Can Modern Standard Arabic Approaches be used for Arabic Dialects? Sentiment Analysis as a Case Study

Kathrein Abu kwaik Stergios Chatzikyriakidis Simon Dobnik
CLASP and FLOV, University of Gothenburg, Sweden
{kathrein.abu.kwaik,stergios.chatzikyriakidis,simon.dobnik}@gu.se

Abstract

We present the Shami-Senti corpus, the first Levantine corpus for Sentiment Analysis (SA), and investigate the usage of off-the-shelf models that have been built for Modern Standard Arabic (MSA) on this corpus of Dialectal Arabic (DA). We apply the models on DA data, showing that their accuracy does not exceed 60%. We then proceed to build our own models involving different feature combinations and machine learning methods for both MSA and DA and achieve an accuracy of 83% and 75% respectively.

1 Introduction

There is a growing need for text mining and analytical tools for Social Media data, for example Sentiment Analysis (SA) tools which aim to distinguish people's views into positive and negative, objective and subjective responses, or even into neutral opinions. The amount of internet documents in Arabic is increasing rapidly (Ibrahim et al., 2015; Abdul-Mageed et al., 2011; Abdul-Mageed and Diab, 2011; Mourad and Darwish, 2013). However, texts from Social media are typically not written in Modern Standard Arabic (MSA) for which computational resources and corpora exist. These systems achieve reasonable accuracy on the designated tasks. For example, Abdul-Mageed et al. (2011) achieve an accuracy of 95% on the news domain. On the other hand, research on Dialectal Arabic (DA) in terms of SA is an open research question and presents considerable challenges (Badaro et al., 2019; Ibrahim et al., 2015).

The degree to which tools trained on MSA can be used on DA is still also an open research question. This is partly because different dialects differ from MSA to varying degrees(Kwaik et al., 2018). Furthermore, the speakers of Arabic present us with clear cases of Diglossia (Ferguson, 1959), where MSA is the official language used for education, news, politics, religion and, in general, in any type of formal setting, but dialects are used in everyday communication, as well as in informal writing (Versteegh, 2014).

In this paper, we examine whether it is possible to adapt classification models that have been trained and built on MSA data for DA from the Levantine region, or whether we should build and train specific models for the individual dialects, therefore considering them as stand-alone languages. To answer this question we use Sentiment Analysis as a case study. Our contributions are the following:

- We systematically evaluate how well the ML models on MSA for SA perform on DA of Levantine;
- We construct and present a new sentiment corpus of Levantine DA;
- We investigate the issue of domain adaptation of ML models from MSA to DA.

The paper is organised as follows: in Section 2, we briefly discuss the task of SA and present related work on Arabic. In Section 3, we describe an extension of the Shami corpus of Levantine dialects (Qwaider et al., 2018) annotated for Sentiment, Shami-Senti. In Section 4, we present the experimental setting and results of adapting MSA models to the dialectal domain as well as training specific models. We conclude and discuss directions for future research in Section 5.

2 Arabic Sentiment Analysis

Manually gathering information about users' opinions and sentiment data is time-consuming. This is why more and more companies and organisations are interested in automatic SA methods to help them understand it. SA refers to the usage of variety of tools from Natural Language Processing (NLP), Text Mining and Computational Lin-

guistics to examine a given piece of text and identify the dominant sentiment subjectivity in it (Liu, 2012; Ravi and Ravi, 2015). SA is usually categorised into three main sentiment polarities: Positive (POS), Negative (NEG) and Neutral (NUT). SA is frequently used interchangeably with Opinion Mining (Abdullah and Hadzikadic, 2017).

At first glance, Sentiment Analysis is a classification task. It is a complex classification task as if one dives deeper, they are faced with a number of challenges that affect the accuracy of any SA model. Some of these challenges are: (i) Negation terms (Farooq et al., 2017), (ii) Sarcasm (Ghosh and Veale, 2016), (iii) Word ambiguity and (iv) Multi-polarity.

As a result of the rapid development of social media and the use of Arabic dialectal texts, there is an emerging interest in DA. Farra et al. (2010) propose a model of sentence classification (SA) in Arabic documents. They extract sets of features and calculate the total weight for every sentence. A J48 Decision tree algorithm is used to classify the sentences w.r.t. sentiment, achieving an accuracy of 62%.

Gamal et al. (2018) collect tweets from different Arabic regions using different keywords and phrases. The tweets include opinions about a variety of topics. They annotate their polarity by checking if they contain positive or negative terms and without considering the reverse polarity in the presence of negation terms. Then, they apply six machine learning algorithms on the data and achieve an accuracy between 82% and 93%.

