Empirical Evaluation of Leveraging Named Entities for Arabic Sentiment Analysis

Hala Mulki[1], Hatem Haddad[2], Mourad Gridach[3], and Ismail Babaoğlu[1]
[1]Computer Engineering Department, Konya Technical University, Turkey
[2]Computer Science Department, University of Manouba, Tunisia
[3]Computational Bioscience Departments, University of Colorado Boulder, USA

Abstract: *Social media reflects the attitudes of the public towards specific events. Events are often related to persons, locations or organizations, the so-called Named Entities (NEs). This can define NEs as sentiment-bearing components. In this paper, we dive beyond NEs recognition to the exploitation of sentiment-annotated NEs in Arabic sentiment analysis. Therefore, we develop an algorithm to detect the sentiment of NEs based on the majority of attitudes towards them. This enabled tagging NEs with proper tags and, thus, including them in a sentiment analysis framework of two models: supervised and lexicon-based. Both models were applied on datasets of multi-dialectal content. The results revealed that NEs have no considerable impact on the supervised model, while employing NEs in the lexicon-based model improved the classification performance and outperformed most of the baseline systems.*

1. Introduction

Social media represents a huge source of information from which opinions can be extracted and exploited in many analytical studies. Sentiment Analysis (SA) is a Natural Language Processing (NLP) task that mines the subjective content in a piece of text and categorizes it into positive, negative or neutral polarity using computational linguistics techniques [16]. SA can be performed at three levels of granularity:

a) Document-level: where a piece of text is analyzed as a whole to produce an overall sentiment.
b) Entity-level: recognizes the sentiment of specific aspects in a piece of text.
c) Sentence-level: provides the sentiment for each sentence in the corpus. Recently, sentence-level SA has gained an increased attention to support opinion analysis of social media platforms such as Twitter and Facebook whose online textual contents (tweets/comments) are mostly shared in the form of sentences [20].

Social media often combines the opinions of the public towards all Named Entities (NE) types (persons, locations or organizations). Thus, NEs in a sentence can be considered essential components without which the subjectivity of the sentence might be lost. To clarify that, in "من لم ينتخب نداء تونس كأنه انتخب حركة النهضة" (Those who didn't vote for Nidaa Tounes, as if they voted for Ennahda Movement.) there are two NEs representing two political parties: "نداء تونس" (Nidaa Tounes) and "حركة النهضة" (Ennahda Movement); if we

omit these NEs, the subjectivity of the sentence cannot be recognized while with them retained, the tweet's polarity would not be correctly inferred unless the sentiment borne by each NE is identified. In addition, the polarity of a tweet, containing an NE and posted during a specific period of time, is affected by this very NE and the attitudes towards it at that time. For example, when exploring the dataset of [19] that was collected during the post-revolution Tunisian elections, we find that 80% of the tweets that contained the NE "بن علي" (Ben Ali), which refers to the former Tunisian president, has a negative sentiment. Similarly, in the dataset of [3], the location name "سوريا" (Syria) which has been recently related to war incidents, was encountered in 30 tweets, 75% of them was negative.

Therefore, given a Twitter/Facebook dataset collected in a certain period of time, we hypothesize that recognizing the sentiment of an NE can contribute in identifying the polarity of the sentence in which it is mentioned. NEs sentiment recognition is not trivial and has not been tackled in previous studies; where most of them focused on NEs recognition rather than NEs exploitation for further NLP tasks. Moreover, combining Named Entity Recognition (NER) with sentence-level SA poses another level of difficulty especially when Arabic language is tackled. On one hand, compared to Indo-European languages, Arabic texts has no notion of capital letters, therefore, Arabic NER systems have to recognize NEs without using capitalization among the features. On the other hand, Arabic Sentiment Analysis (ASA) is challenging especially with the existence of two Arabic language

variants: Modern Standard Arabic (MSA) and Dialectal Arabic (DA) where the latter is commonly used in social media. Both variants have a complex morphology as words are of a highly inflectional and derivational nature [6]. Thus, some adjectives and NEs might be identical. This can be seen in person names where many Arabic person names are derived from adjectives as in the positive adjective "سعيد" (happy) and can be also used as a male person name which misleads the sentiment classifier. To avoid such confusion, NEs are usually recognized then person names are eliminated while mining the sentiment [5, 7]. Finally, most of the Arabic NLP resources needed for NER or SA are not publicly available.

