
Shami-Senti	49%	52.3%
LABR 2 unbalanced		**80.6%**
LABR 2 balanced		53.1%
LABR 3	60%	
ASTD	59.3%	**70.7%**

Table 7. Confusion matrix for the Kaggle model on the ASTD and LABR 2 unbalanced corpora.

ASTD corpus			
		Predicted	
		Positive	Negative
Actual	Positive	5	**51**
	Negative	12	**147**

LABR 2 unbalanced			
		Predicted	
		Positive	Negative
Actual	Positive	**8153**	387
	Negative	**1591**	78

4 Our Model

In the previous section we have seen that using a combination of LSTM with a CNN enhances the accuracy of the model. Given these results, we propose a more sophisticated model than the one used in the Kaggle experiments that uses several CNN layers employing different filters and kernels to extract as many features as possible. In addition, we use a BiLSTM to extract the features from both directions and keep track of their effects. In contrast to the Kaggle model, in our model the BiLSTM precedes the CNN layers. We assumed that this configuration would provide a more informative representation of the sequential structure of sentences. The results, we get as shown in Table 8, seem to justify this assumption as our model performs better than Kaggle in all datasets. Figure 2 shows the best performing configuration which consists of an Embedding layer initialised with pre-trained word embedding vectors of size 300 and a max features of 15K, followed by two BiLSTM layers of 128 and 64 output units respectively and 0.5 dropout. The second BiLSTM layer is fed into parallel CNN layers with 5 region sizes (kernels) [2, 3, 4, 5, 6] and 3 filters [32, 64, 128] where we employ *Keras* functional API to build them. Each CNN layer is followed by Global MaxPooling layer. At the end of the CNN network we have a concatenated layer to merger all the outputs into one dimensions vector. This vector feeds into a fully connected Relu layer with 10 output units. Finally, Sigmoid layer with 3 output units for three-way classification and one binary unit for binary classification is used.

Our model achieves high accuracy results for binary sentiment classification in LABR, ASTD and Shami-Senti. The LABR 2 unbalanced dataset again has a high accuracy of 80.2%, when we look to the confusion matrix it is nearly the

Table 8. Accuracy of the proposed model In addition to the comparing results from the two baselines on the three-way and binary sentiment classification

Corpus	Three-way classification			Binary classification		
	Our model	Kaggle	LSTM	Our model	Kaggle	LSTM
Shami-Senti	76.4%	49%	53%	93.5%	25.3%	54.5%
LABR 2 unbalanced				80.2%	80.6%	55.34%
LABR 2 balanced				81.14%	53.1%	81%
LABR 3	66.42%	60%	41.9%			
ASTD	68.62%	59.3%	53%	85.58%	70.7%	68.5%