Oussous et al. (2018) build an SA model to classify the sentiment of sentences. The authors construct a Moroccan corpus, where the data are collected from Twitter, and annotate it. Multiple algorithms are used, e.g. Support Vector Machines (SVM), Multinomial Nave Bayes (MNB) and Mean Entropy (ME). The SVM model achieves an accuracy of 85%. Ensemble learning by majority voting and stacking is also tried. Using the three aforementioned algorithms in the two models, they attain an accuracy of 83% and 84% respectively. Another work using the same classifiers is described in (El-Halees, 2011). The dataset covers three domains: education, politics and sports. The resulting accuracy is 80%.

A framework for Jordanian SA is proposed in (Duwairi et al., 2014). The authors create a corpus of Jordanian tweets and build a mapping lexicon from Jordanian to MSA that turns any dialectal word into an MSA word, before classifying the tweet. In order for the tweets to be annotated, crowd-sourcing is used. They further use Rapid Miner for pre-processing, filtering, and classification. Three classifiers are used to evaluate the performance of the proposed framework with 1000 tweets: Nave Bayes (NB), (SVM) and k-nearest neighbour (KNN). The NB model gets the highest accuracy with 76.78%.

Binary sentiment classification for Egyptian using a NB classifier is investigated in (Abdul-Mageed et al., 2011). An accuracy of 80% is achieved. Similarly, the Tunisian dialect is addressed in (Medhaffar et al., 2017). Here, the authors create a Tunisian corpus for SA containing 17K comments from social media. Applying Multi-Layer Perceptron (MLP) and SVM on the corpus they get 0.22 and 0.23 error rate respectively. Another line of work addresses the Saudi dialects (Al-Twairesh et al., 2018; Rizkallah, Sandra and Atiya, Amir and ElDin Mahgoub, Hossam and Heragy, Momen", editor="Hassanien, Aboul Ella and Tolba, Mohamed F. and Elhoseny, Mohamed and Mostafa, Mohamed , 2018) and some addresses the United Arab Emirates dialects (Baly et al., 2017a,b).

Several works exploit lexicon-based sentiment classifiers for Arabic. A sentiment lexicon is a lexicon that contains both positive and negative terms along with their polarity weights (Badaro et al., 2014; Abdul-Mageed and Diab, 2012; Badaro et al., 2018). The SAMAR system (Abdul-Mageed et al., 2014) involves two-stage classification based on a sentiment lexicon. The first classifier detects subjectivity and objectivity of documents, which is followed by another classifier to detect the polarity. They employ different datasets and examine various features combinations. Similar work is reported in (Mourad and Darwish, 2013; Al-Rubaiee et al., 2016), where both NB and SVM are explored, achieving an accuracy between 73% and 84% .

Abdulla et al. (2013) compare the performance of corpus-based sentiment classification and lexicon-based classification in Arabic. The accuracy of the lexicon approach does not exceed 60%. They conclude that corpus-based methods perform better using SVM and light stemming.

Overall, there is a considerable amount of work on SA and DA but none of these approaches considered the performance of the classifiers across the domains for which limited data exist.

Lexicon	Negative	Positive	Negation
LABR	348	319	37
Moarlex	13411	4277	
SA lexicon	3537	855	

Table 1: The number of terms in sentiment lexicons

3 Building Shami-Senti

The question of sentiment analysis has not yet been fully examined for Levantine dialects: Palestinian, Jordanian, Syrian and Lebanese. For this reason, we extend the Shami corpus (Qwaider et al., 2018) by annotating part of it for sentiment. We call the new corpus Shami-Senti.

We build Shami-Senti as follows:

1. Manually extract sentences that contains sentiment words, reviews, opinions or feelings from the Shami corpus;
2. Split the sentences and remove any misleading words or very long phrases (set sentences be no longer than 50 words);
3. Try to avoid ironic and sarcastic text where the intended sentiment is reversed. For example, sentences like the following: تصدقوا احنا ناكرين الجميل الرجال زي الفل "I believe we are ungrateful, this man is perfect" (Karoui et al., 2017), are avoided.

3.1 Sentiment annotation

Two methods have been used to annotate the corpus, a lexicon-based annotation and human annotation. The sentence is marked as positive if it contains positive terms or negated negative terms. It is considered negative if it contains negative terms or negation of positive terms. Any sentence that contains a mixture of positive and negative terms or no sentiment terms is marked as mixed or neutral.

In the lexicon-based annotation, we use three sentiment lexicons: the one provided by LABR (Aly and Atiya, 2013) which contains negative, positive and negated terms; the Moarlex (Youssef and El-Beltagy, 2018) and the SA lexicon (ElSahar and El-Beltagy, 2014) which contain only positive and negative terms. Table 1 illustrates the numbers of terms in each lexicon.