Here, we present an empirical evaluation of the effectiveness of NEs in inferring the sentiment of MSA/DA tweets and Facebook comments. To the extent of our knowledge, this is the first effort to pair NEs with ASA. While previous ASA works ignored NEs or eliminated some NE types, we investigate using NEs as expressive features to be included in ASA framework of two model variants: supervised and lexicon-based. This is done by classifying the NEs extracted via the NER system into positive or negative. The sentiment-annotated NEs are, then, replaced with special tags in the corpus. The proposed framework was applied on four datasets of MSA/DA content. We conducted SA once with NEs tagged and included among the features then with them considered as ordinary tokens. This enabled answering these research questions:

- What is the impact of including NEs on ASA models: lexicon-based and supervised?
- For datasets rich of NEs, is it more likely to have a better SA performance?
- Are NEs reliable enough to infer the Arabic sentiment? For which SA models?

2. Arabic Sentiment Analysis

Arabic SA methods can be categorized under two main categories: machine learning and lexicon-based.

2.1. Machine Learning Methods

These methods adopt supervised/unsupervised learning strategies using either hand-crafted features or distributed text representations. The training process depends on learning that a combination of specific features yields a certain polarity [16].

Among the ASA systems that employed hand-crafted-features, we can refer to [1] where bag-of-words along with several levels of stemming were used to train a supervised sentiment classifier of MSA/Jordanian tweets. The best algorithm was SVM with an accuracy of 87.2%.

In the same context, the authors in [19] presented a supervised SA system for Tunisian tweets. With

different bag-of-word schemes used as features, binary and multiclass classifications were conducted. SVM was found of the best results for binary classification with an accuracy of 71.09% and an F-measure of 63%.

A novel SA model based on text embeddings was proposed by [3]. The model was trained with Arabic word embeddings generated via word2vec [11] and applied on MSA/DA datasets. Nu-SVM was the best-performing classifier with an accuracy of 80.21% and an F-measure of 79.62% for the twitter dataset.

Medhaffar *et al.* [12], doc2vec algorithm [11] was used to produce document embeddings of Tunisian comments. The generated embeddings were fed to train a Multi-Layer Perceptron (MLP) classifier where both the achieved accuracy and F-measure values were 78%.

2.2. Lexicon-based Methods

The core components of such models are manually-built, predefined or translated sentiment lexicons. A sentiment lexicon contains subjective words along with their polarities (positive or negative) and polarity scores also known as weights [16]. Thus, the polarity of a word or a sentence can be determined by one of the following algorithms [5]:

- *Straight Forward Sum (SFS)*: adopts the uniform weighting scheme, where negative words have the weight of -1 while positive ones have the weight of 1. The polarity of a given text is calculated by accumulating the weights of negative and positive terms. The sign of the resulting sum infers a positive sentiment if it is positive and a negative one if it is negative.
- *Double Polarity (DP)*: Assigns both a positive and a negative weight for each entry in the lexicon. For example, if a positive term has a score of 0.7, then its negative score: (1+0.7) = -0.3 and vice versa for negative terms. To define the polarity of a sentence, two scores are accumulated: the positive and negative where the one of the greater absolute value defines the total sentiment.

Abdulla *et al.* [2], manually-built lexicons were compared against automatically-built ones for the SA task. Each lexicon type was expanded by adding synonyms, stemming and most common words via Term Frequency (TF) weighting, emotions and dialectal terms. A dataset from [1] was used in the experiments. The study showed that stemming degraded the performance for both lexicon types. The merged lexicon with light stemming achieved the best accuracy equals to 74.6%.

Karmani [10], a lexicon-based SA system was used to classify the sentiment of Tunisian tweets. The author developed a Tunisian morphological analyzer to produce linguistic features. Using a Tunisian version

developed in [9], the model achieved an accuracy of 72.1% considering only the positive/negative tweets.

3. Arabic Named Entity Recognition

NER approaches fall into three categories: rule-based, Machine Learning (ML) and hybrid of both. For the English language, the state-of-the-art research is dominated by ML approaches. Systems that employ deep neural networks as the main building blocks have recently become the dominant methods among ML approaches. These systems combine various Neural Network (NN) architectures with other ML methods to handle NEs. They rely on Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs) and Conditional Random Fields (CRF). In addition, [4] indicated that NN-based approaches do not rely on hand-crafted features as they use pre-trained word embeddings to initialize the word vectors and character-level embeddings.

For the Arabic language, [8] developed an Arabic NER for social media based on deep neural networks. This system achieved the state-of-the-art on the Twitter dataset of [21], where it scored an F-measure of 85.71%. In our study, we employed this system to recognize Arabic NEs in the input data.