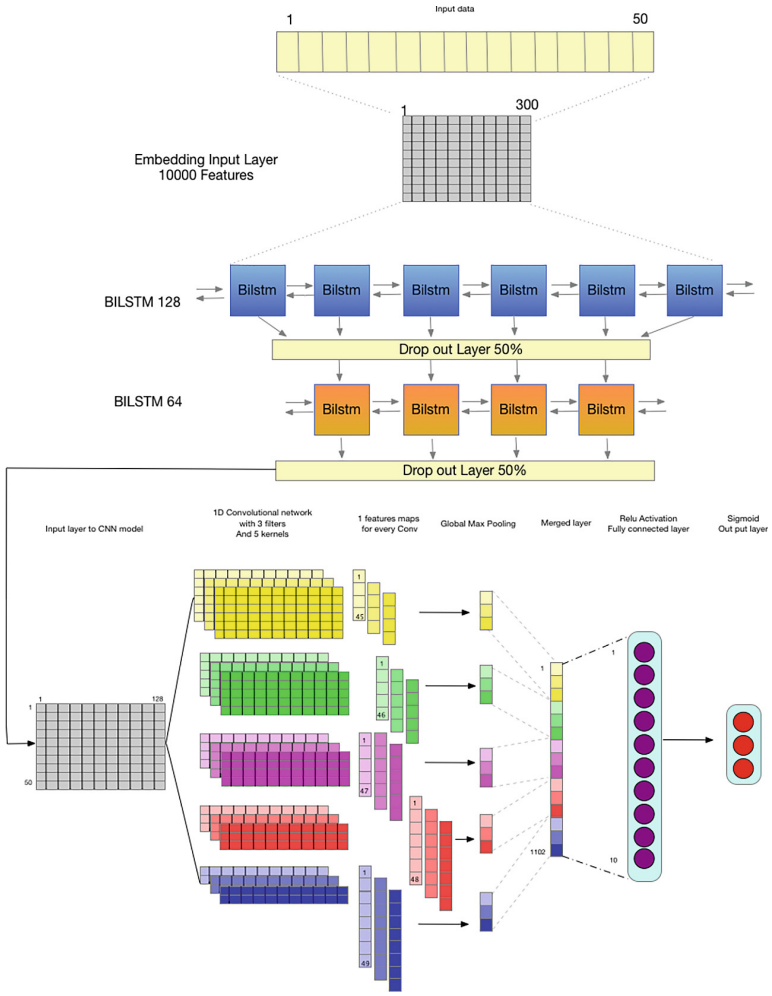

Fig. 2. Final model with BiLSTM and CNN networks

same like the one that has shown in Table 7. It is very clear that the LABR 2 unbalanced dataset does not learn well due to the data imbalance problem, which misleads the performance of the DL network although it has a reasonable size of training data.

Table 9 shows the confusion matrix for the three corpora. Even though the multi-classification results are not very high, our model outperforms the state-of-the-art deep learning models for some corpora like ASTD, where they achieve accuracy of 65% and F-score 64.5% [16]. In our proposed model we get an accuracy of 68.62% and an F-score equal to 69%. Both LABR 3 and ASTD are still suffering from the inaccurate annotation for the third neutral class. They assign the 3 star rating to neutral sentiment which complicates things, given that a 3 star rating might be quite positive or quite negative depending on a number of contextual parameters. This problem makes it hard to achieve very high accuracy when building a multi classification system using these corpora.

Table 9. Confusion matrix for the proposed model in the ASTD, Shami-Senti and the LABR 2 balanced corpora.

ASTD corpus			
		Predicted	
		Pos	Neg
Actual	Pos	46	18
	Neg	13	138

Shami-Senti			
		Predicted	
		Pos	Neg
Actual	Pos	94	4
	Neg	9	93

LABR2 Balanced			
		Predicted	
		Pos	Neg
Actual	Pos	561	80
	Neg	168	506

5 Conclusion and Future Work

In this paper we have investigated the use of Deep Learning architectures for dialectal SA. We first started by experimenting with a simple LSTM architecture on three dialectal SA datasets with poor results. We then took an off-the-shelf SA model that uses a combination of an LSTM and a CNN, i.e. Kaggle, and observed a better performance. Finally, we proposed our own model, which is a more elaborate BiLSTM → CNN with more convolutional layers, and obtained state-of-the-art results on the datasets that DL approaches have been previously applied to (i.e. the ASTD). In general, the results are promising but there is definitely room for improvement, especially on the threeway classification task.

One of the things that we would like to try in the future is the use of word embeddings specifically trained for the SA task, as well as even more complex DL architectures, for example those that use an attention mechanism. Another thing we want to do is to increase ShamiSenti's size, so that it is size-wise comparable to LABR3. It will then be possible to check whether the quality of the data will help the model obtain better accuracy scores, and furthermore check the effect of data size on the model's performance.

Acknowledgements. Kathrein Abu Kwaik, Stergios Chatzikyriakidis and Simon Dobnik are supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg.

References

1. Versteegh, K.: The Arabic Language. Edinburgh University Press, Edinburgh (2014)
2. Ferguson, C.A.: Diglossia. Word **15**(2), 325–340 (1959)
3. Mustafa, S.: The Arabic Language. Routledge, London (2008)
4. Gamal, D., Alfonse, M., El-Horbaty, E.-S.M., Salem, A.-B.M.: Opinion mining for Arabic dialects on twitter. Egypt. Comput. Sci. J. **42**(4), 52–61 (2018)
5. Oussous, A., Lahcen, A.A., Belfkih, S.: Improving sentiment analysis of Moroccan tweets using ensemble learning. In: Tabii, Y., Lazaar, M., Al Achhab, M., Enneya, N. (eds.) BDCA 2018. CCIS, vol. 872, pp. 91–104. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96292-4_8
6. Farra, N., Challita, E., Assi, R.A., Hajj, H.: Sentence-level and document-level sentiment mining for Arabic texts. In: 2010 IEEE International Conference on Data Mining Workshops, pp. 1114–1119. IEEE (2010)
7. Duwairi, R.M.: Sentiment analysis for dialectical Arabic. In: 2015 6th International Conference on Information and Communication Systems (ICICS), pp. 166–170. IEEE (2015)
8. Duwairi, R.M., Marji, R., Sha'ban, N., Rushaidat, S.: Sentiment analysis in Arabic tweets. In: 2014 5th International Conference on Information and Communication Systems (ICICS), pp. 1–6. IEEE (2014)
9. Elarnaoty, M., AbdelRahman, S., Fahmy, A.: A machine learning approach for opinion holder extraction in Arabic language. arXiv preprint arXiv:1206.1011 (2012)
10. Abandah, G.A., Graves, A., Al-Shagoor, B., Arabiyat, A., Jamour, F., Al-Taee, M.: Automatic diacritization of Arabic text using recurrent neural networks. Int. J. Doc. Anal. Recogn. (IJDAR) **18**(2), 183–197 (2015)
11. Lulu, L., Elnagar, A.: Automatic Arabic dialect classification using deep learning models. Procedia Comput. Sci. **142**, 262–269 (2018)
12. Elaraby, M., Abdul-Mageed, M.: Deep models for Arabic dialect identification on benchmarked data. In: Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018), pp. 263–274 (2018)
13. Aly, M., Atiya, A.: LABR: a large scale Arabic book reviews dataset. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 494–498 (2013)
14. Nabil, M., Aly, M., Atiya, A.: ASTD: Arabic sentiment tweets dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2515–2519 (2015)
15. Qwaider, C., Saad, M., Chatzikyriakidis, S., Dobnik, S.: Shami: a corpus of Levantine Arabic dialects. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018) (2018)
16. Heikal, M., Torki, M., El-Makky, N.: Sentiment analysis of Arabic tweets using deep learning. Procedia Comput. Sci. **142**, 114–122 (2018)