First, for the lexicon-based annotation we extracted 1,000 sentences from the Shami corpus and commissioned a Levantine native speaker to annotate them for sentiment. Then, we implemented Algorithm 1 to automatically annotate the same 1,000 sentences. We computed the inter-annotator agreement but the result was very bad, the disagreement was up to 80%. As a result, we did not consider this method as reliable for annotation, hence we chose to annotate the data set manually.

Result: Annotate 1,000 sentences
Build Positive, Negative, Negation lists of words extracted from the three lexicons;
Polarity = 0;
for *sentence in Shami-Senti* **do**
  count number of positive terms; Then Polarity $++$;
  count number of negative terms; Then Polarity $--$;
  check if there is a negation,Then Polarity $* -1$;
  if *Polarity* > 0 **then**
    Polarity is Positive;
  else if *Polarity* < 0 **then**
    Polarity is negative;
  else
    Polarity is mixed;
  end
end

Algorithm 1: Lexicon-based annotation of 1,000 Shami sentences

For the human annotation method, we asked two native speakers, one from Palestine and another from Syria, to annotate 533 sentences with 1 if these are positive, 0 if negative and -1 if neutral or mixed sentences. Then we calculated the inter-annotator agreement between them using Kappa statistics (Carletta, 1996) giving us $\kappa = 0.838$ which is a very good agreement. Since the data was split into separate dialects, we asked the annotators to annotate the parts that they were most familiar with, for example, the Palestinian speaker annotated the sentences in Palestinian and Jordanian, while the Syrian speaker annotated the Syrian and Lebanese sentences. We extracted more than 5,000 sentences/tweets for this purpose, and have annotated nearly 2,000 of them so far. Table 2 shows the number of tweets per category.

4 Experiments

In order to estimate the performance of the SA models, which have built on MSA data, on DA evaluation data, we use the following two corpora in our experiments.

- LABR (Aly and Atiya, 2013): this is one of the largest SA datasets to-date for Arabic. It consists of over 63k book reviews written

Corpus	NEG	POS	Mix
Shami-Senti	935	1064	243
LARB 3 Balanced	6580	6578	6580
LABR 2 Balanced	6578	6580	
ASTD	1496	665	738

Table 2: The number of instances per category in Shami-Senti and other sentiment corpora used in our experiments

in MSA with some dialectal words. LABR is available with different subsets: the authors split it into 2,3,4 and 5 sentiment polarities with balanced and unbalanced divisions. They depend on the user ratings to classify sentences. Thus, 4 and 5 stars ratings are taken as positive, 1 and 2 star ratings are taken as negative and 3 star ratings are taken as mixed or neutral. The fact that LABR is limited to one domain, book reviews, makes it difficult to use it as a general SA model.

- ASTD (Nabil et al., 2015): it is an Arabic SA corpus collected from Twitter and focuses on the Egyptian dialects. It consists of about 10k tweets, which are classified as objective, subjective positive, subjective negative, and subjective mixed.

Table 2 shows the number of instances of each polarity label in different corpora.

In all experiments, we use the same machine learning algorithms that have been used by the LABR baseline. These are:

1. Logistic Regression (LR)
2. Passive Aggressive (PA)
3. Linear Support Vector classifier (LinearSVC)
4. Bernoulli Naive-Bayes (BNB)
5. Stochastic Gradient Descent (SGD)

The choice is motivated as follows. LR is strong in explaining the relationship between one dependent variable and independent variables (Feng et al., 2014), while PA is suitable for large-scale learning (Crammer et al., 2006). LinearSVC is effective in cases where the number of dimensions is greater than the number of samples (Kumar and Goel, 2015). BNB is suitable for discrete data (Shimodaira, 2014), and SGD is a linear classifier which implements regularised linear models with stochastic gradient descent (SGD) learning. It is a simple baseline classifier related to neural networks (Günther and Furrer, 2013).

In addition, we also use some popular linear and probabilistic classifiers. Hence, we use Multinomial Naive-Bayes (MNB), which is suitable for classification of discrete features. The multinomial distribution normally requires integer feature counts and it works well for fractional counts like tf-idf (Xu et al., 2017). We further use Complement Naive-Bayes (CNB), which is particularly suited for imbalanced data sets. CNB uses statistics taken from the complement of each class to compute the models weights.[1] Generally speaking, a NB classifier converges quicker than discriminative models like logistic regression, so one need less training data. The last one is the Ridge Classifier (RC). Its most important feature is that it does not remove irrelevant features but rather minimise their impact on the trained model (Drucker et al., 1997). All of the algorithms are implemented using the `scikit learn` library in Python (Pedregosa et al., 2011) .