4. Methodology

In this study, we present a SA framework Tw-StAR which mines the sentiment with the assistance of NEs using supervised and lexicon-based models developed in [14]. NEs were extracted by the system of [8], then, they were fed to our NEs sentiment detection algorithm to associate each NE with its relevant polarity. Later, the sentiment-annotated NEs were included in Tw-StAR to assist in the SA task. For both model variants, we adopted n-gram schemes such that they were fed to train the classifiers of the supervised model, while they were looked up in the sentiment lexicon adopted by the lexicon-based model. The pipeline of the proposed framework is illustrated in Figure 1.
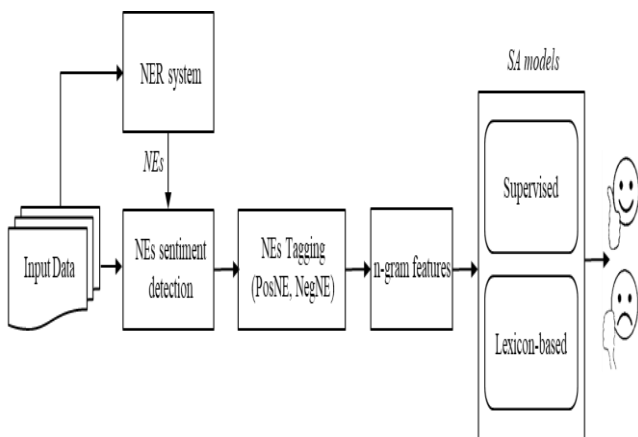

Figure 1. Tw-StAR sentiment analysis framework.

4.1. Arabic NER System

To extract Arabic NEs, we used the system of [8]. This system can handle NEs encountered in social media data. Figure 2 shows its main architecture while recognizing the NEs of the sentence "يفوز مدريد ريال بالدوري الاسباني" (Real Madrid wins the Spanish league).
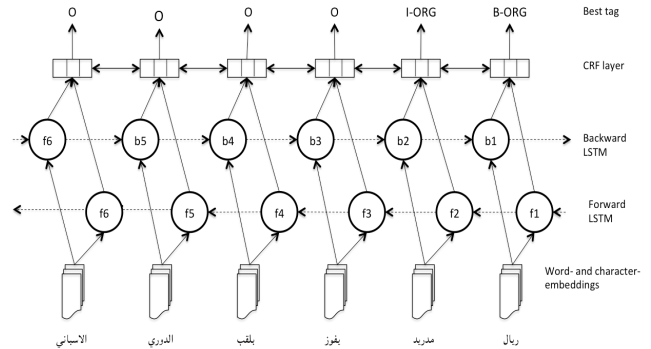

Figure 2. The architecture of the used Arabic NER system.

4.2. NEs Sentiment Detection Algorithm

To include NEs in the SA task and inspired by SFS method [5], we have developed an algorithm (Algorithm 1) to detect the sentiment of the NEs extracted by the system of [8].

Algorithm 1: NEs Sentiment Detection Algorithm

T=Tweet tokens
N=Named Entity
Pol_t=Tweet Polarity
N_score=NE Sentiment Score
N_score=0
#Browse all NEs in the corpus
For each (N in NEs)
*   For each (tweet in corpus)*
*   {*
*   If (N in T) then*
*   {*
*   If (Pol_t=positive) then*
*   Increase N-score by 1*
*   Else*
*   Decrease N-score by 1*
*   }*
*   }*
N-pol= NEs polarity
If (N_score>0) then
*   N_pol=positive*
Else
*   N_pol=negative*
return N_Pol

In Algorithm 1, the polarity of an NE in a corpus is defined by the majority of attitudes towards it. In other words, the sentiment of each NE in the dataset is identified as positive or negative according to how frequently this NE is mentioned within positive or negative tweets. This can address the confusion of detecting the sentiment of two NEs that have contradict polarities and mentioned in the same tweet as in "إنه خالد بن الوليد القائد النبيل يا من يفتخر بـ هتلر" (To those who boast of Hitler, it is Khalid Ibn Al-Walid, the noble

leader) where "هتلر" (Hitler) who was known as a dictator and "خالد بن الوليد" (Khalid Ibn Al-Waleed) who was a commander, were mentioned together in a positive tweet. In this case, the algorithm gives both NEs a positive score at the beginning, however, after browsing the rest of tweets the score of "هتلر" will decrease if it is mostly mentioned in negative contexts while the score related to "خالد بن الوليد" will increase if the majority of the tweets containing it are positive.