17. Mountassir, A., Benbrahim, H., Berrada, I.: An empirical study to address the problem of unbalanced data sets in sentiment classification. In: 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3298–3303. IEEE (2012)
18. Shoukry, A., Rafea, A.: Sentence-level Arabic sentiment analysis. In: 2012 International Conference on Collaboration Technologies and Systems (CTS), pp. 546–550. IEEE (2012)
19. Elawady, R.M., Barakat, S., Elrashidy, N.M.: Different feature selection for sentiment classification. Int. J. Inf. Sci. Intell. Syst. **3**(1), 137–150 (2014)
20. Omar, N., Albared, M., Al-Shabi, A.Q., Al-Moslmi, T.: Ensemble of classification algorithms for subjectivity and sentiment analysis of Arabic customers' reviews. Int. J. Adv. Comput. Technol. **5**(14), 77 (2013)
21. Al-Saqqa, S., Obeid, N., Awajan, A.: Sentiment analysis for Arabic text using ensemble learning. In: 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), pp. 1–7. IEEE (2018)
22. Al-Ayyoub, M., Khamaiseh, A.A., Jararweh, Y., Al-Kabi, M.N.: A comprehensive survey of Arabic sentiment analysis. Inf. Process. Manag. **56**(2), 320–342 (2019)
23. Abdul-Mageed, M., Diab, M.T., Korayem, M.: Subjectivity and sentiment analysis of modern standard Arabic. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2, pp. 587–591. Association for Computational Linguistics (2011)
24. Badaro, G., et al.: A light lexicon-based mobile application for sentiment mining of Arabic tweets. In: Proceedings of the Second Workshop on Arabic Natural Language Processing, pp. 18–25 (2015)
25. Badaro, G., Baly, R., Hajj, H., Habash, N., El-Hajj, W.: A large scale Arabic sentiment lexicon for Arabic opinion mining. In: Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP), pp. 165–173 (2014)
26. Abdul-Mageed, M., Diab, M., Kübler, S.: SAMAR: subjectivity and sentiment analysis for Arabic social media. Comput. Speech Lang. **28**(1), 20–37 (2014)
27. Al Sallab, A., Hajj, H., Badaro, G., Baly, R., El Hajj, W., Shaban, K.B.: Deep learning models for sentiment analysis in Arabic. In: Proceedings of the Second Workshop on Arabic Natural Language Processing, pp. 9–17 (2015)
28. Baly, R., et al.: Comparative evaluation of sentiment analysis methods across Arabic dialects. Procedia Comput. Sci. **117**, 266–273 (2017)
29. Soumeur, A., Mokdadi, M., Guessoum, A., Daoud, A.: Sentiment analysis of users on social networks: overcoming the challenge of the loose usages of the Algerian dialect. Procedia Comput. Sci. **142**, 26–37 (2018)
30. Abdullah, M., Hadzikadicy, M., Shaikhz, S.: SEDAT: sentiment and emotion detection in Arabic text using CNN-LSTM deep learning. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 835–840. IEEE (2018)
31. Soliman, A.B., Eissa, K., El-Beltagy, S.R.: AraVec: a set of Arabic word embedding models for use in Arabic NLP. Procedia Comput. Sci. **117**, 256–265 (2017)
32. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
33. Goldberg, Y.: Neural network methods for natural language processing. Synth. Lect. Hum. Lang. Technol. **10**(1), 1–309 (2017)
34. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning, pp. 1310–1318 (2013)

35. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
36. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press, Cambridge (2016)
37. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
38. Gulli, A., Pal, S.: Deep Learning with Keras. Packt Publishing Ltd., Birmingham (2017)