4.1 Three class sentiment classification

We start with the baseline from LABR, and use the 3-class balanced data set. Table 3 states the number instances of each polarity class for both training and testing. The baseline method from LABR uses the language model to predict the polarity class. We conduct two experiments: one with unigrams, and one with both unigrams and bigrams. We build the models by transforming the data into a numerical vectors using the Term Frequency vectorize method. First, a Language Model is built by extracting unigrams and bigrams from the dataset and computing their term-frequencies to create the two models, the unigrams, and the combined unigrams and bigrams. Then, every sentence goes through a classifier which produces a probability of the class the sentence belongs to. Table 4 shows the accuracy of the classifiers on the test set trained on the 3-class balanced LABR. The unigram and bigram TF method is doing marginally better than the unigram language model, particularly with the PA classifier. The four classifiers achieve an accuracy between 58% and 59% to classify MSA sentences. BNB is the worst performing classifier with 35% and 34% accuracy respectively. The reason for this might be that we have a large number of features (i.e. individual words) and since BNB models are counting the words that are not present in the document they do not perform well.

[1] https://scikit-learn.org/dev/modules/naive_bayes.html

	Positive	Negative	Mix
Train	4936	4935	4936
Test	1644	1643	1644

Table 3: The number of instances per category in balanced LABR3

Classifier	Accuracy TF_wg1	Accuracy TF_wg1+2
Logistic Regression	59	59
Passive Aggressive	54	58
Linear SVC	57	58
Bernoulli NB	35	34
SGD Classifier	59	59

Table 4: Accuracy of the baseline on LABR3 (Tf-wg : is the Term Frequency on Word grams)

	Training Dataset	
Classifier	**LABR3**	**Shami-Senti**
Logistic Regression	46	62
Passive Aggressive	43	64
Linear SVC	44	64
Bernoulli NB	11	48
SGD Classifier	45	65

Table 5: Accuracy of the baseline TF_wg1+2 trained on LABR3 and Shami-Senti and tested on Shami-Senti

MSA has been researched more from an NLP perspective than DA, and therefore several sentiment analysis approaches have been built for it. The question we want to ask, is whether we can apply these NLP approaches directly on DA or new resources and models are required for DA. We, thus, test the reliability of models that are built on MSA data and adapt them to DA data. Here, we test the baseline bigram TF model on the test part of the Shami-Senti corpus. Table 5 shows the accuracy from this experiment where we trained the baseline by LABR3 and tested it using Shami-Senti. The accuracy is significantly worse, with a drop of more than 10%. The table also shows the accuracy of the baseline when we trained and tested it on Shami-Senti. The highest accuracy was 65% using SGD classifier.

Given the baseline model's poor performance on DA, we build a new SA model. This model also depends on language modelling, where we use a combination of both word-level and character-level n-grams. After several experiments, we observe that a language model that combines features of word-level unigrams and bigrams with character-level n-grams from 2 to 5 gives the best accuracy. We test eight different machine learning algorithms to predict sentiment classification.

Classifier	Model 1	Model 2
Ridge Classifier	57	59
Logistic Regression	59	60
Passive Aggressive	55	58
Linear SVC	57	59
SGD Classifier	59	60
Multinomial NB	57	59
Bernoulli NB	49	49
Complement NB	57	59

Table 6: Accuracy of the proposed model trained and tested on LABR3; Model 1: unigram word level with (2,5) character grams; In Model 2 (unigram,bigrams) word level with (2,5) character grams

Classifier	Accuracy
Ridge Classifier	43
Logistic Regression	46
Passive Aggressive	43
Linear SVC	45
SGD Classifier	50
Multinomial NB	40
Bernoulli NB	44
Complement NB	42

Table 7: Accuracy of the proposed model trained on LABR3 and tested on Shami-Senti

Table 6 shows the accuracy of our model on the LABR 3-class balanced dataset. In Model 1, we test using only unigram words and character grams from 2 to 5, while in Model 2 we add an extra bigram word-level to Model 1. The SGD and LR classifiers give the highest accuracy 60% on Model 2 which is slightly higher than the base line where it was 59%. In all experiments later we will refer to Model 2 as our proposed model. We test this model which was trained on LABR 3 on Shami-Senti. Table 7 shows the results. The model is not performing well on DA achieving an accuracy of 50% using the SGD classifier. This indicates that MSA models are not transferable to DA.