4.3. Sentiment Analysis Using Tw-StAR

The following steps describe the general SA pipeline adopted for both models:

- *Preprocessing*: the input data was normalized such that URLs, tweet-related symbols, punctuation and non-Arabic characters were removed, while stopwords and negations were kept to capture the sentiment borne by compound terms. So, a tweet such as "هاري بوتر ريحة الغالي!#" becomes after preprocessing: "ريحة الغالي هاري بوتر".
- *NEs tagging*: after NEs are extracted from the training corpus and their polarities are detected, each NE was replaced with a textual tag indicating either a positive NE (PosNE) or a negative one (NegNE). Thus, these tags are considered as corpus tokens and will be included among the generated n-gram features representing the training and test divisions of each dataset.
- *Feature extraction*: with tokenization applied at the word-level, several n-gram schemes were generated as features for both models. For the supervised model, each input sentence is represented by unigrams (single words), bigrams (word pairs), trigrams (word triples) and combinations of them were adopted, as they can capture information about the local word order and save the training time. However, in the lexicon-based model, unigrams and a combination of unigrams and bigrams were used to cover single and compound phrases of the used lexicon [6].
- *Sentiment classification*: supervised algorithms were employed by the supervised model, while DA lexicons along with the scoring algorithm SFS [5] were used to detect the sentiment in the lexicon-based model.

4.4. Tw-StAR Supervised Model

In this model, the feature vector of each sentence is constructed via examining the presence/absence of the n-gram schemes among the sentence's tokens. The produced n-gram schemes include trigrams in addition to unigrams and bigrams since higher-order n-grams can better capture the contextual information [18]. Later, feature selection was conducted using the Term Frequency (TF) weighting by FreqDist module. The

supervised model was trained using NB from Scikit-Learn and linear SVM from LIBSVM.

Having all the NEs recognized, identified as having positive or negative polarity and tagged properly, they were involved in inferring the sentiment. This is because the tags of NEs are included among the tweets' n-grams from which the feature vectors are constructed. Figure 3 shows the pipeline of this model.
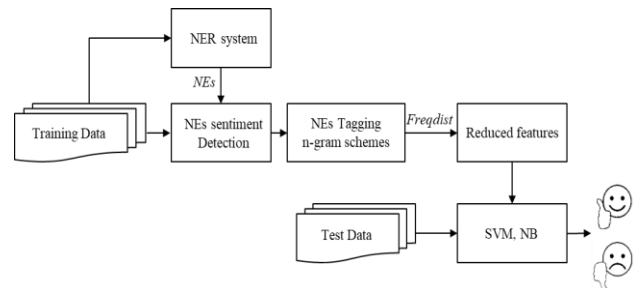

Figure 3. Tw-StAR supervised model with NEs included.

4.5. Tw-StAR Lexicon-based Model

This model uses an integrated lexicon shown in Table 1.

Table 1. The used sentiment lexicons.

Sentiment Lexicon	Positive	Negative	Size
NileULex	1697	4256	5953
AEL	12	11	23
AHL	107	118	225
LevLex	258	559	817
GulfLex	33	67	100
TunLex	1953	3329	5282

This lexicon is composed of pre-built lexicons: MSA/Egyptian NileULex [5], MSA/DA seeds from Arabic Emotion Lexicon (AEL) and Arabic Hashtag Lexicon seeds (AHL) [17, 18] plus two manually-built lexicons for Levantine (LevLex) and Gulf (GulfLex) dialects. For the Tunisian datasets, we built a Tunisian lexicon (TunLex). Table 1 lists these lexicons and their sizes.

To recognize the sentiment of the input data, the tokens of a sentence either unigrams or the combination of unigrams and bigrams are looked up in the proper lexicon. When a match is found, the sentence's polarity score is calculated using SFS algorithm. The textual tags of NEs were included in the n-gram features to be looked up in the lexicon. Consequently, both NEs tags: PosNE and NegNE were added to the lexicon as positive and negative entries having the scores of 1 and 0, respectively. Figure 4 shows the pipeline of the lexicon-based model.
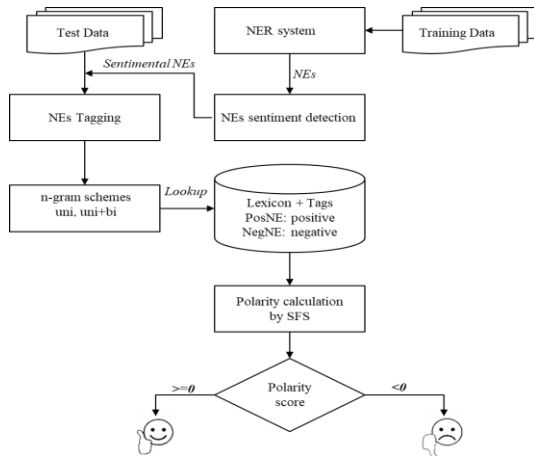
Figure 4. Tw-StAR lexicon-based model with NEs included.

5. Experiments and Results Evaluation

5.1. Datasets

Tw-StAR was evaluated using the datasets in Table 2.

Table 2. Polarity distribution in the training and test sets.

Dataset	Training		Test		Total size
	Positive	Negative	Positive	Negative	
JEG	1732	1702	415	445	4294
TAC	306	290	76	74	746
TEC	968	1466	276	333	3043

The specifications of these datasets are as follows:

1. Jordanian Egyptian Gulf (JEG): was tackled in [3], it combines positive/negative tweets of the dialects:

   a) Jordanian: ArTwitter [1].
   b) Egyptian: ASTD [15].
   c) Gulf: QCRI [13].

2. Tunisian Arabic Corpus (TAC): consists of tweets about media, telecom and politics. It was collected by [10] and annotated for positive, negative and neutral polarity.
3. Tunisian Election Corpus (TEC): refers to a set of MSA/Tunisian tweets collected by [19] during the Tunisian elections in 2014.
4. Tunisian Sentiment Analysis Corpus (TSAC): collected by [12], it combines positive and negative Facebook comments about TV shows.

Each dataset was divided into training and test sets as it is shown in Table 2.

5.2. Named Entities Results

The statistics of the extracted NEs using the NER system [8] along with the number of positive/negative NEs defined by the proposed NEs sentiment detection algorithm, are listed in Table 3. E-NEs, Pos-NEs, Neg-NEs and A-NEs denote the number of the extracted NEs, positive NEs, negative NEs and the sentiment-annotated NEs, respectively.

Table 3. NEs statistics extracted from each dataset.

Dataset	E-NEs	Pos-NEs	Neg-NEs	A-NEs
JEG	841	395	487	782
TAC	240	99	129	228
TEC	658	192	410	602
TSAC	615	198	350	548

We notice that from large-sized datasets such as TSAC, JEG and TEC, more NEs could be extracted. In addition, in all datasets, the number of negative NEs is greater than that of the positive ones.

On the other hand, although the Tunisian datasets: TSAC and TEC have a larger or close size compared to JEG; yet the used NER system extracted less NEs from them compared to those extracted from JEG. This is due to the fact that the used NER system employed pre-trained word embeddings from [21] produced with corpora composed of MSA, Egyptian and Levantine content which is quite similar to that of JEG while it is far from the Tunisian dialect. This made most of the Tunisian terms unknown when they looked up in the lookup table of the NER system; therefore they will be initialized randomly instead of being initialized with pre-trained word embeddings.

5.3. Sentiment Analysis Results

5.3.1. Supervised Model Results

The supervised model was trained once without tagging NEs (Tw-StAR) then with NEs tagged and included in the features (Tw-StAR+NEs). Three experiment variants were conducted, where the first involved using all n-gram features, while the second and third used a reduced number of features obtained by the TF scheme for the thresholds: 2 and 3, respectively.

Table 4. Supervised Tw-StAR performance for all datasets.

Dataset	NEs	Algorithm	Prec (%)	Rec (%)	F1 (%)	Acc (%)
JEG	No	NB	**77.0**	**77.0**	**76.9**	**76.9**
		SVM	71.6	71.2	71.2	71.4
	Yes	NB	**76.8**	**76.8**	**76.7**	**76.7**
		SVM	69.9	69.8	69.8	69.9
TAC	No	NB	83.4	81.9	81.8	82.0
		SVM	**85.2**	**84.6**	**84.6**	**84.7**
	Yes	NB	**84.4**	83.2	83.2	83.3
		SVM	83.4	**83.3**	**83.3**	**83.3**
TEC	No	NB	71.8	68.8	68.7	70.4
		SVM	**75.0**	**71.4**	**71.4**	**73.1**
	Yes	NB	72.3	69.6	69.6	71.1
		SVM	**74.4**	**71.2**	**71.2**	**72.7**
TSAC	No	NB	91.2	92.0	91.4	91.4
		SVM	**92.8**	**92.5**	**92.7**	**92.8**
	Yes	NB	91.6	**92.4**	91.7	91.7
		SVM	**92.4**	92.2	**92.3**	**92.4**

We chose to review the results of the experiment of the best averaged F-measure, with/without NEs. Table 4 lists this model's results where uni, bi and tri refer to unigrams, bigrams and trigrams, respectively. While Prec, Rec, F1 and Acc indicate the averaged precision,

recall, F-measure and accuracy, respectively. A comparison with baseline systems is shown in Table 5.