We also train the selected classifier configurations on the Shami-Senti corpus (Table 8). NB algorithms give the highest accuracy with 71%,

Classifier	Accuracy
Ridge Classifier	69
Logistic Regression	67
Passive Aggressive	68
Linear SVC	69
SGD Classifier	68
Multinomial NB	71
Bernoulli NB	71
Complement NB	71

Table 8: Accuracy of the proposed model 3-class classification trained and tested on Shami-Senti

while the differences between the classifiers are marginal. We train the model using 1,000 samples and get an accuracy of 69% by MNB which indicates that increasing the size of the data set has a significant impact on the model accuracy.

4.2 Binary Sentiment classification

The accuracy obtained for the 3-class classification is not very high. This seems to be, at least partly, because the mixed class contains both positive and negative examples which makes the classification task difficult. LABR considers a 3-star rating as a mixed or neutral class. This is not very accurate since, in some cases, users use this rating as negative, while in others as somewhat positive. Table 9 shows three samples from the third neutral class in LABR that we consider should potentially belong to different classes.

We reduce the classification to a binary classification task, by focusing on the positive and negative classes only. Using the LABR, we build a baseline with bigram word counts and another model based on term frequency of unigram and bigram words. After that, we build a unigram and bigram TF words model and a (2-5) TF character model (the proposed model) and apply the LABR 2 classes dataset. The accuracy for the three models, in addition the accuracy of the same models tested on Shami-Senti are shown in Table 10.

We also test the transfer of models between different dialects. We train the classifiers with the proposed configurations to build a model on the ASTD corpus that contains Egyptian dialect data, and test it on both the ASTD and the Shami-Senti corpus. The results are shown in Table 11. The proposed model gives an accuracy up to 83% using linear classifiers like SVC and SGD when it is trained and tested on MSA LABR data set, while it gives an accuracy up to 58% when it is tested on Shami-Senti. We also get an accuracy of 83% when we train and test the model on the ASTD corpus and using an MNB classifier and 57% accuracy when we test it on Shami-Senti.

Models which are trained and built on MSA data can not fit well in dialectal data, even though both of them are considered similar languages. The accuracy for any model tested on Shami-Senti does not exceed 60% (Table 10 and Table 11) in all experiments. Table 12 shows that the model works better for binary sentiment classification with 74% accuracy using MNB, when the model is trained and tested on Shami-Senti. The high accuracy could be due to the quality of the data and human performed annotations. The high accuracy achieved (83%) on both LABR and ASTD indicates that increasing the size of the corpus improves the classification task.

4.3 Feature engineering

In order to improve 3-class sentiment classification, we consider adding more features to the language model. The classifiers with the new features are applied to both the LABR and the Shami-Senti corpus. Based on the three lexicons, (LABR, Moarlex and SA lexicon) we count the number of positive and negative terms in the sentence, and then calculate their probability using Equation 1 and 2. In addition, we use an additional binary feature to indicate if the sentence contains a negation term or not.

$$P(POS) = \frac{\#pos\_terms\_in\_the\_sentence}{total\_length} \quad (1)$$

$$P(NEG) = \frac{\#neg\_terms\_in\_the\_sentence}{total\_length} \quad (2)$$

The three extra features and the word and character n-gram features are combined through the FeatureUnion estimator function in scikit-learn [2] to build and train the models. After many trials we chose to specify the weight of the transformer matrix to 0.4 for the positive feature, 0.2 for the negative feature, 0.4 for the negation feature and 2 for the language model features. The weight for the language module feature is doubled in order to increase their impact. Table 13 shows the result for the SGD and MNB classifiers on both the MSA and Shami corpus. On the MSA data set we get an accuracy of 58.1% and 58.2% using SGD and MNB respectively, which is not a valuable improvement compared to the results in Table 4.

[2] https://scikit-learn.org/0.18/modules/pipeline.html

Sentence		Corrected Polarity
Arabic	بعض الكلمات استوقفتني وجعلتني أفكر وبعضها الاخر جعلني أبتسم. والبعض جعلني أغرق في الضحك. اشتقت لهذا الأسلوب في الكتابة	Positive
English	Some words stopped me and made me think. Some of them made me smile. And some made me drowned in laughter !!! I missed this method in writing.	
Arabic	الكتاب ليس بسيء ولكنه أثار ضجة اعلانية أكثر من اللازم	Mix
English	The book is not bad but it has too much publicity more than it deserves	
Arabic	بالكاد اكملتها تفاصيلها كثيرة ومبهمة ومملة وبشعه جدا أشبه بالكوابيس	Negative
English	Barely completed, the details are many, opaque, boring and very ugly like nightmares	

Table 9: Examples annotated as neutral in LABR3 and our corrected polarity

	counting 2g		TF_wg 1+2		OUR Model	
Classifier	**LABR**	**Shami**	**LABR**	**Shami**	**LABR**	**Shami**
Ridge Classifier	78	53	81	54	83	57
Logistic Regression	80	57	80	56	82	58
Passive Aggressive	78	53	81	53	82	56
Linear SVC	78	55	81	55	83	58
SGD Classifier	80	53	82	54	83	56
Multinomial NB	78	52	80	53	82	55
Bernoulli NB	76	48	76	47	74	48
Complement NB	78	51	80	53	82	55

Table 10: Accuracy for binary classifiers with different feature sets trained on the LABR2 dataset and tested on LABR2 and Shami-Senti

	Testing Dataset	
Classifier	**ASTD**	**Shami-Senti**
Ridge Classifier	81	55
Logistic Regression	77	55
Passive Aggressive	82	57
Linear SVC	81	56
SGD Classifier	82	56
Multinomial NB	83	57
Bernoulli NB	82	58
Complement NB	82	58

Table 11: Accuracy of the proposed model on binary classification trained on ASTD and tested on ASTD and Shami-Senti

Classifier	**2 classes**
Ridge Classifier	73
Logistic Regression	74
Passive Aggressive	73
Linear SVC	73
SGD Classifier	73
Multinomial NB	74
Bernoulli NB	72
Complement NB	75

Table 12: Accuracy of the proposed model on binary classification trained and tested on Shami-Senti

On the dialectal data set, the accuracy of the SGD classifier is decreased from 68% in Table 8 to 66%. We hypothesise that this is because of the lexicon which includes primarily MSA terms and Egyptian terms rather than Levantine sentiment terms so the probabilities of features are less accurate. Even though, MNB is still able to improve the classification accuracy from 71% to 75.2%.

The effect of feature engineering has more effect on the dialectal data, as the size of the dataset

	F.Eng	
Classifier	**LABR**	**Shami**
SGD Classifier	58.1	66
Multinomial NB	58.2	75.2

Table 13: Accuracy of two classifiers using feature engineering on 3-class classification task

	Accuracy	
Experiment name	**LABR**	**Shami-Senti**
LSTM(100)	42	64.7
BiLSTM(200)	41.3	61.8

Table 14: Accuracy of deep learning models 3-class LABR and Shami-Senti

plays an important rule. Adding more informative features to a small dataset help the system to learn and predict the correct class.

4.4 Deep learning models

Deep learning has emerged as a powerful machine learning technique and has already produced state-of-the-art prediction results for SA (Zhang et al., 2018; Rojas-Barahona, 2016; Tang et al., 2015). In this section, we conduct a small experiment implemented using the Keras library to test two standard deep learning models to classify sentiment in our datasets.

The first model is a Long Short-Term Memory (LSTM) model. It consists of:

1. an embedding layer with max_features (MF) equal to the maximum number of words (7000), weighted matrix which is a 7000 * 100 matrix extracted from Aravec, a pre-trained Arabic word embedding model (Soliman et al., 2017), and max_lenght = 50 as the maximum number of words in each sentence;
2. an LSTM layer with an output of 100 and 50% of dropout rate;
3. a dense layer with an output of 30 followed by a final sigmoid layer with 3 sentiment classes.

The second model, BiLSTM(200), uses a Bidirectional LSTM layer with an output of 200 rather than an LSTM layer with an output of 100. We train the model using the Adam optimiser and a batch size of 50. We train the two models on the LABR3 balanced corpus. In addition, we do the same experiments on Shami-Senti. Table 14 shows the results for both datasets.

The test accuracy, in general, is not at the desired level. It is clear that feature-based machine learning classifiers outperform deep learning networks.

5 Conclusion and future work

In this paper, we have investigated different ML algorithms and built a model for SA that combines word n-grams with character n-grams, in addition to other supportive features. The model outperforms the baseline on both big and small datasets, and gets an accuracy of 83% for MSA and 75.2% for Shami-Senti. What is more important, we have shown that using a model trained on MSA SA data and then testing it on dialectal SA data, does not produce good results. This suggests that MSA models cannot be easily, if at all, used in dealing with DA. There is, thus, a growing need for the creation of computational resources, not only for MSA, but also for DA. The extent of this need, and whether some resources can be re-used up to some point, is something that needs to be further investigated. In the case we have been looking at in this paper, it seems that the existing MSA approaches will not be very usable when thrown at dialectal data. It goes without saying that the same situation holds when one tries to use computational resources used for a specific dialect of Arabic to another one, modulo the closeness (in some computational measure to be defined) between the two varieties.