Table 5. Supervised Tw-StAR against baselines.

Dataset	Model	Prec (%)	Rec (%)	F1 (%)	Acc (%)
JEG	word2vec+supervised	76.5	83.0	79.6	80.2
	Tw-StAR	77.0	77.0	76.9	76.9
	Tw-StAR + NEs	76.8	76.8	76.7	76.7
TAC	lexicon-based	63.0	72.9	67.3	72.1
	Tw-StAR	85.2	84.6	84.6	84.7
	Tw-StAR + NEs	83.4	83.3	83.3	83.3
TEC	supervised + n-grams	67.0	71.0	63.0	71.1
	Tw-StAR	75.0	71.4	71.4	73.1
	Tw-StAR + NEs	74.4	71.2	71.2	72.7
TSAC	Doc2vec + MLP	78.0	78.0	78.0	78.0
	Tw-StAR	92.8	92.5	92.7	92.8
	Tw-StAR + NEs	92.4	92.2	92.3	92.4

5.3.2. Lexicon-based Model Results

In this model, Tw-StAR experiments were conducted with two lexicons:

a) An integrated lexicon constructed out of NileULex, AEL, AHL, LevLex and GulfLex (see Table 1) to handle JEG dataset whose content combines Egyptian, Levantine and gulf dialects in addition to MSA.

b) TunLex to mine the sentiment of TAC, TEC and TSAC datasets. In Tw-StAR+NEs experiments, however, the previous lexicons were used but with NEs tags: PosNE, NegNE added as entries having positive and negative scores, respectively. The sentiment detection procedure was carried out by looking for a tweet's unigrams (uni) then unigrams and bigrams (uni+bi) in the relevant lexicon, once with NEs tagged then with them treated as ordinary tokens.

Table 6. Lexicon-based Tw-StAR performance for all datasets.

Dataset	NEs	Features	Prec (%)	Rec (%)	F1 (%)	Acc (%)
JEG	No	uni+bi	**71.6**	67.4	66.3	68.3
	Yes	uni+bi	70.7	**69.2**	**68.9**	**69.7**
TAC	No	uni+bi	66.9	66.7	66.6	66.7
	Yes	uni+bi	**70.8**	**70.6**	**70.6**	**70.7**
TEC	No	uni+bi	66.6	61.5	59.8	64.0
	Yes	uni+bi	**69.1**	**65.6**	**65.0**	**67.5**
TSAC	No	uni+bi	84.5	83.8	81.8	81.8
	Yes	uni+bi	**84.6**	**84.7**	**82.8**	**82.8**

Table 6 shows the best results of the lexicon-based model, as unigram+bigram features scored the best results with/without NEs. These performances were, further, compared against baseline systems in Table 7.

Table 7. Lexicon-based Tw-StAR performance against baselines.

Dataset	Model	Prec (%)	Rec (%)	F1 (%)	Acc (%)
JEG	word2vec+supervised	76.5	83.0	79.6	80.2
	Tw-StAR	71.6	67.4	66.3	68.3
	Tw-StAR + NEs	70.7	69.2	68.9	69.7
TAC	lexicon-based	63.0	72.9	67.3	72.1
	Tw-StAR	66.9	66.7	66.6	66.7
	Tw-StAR + NEs	70.8	70.6	70.6	70.7
TEC	supervised + n-grams	67.0	71.0	63.0	71.1
	Tw-StAR	66.6	61.5	59.8	64.0
	Tw-StAR + NEs	69.1	65.6	65.0	67.5
TSAC	Doc2vec + MLP	78.0	78.0	78.0	78.0
	Tw-StAR	84.5	83.8	81.8	81.8
	Tw-StAR + NEs	84.6	84.7	82.8	82.8

6. Results Discussion

When exploring the performances of the supervised model in Table 4, we notice that although tagging NEs in the training corpus was expected to enhance the performance as it decreases the features' size by reducing all the NEs to either PosNE or NegNE terms, a degraded performance could be noticed. While comparable results were scored with and without NEs in JEG, TEC and TSAC datasets, the performance degraded when NEs were added in TAC as the F-measure decreased by 1.3%. This could be due to the fact that inferring the sentiment using n-gram schemes depends on capturing the contextual information with which a specific n-gram scheme is learned to be an indicator of a specific sentiment. As the sentiment of an NE was deduced based only on how frequent it is mentioned within a context of a positive or a negative polarity regardless of the context's words, it is possible for a positive NE to be included within a negative context (n-gram scheme) and vice versa which misleads the classifier.