In the future, we plan to continue our work on the annotation of the Shami-Senti corpus exploiting more automatic ways and aiming at enhancing it in terms of size, quality and distribution. Once this happens, we plan to investigate the application of the same deep learning models used in this paper, as well as more sophisticated ones. On a similar note, we are currently working on using more sophisticated deep learning models for the same sized dataset we have been using in this paper. This is part of a more general question of using deep learning with small datasets: whether such an endeavour is possible, and if yes, what are the techniques and network tweaks that make this possible.

Acknowledgements

The authors 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

Muhammad Abdul-Mageed and Mona Diab. 2012. Toward building a large-scale Arabic sentiment lexicon. In *Proceedings of the 6th international global WordNet conference*, pages 18–22.

Muhammad Abdul-Mageed, Mona Diab, and Sandra Kübler. 2014. SAMAR: Subjectivity and sentiment analysis for Arabic social media. *Computer Speech & Language*, 28(1):20–37.

Muhammad Abdul-Mageed and Mona T Diab. 2011. Subjectivity and sentiment annotation of modern standard Arabic newswire. In *Proceedings of the 5th linguistic annotation workshop*, pages 110–118. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Mona T Diab, and Mohammed Korayem. 2011. 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*, pages 587–591. Association for Computational Linguistics.

Nawaf A Abdulla, Nizar A Ahmed, Mohammed A Shehab, and Mahmoud Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan conference on applied electrical engineering and computing technologies (AEECT)*, pages 1–6. IEEE.

Malak Abdullah and Mirsad Hadzikadic. 2017. Sentiment analysis on Arabic tweets: Challenges to dissecting the language. In *International Conference on Social Computing and Social Media*, pages 191–202. Springer.

Hamed Al-Rubaiee, Renxi Qiu, and Dayou Li. 2016. Identifying Mubasher software products through sentiment analysis of Arabic tweets. In *2016 International Conference on Industrial Informatics and Computer Systems (CIICS)*, pages 1–6. IEEE.

Nora Al-Twairesh, Hend S. Al-Khalifa, AbdulMalik Al-Salman, and Yousef Al-Ohali. 2018. Sentiment Analysis of Arabic Tweets: Feature Engineering and A Hybrid Approach. *CoRR*, abs/1805.08533.

Mohamed Aly and Amir Atiya. 2013. 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)*, volume 2, pages 494–498.

Gilbert Badaro, Ramy Baly, Hazem Hajj, Wassim El-Hajj, Khaled Bashir Shaban, Nizar Habash, Ahmad Al-Sallab, and Ali Hamdi. 2019. A Survey of Opinion Mining in Arabic: A Comprehensive System Perspective Covering Challenges and Advances in Tools, Resources, Models, Applications, and Visualizations. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(3):27.

Gilbert Badaro, Ramy Baly, Hazem Hajj, Nizar Habash, and Wassim El-Hajj. 2014. A large scale Arabic sentiment lexicon for Arabic opinion mining. In *Proceedings of the EMNLP 2014 workshop on Arabic Natural Language Processing (ANLP)*, pages 165–173.

Gilbert Badaro, Hussein Jundi, Hazem Hajj, Wassim El-Hajj, and Nizar Habash. 2018. ArSEL: A Large Scale Arabic Sentiment and Emotion Lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Ramy Baly, Gilbert Badaro, Georges El-Khoury, Rawan Moukalled, Rita Aoun, Hazem Hajj, Wassim El-Hajj, Nizar Habash, and Khaled Shaban. 2017a. A characterization study of Arabic twitter data with a benchmarking for state-of-the-art opinion mining models. In *Proceedings of the third Arabic natural language processing workshop*, pages 110–118.

Ramy Baly, Georges El-Khoury, Rawan Moukalled, Rita Aoun, Hazem Hajj, Khaled Bashir Shaban, and Wassim El-Hajj. 2017b. Comparative evaluation of sentiment analysis methods across Arabic dialects. *Procedia Computer Science*, 117:266–273.

Jean Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 2(22):249–254.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585.

Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *Advances in neural information processing systems*, pages 155–161.

Rehab M Duwairi, Raed Marji, Narmeen Sha'ban, and Sally Rushaidat. 2014. Sentiment analysis in Arabic tweets. In *2014 5th International Conference on Information and Communication Systems (ICICS)*, pages 1–6. IEEE.

Alaa M El-Halees. 2011. Arabic opinion mining using combined classification approach. *Arabic opinion mining using combined classification approach*.