Unlike the supervised model, the performance of the lexicon-based model was favorably impacted by involving NEs in the SA task. As it can be seen in Table 6, for uni+bi features, the sentiment classification performance with NEs considered and NE tags added to the lexicons could outperform the one obtained by the ordinary lexicons. Indeed, the evaluation measures increased in all datasets as the F-measure values of Tw-StAR+NEs were 68.9%, 70.6%, 65% and 82.8% compared to 66.3%, 66.6%, 59.8% and 81.8% achieved by Tw-StAR for JEG, TAC, TEC and TSAC datasets, respectively. The reason behind such improvement is that uniform weight scheme lexicons ignore the contextual-related information where a sentence's polarity is defined based on the polarity scores of its constituent words [5, 7]. This in turn enables the sentiment-annotated NEs deduced regardless of the context, to effectively contribute in recognizing the polarity of the tweet containing it. Moreover, with NEs tagged in the test corpus, it became possible to employ NEs of the type person names in the SA task. Hence, the issue caused by confusing a person name with an adjective [5, 7] could be avoided without the need to eliminate person names.

Considering Table 7 which compares the lexicon-based model against the baseline systems, it should be noted that this comparison is meaningful only for TAC dataset where the baseline system [10] is a lexicon-based one; though we observed that Tw-StAR+NEs outperformed the baselines in Tunisian datasets: TAC, TEC and TSAC. This could be explained by the positive impact of NEs on the polarity detection in addition to the good coverage provided by the used Tunisian lexicon. In contrast, it is reasonable that the performance degraded in JEG dataset as the F-measure

decreased by 10.7% compared to [3] that used pre-trained word embeddings. In addition, the efficiency of Tw-StAR with and without NEs can be attributed to the looking up for uni+bi tokens in the lexicon increases the hits ratios of compound terms.

Finally, for datasets rich of NEs (see Table 3); we could not determine the impact of the number of the sentiment-annotated NEs on SA within Tw-StAR+NE lexicon-based model. To clarify that, although JEG has the greatest number of sentiment-annotated NEs, the improvement recorded in the F-measure value was 2.6%, while for TEC that has less annotated NEs, the F-measure increased by 5.2%. We believe that the performance of the lexicon-based model for a specific corpus, with NEs included, is not related to the number of the sentiment-annotated NEs in the corpus as much as it is to the consistency of that corpus. More specifically, in a corpus having a good degree of consistency, the training and test data tend to contain more similar NEs. Thus, it is more likely to have a consensus on the sentiment of a specific NE which leads to an accurate sentiment assignment of that NE and hence to a better sentiment classification.

7. Conclusions and Future Work

We presented a pioneering step towards leveraging NEs in Arabic sentiment analysis. It was observed that NEs can form reliable indicators of Arabic sentiment if they are included within the lexicon-based model of Tw-StAR framework, while a similar behavior could not be noticed in the supervised model. In addition, adding NEs to the lexicon-based model enabled the exploitation of person names in the SA task. On the other hand, it was revealed that the impact of the number of sentiment-annotated NEs is less important than the consistency of the tackled corpora as the latter affects the sentiment classification performance. For the future work, we will investigate the impact of NEs on multiclass SA. Furthermore, as NEs performed better in the lexicon-based model, it would be interesting to develop a sentiment recognition method that considers the negation and sarcasm. Regarding the NER task, it could be enhanced if the different writing styles of NEs are handled. Lastly, Tw-StAR framework would be further examined on datasets of other languages such as English, French and Turkish.

References

[1] Abdulla N., Ahmed N., Shehab M., and Al-Ayyoub M., "Arabic Sentiment Analysis: Lexicon-Based and Corpus-Based," *in Proceedings of IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies*, Amman, pp. 1-6, 2013.

[2] Abdulla N., Ahmed N., Shehab M., and Al-Ayyoub M., "Towards Improving The Lexicon-Based Approach for Arabic Sentiment Analysis," *International Journal of Information Technology and Web Engineering*, vol. 9, no. 3, pp. 55-71, 2014.

[3] Altowayan A. and Tao L., "Word Embeddings for Arabic Sentiment Analysis," *in Proceedings of IEEE International Conference on Big Data*, Washington, pp. 3820-3825, 2016.

[4] Darwish K., "Named Entity Recognition Using Cross-Lingual Resources: Arabic As An Example," *in Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, pp. 1558-1567, 2013.

[5] El-Beltagy S. and Ali A. "Open Issues in the Sentiment Analysis of Arabic Social Media: A Case Study," *in Proceedings of 9th IEEE International Conference on Innovations in Information Technology*, Abu Dhabi, pp. 215-220, 2013.