Hady ElSahar and Samhaa R El-Beltagy. 2014. A fully automated approach for Arabic slang lexicon extraction from microblogs. In *International conference on intelligent text processing and computational linguistics*, pages 79–91. Springer.

Umar Farooq, Hasan Mansoor, Antoine Nongaillard, Yacine Ouzrout, and Muhammad Abdul Qadir. 2017. Negation Handling in Sentiment Analysis at Sentence Level. *JCP*, 12(5):470–478.

Noura Farra, Elie Challita, Rawad Abou Assi, and Hazem Hajj. 2010. Sentence-level and document-level sentiment mining for Arabic texts. In *2010 IEEE international conference on data mining workshops*, pages 1114–1119. IEEE.

Jiashi Feng, Huan Xu, Shie Mannor, and Shuicheng Yan. 2014. Robust logistic regression and classification. In *Advances in neural information processing systems*, pages 253–261.

Charles A Ferguson. 1959. Diglossia. *word*, 15(2):325–340.

Donia Gamal, Marco Alfonse, El-Sayed M. El-Horbaty, and Abdel-Badeeh M.Salem. 2018. Opinion Mining for Arabic Dialects on Twitter. *Egyptian Computer Science Journal*, 42(4):52–61.

Aniruddha Ghosh and Tony Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 161–169.

Tobias Günther and Lenz Furrer. 2013. GU-MLT-LT: Sentiment analysis of short messages using linguistic features and stochastic gradient descent. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 328–332.

Hossam S Ibrahim, Sherif M Abdou, and Mervat Gheith. 2015. Sentiment analysis for Modern Standard Arabic and colloquial. *arXiv preprint arXiv:1505.03105*.

Jihen Karoui, Farah Banamara Zitoune, and Veronique Moriceau. 2017. Soukhria: Towards an irony detection system for Arabic in social media. *Procedia Computer Science*, 117:161–168.

Suresh Kumar and Shivani Goel. 2015. Enhancing Text Classification by Stochastic Optimization method and Support Vector Machine. *International Journal of Computer Science and Information Technologies, 6 (4)*, pages 3742–3745.

Kathrein Abu Kwaik, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnika. 2018. A Lexical Distance Study of Arabic Dialects. *Procedia computer science*, 142:2–13.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Salima Medhaffar, Fethi Bougares, Yannick Estève, and Lamia Hadrich-Belguith. 2017. Sentiment analysis of Tunisian dialects: Linguistic ressources and experiments. In *Proceedings of the third Arabic natural language processing workshop*, pages 55–61.

Ahmed Mourad and Kareem Darwish. 2013. Subjectivity and sentiment analysis of modern standard Arabic and Arabic microblogs. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 55–64.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519.

Ahmed Oussous, Ayoub Ait Lahcen, and Samir Belfkih. 2018. Improving Sentiment Analysis of Moroccan Tweets Using Ensemble Learning. In *International Conference on Big Data, Cloud and Applications*, pages 91–104. Springer.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct):2825–2830.

Chatrine Qwaider, Motaz Saad, Stergios Chatzikyriakidis, and Simon Dobnik. 2018. Shami: A corpus of levantine arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Kumar Ravi and Vadlamani Ravi. 2015. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems*, 89:14–46.

Rizkallah, Sandra and Atiya, Amir and ElDin Mahgoub, Hossam and Heragy, Momen", editor="Hassanien, Aboul Ella and Tolba, Mohamed F. and Elhoseny, Mohamed and Mostafa, Mohamed . 2018. Dialect Versus MSA Sentiment Analysis. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, pages 605–613, Cham. Springer International Publishing.

Lina Maria Rojas-Barahona. 2016. Deep learning for sentiment analysis. *Language and Linguistics Compass*, 10(12):701–719.

Hiroshi Shimodaira. 2014. Text classification using naive bayes. *Learning and Data Note*, 7:1–9.

Abu Bakr Soliman, Kareem Eissa, and Samhaa R El-Beltagy. 2017. Aravec: A set of Arabic word embedding models for use in Arabic nlp. *Procedia Computer Science*, 117:256–265.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Deep learning for sentiment analysis: successful approaches and future challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(6):292–303.

Kees Versteegh. 2014. *The Arabic language*. Edinburgh University Press.

Shuo Xu, Yan Li, and Zheng Wang. 2017. Bayesian multinomial Naïve Bayes classifier to text classification. In *Advanced multimedia and ubiquitous engineering*, pages 347–352. Springer.

Mohab Youssef and Samhaa R El-Beltagy. 2018. MoArLex: An Arabic Sentiment Lexicon Built Through Automatic Lexicon Expansion. *Procedia computer science*, 142:94–103.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.