[6] El-Beltagy S., Khalil T., Halaby A., and Hammad M., "Combining Lexical Features and a Supervised Learning Approach for Arabic Sentiment Analysis," *in Proceedings of International Conference on Intelligent Text Processing and Computational Linguistics*, Konya, pp. 307-319, 2016.

[7] El-Makky N., Nagi K., El-Ebshihy A., Apady E., Hafez O., Mostafa S., and Ibrahim S., "Sentiment Analysis of Colloquial Arabic Tweets," *in Proceedings of the 3rd ASE International Conference on Social Informatics*, Cambridge, pp. 1-9, 2013.

[8] Gridach M., "Character-Aware Neural Networks for Arabic Named Entity Recognition for Social Media," *in Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing*, Osaka, pp. 23-32, 2016.

[9] Karmani N., Soussou H., and Alimi A., "Tunisian Arabic Chat Alphabet Transliteration Using Probabilistic Finite State Transducers," *The International Arab Journal of Information Technology*, vol. 16, no. 2, pp. 295-303, 2019.

[10] Karmani N., Tunisian Arabic Customer's Reviews Processing and Analysis for an Internet Supervision System, Thesis, Sfax University, 2017.

[11] Le Q. and Mikolov T., "Distributed Representations of Sentences and Documents," *in Proceedings of International Conference on Machine Learning*, Beijing, pp. 1188-1196, 2014.

[12] Medhaffar S., Bougares F., Estève Y., and Hadrich-Belguith L., "Sentiment Analysis of Tunisian Dialects: Linguistic Resources and Experiments," *in Proceedings of the 3rd Arabic Natural Language Processing Workshop*, Valencia, pp. 55-61, 2017.

[13] Mourad A. and Darwish K., "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*, Atlanta, pp. 55-64. 2013.

[14] Mulki H., Haddad H., Gridach M., and Babaoğlu I., "Tw-StAR at SemEval-2017 Task 4: Sentiment classification of Arabic Tweets," *in Proceedings of the 11th International Workshop on Semantic Evaluation*, At Vancouver, pp. 664-669, 2017.

[15] Nabil M., Aly M., and Atiya A., "Astd: Arabic Sentiment Tweets Dataset," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Lisbon, pp. 2515-2519, 2015.

[16] Piryani R., Madhavi D., and Singh V., "Analytical Mapping of Opinion Mining and Sentiment Analysis Research During 2000-2015," *Information Processing and Management*, vol. 53, no. 1, pp. 122-150, 2017.

[17] Salameh M., Mohammad S., and Kiritchenko S., "Sentiment after Translation: A Case-Study on Arabic Social Media Posts," *in Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, pp. 767-777, 2015.

[18] Salameh M., Mohammad S., and Kiritchenko S., "How Translation Alters Sentiment," *Journal of Artificial Intelligence Research*, vol. 55, pp. 95-130, 2016.

[19] Sayadi K., Liwicki M., Ingold R., and Bui M., "Tunisian Dialect and Modern Standard Arabic Dataset for Sentiment Analysis: Tunisian Election Context," *To appear in IEEE Proceedings of ACLing*, 2016.

[20] Yang B. and Cardie C., "Context-Aware Learning for Sentence-Level Sentiment Analysis With Posterior Regularization," *in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, Baltimore, pp. 325-335, 2014.

[21] Zahran M., Magooda A., Mahgoub A., Raafat H., Rashwan M., and Atyia A., "Word Representations in Vector Space and Their Applications for Arabic," *in Proceedings of International Conference on Intelligent Text Processing and Computational Linguistics*, Cairo, pp. 430-443, 2015.

Hala Mulki is a Ph.D. candidate at Computer Engineering Department, Konya Technical University, Turkey. She received her B.S. (2003) and MSc. (2009) from Aleppo University, Syria. Her research interests include Machine Learning and NLP.

Hatem Haddad is Assistant Professor at Manouba University, Tunisia. He received his Ph.D. (2002) from University Grenoble Alpes, France. His current research interests include Machine Learning, NLP and Deep Learning.
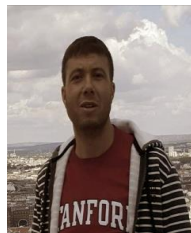
Mourad Gridach is a Postdoctoral Researcher at the University of Colorado, USA. He is a Professor at Ibn Zohr University, Morocco. His research interests include Artificial Intelligence, Computer Vision and NLP.

Ismail Babaoğlu is Associate Professor at Department of Computer Engineering, Konya Technical University, Turkey. He received his Ph.D. (2010) from Selçuk University, Turkey. His research interests include swarm intelligence, and evolutionary algorithms.