ONE: Toward ONE model, ONE algorithm, ONE corpus dedicated to sentiment analysis of Arabic/Arabizi and its dialects

Imane Guellil[1,2,3], Faical Azouaou[3], Fodil Benali[4],Ala-Eddine Hachani[5]

1 SEA Research group, Aston university / Birmingham, UK
2 Microports.net Ltd / Birmingham, UK
3 Laboratoire des Méthodes de Conception des Systèmes (LMCS),
Ecole nationale Supérieure d'Informatique,BP 68M, 16309, Oued-Smar, Alger, Algérie
4 Capgemini / Paris, France
5 Legal doctrine / Algiers, Algeria
i.guellil@aston.ac.uk

Abstract

Arabic is the official language of 22 countries, spoken by more than 400 million speakers. Each one of this country uses at least on dialect for daily life conversation. Then, Arabic has at least 22 dialects. Each dialect can be written in Arabic or Arabizi Scripts. The most recent researches focus on constructing a language model and a training corpus for each dialect, in each script. Following this technique means constructing 46 different resources (by including the Modern Standard Arabic, MSA) for handling only one language. In this paper, we extract ONE corpus, and we propose ONE algorithm to automatically construct ONE training corpus using ONE classification model architecture for sentiment analysis MSA and different dialects. After manually reviewing the training corpus, the obtained results outperform all the research literature results for the targeted test corpora.

1 Introduction

All the survey works in the literature (Habash, 2010; Farghaly and Shaalan, 2009; Harrat et al., 2017) classify Arabic in three main varieties: 1) Classical Arabic (CA), 2) Modern Standard Arabic (MSA) and 3) Dialectal Arabic (Boudad et al., 2017). Arabic Dialects are another form of Arabic used in daily life communication. Each dialect shares many features with MSA, but they globally differ in some aspects. Arabic and its dialects can be written either in Arabic Script or in Arabizi one. Arabizi is a form of writing Arabic text that relies on Latin letters, numerals and punctuation rather than Arabic letters (Guellil et al., 2019a,b). For example, the Arabic sentence: رَاني فرحَانة, meaning "I am happy," is written in Arabizi as "rani fer7ana". Arabizi is generally used by Arab speakers in social media or chat and SMS applications. Almost all the work on Arabic sentiment analysis focus on constructing new resources (new lexicons(Abdul-Mageed and Diab, 2012; Mataoui et al., 2016; Mohammad et al., 2016a; Gilbert et al., 2018), new training corpora(Aly and Atiya, 2013; ElSahar and El-Beltagy, 2015; Mourad and Darwish, 2013; Rahab et al., 2019; Alahmary et al., 2019; Al-Twairesh et al., 2017), new language models(Baly et al., 2020)) for each dialect. More recently, particular attention has been given to Arabizi as well(Baert et al., 2020). However, constructing a unique resource for each dialect is time and effort consuming. Moreover, this resource will be exploitable only for the targeted dialect.

This paper proposes a general algorithm constructing a language model from a large corpus and a training corpus automatically to bridge the gap. It also proposes the transliteration of the Arabizi messages into Arabic. This approach was applied to Algerian dialect (a Maghrebi dialect), having a lack of resources. However, the constructed model was used for classifying the sentiment of messages written in MSA, Tunisian dialect or even Egyptian dialect. The results were very encouraging. However, the manual review of a small part of the training corpus constructed automatically leads to outperform all the research literature results for the testing corpora cited above.

Proceedings of the 11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pages 236–249
April 19, 2021. ©2021 Association for Computational Linguistics

2 The research works inspiring the proposed work

The aim of the proposed model is to analyse the sentiment of Arabic message (written with both Arabic/Arabizi scripts). In this context, we need to focus on three categories of works: 1) Works on Arabizi transliteration. 2) Works on lexicon-based approach. 3) Works on corpus-based approach. In the following sections, we present the set of strengths/weaknesses of the research works inspiring our proposed approach.

2.1 The research works inspiring our transliteration approach

The transliteration approach is firstly inspired by the work presented by van et al. (van der Wees et al., 2016), where the authors used a table extracted from Wikipedia[1] for the passage from Arabizi to Arabic. We also present a passage table from Arabizi to Arabic. However, we also use a set of passage rules for handling the position of letters and some missed cases in the literature studied approaches. The proposed approach is also inspired by the works presented in (Al-Badrashiny et al., 2014; Darwish, 2013; May et al., 2014; van der Wees et al., 2016). All these authors generate a set of possible candidates for the transliteration of an Arabizi word into Arabic. The major issue of these approaches is the omission of some candidates because the vowels are not properly handled. Finally, This work is also inspired by the proposed approach in (Darwish, 2014; van der Wees et al., 2016) using a language model to determine the best possible candidate for a word in Arabizi. On the other hand, these works assimilate the task of transliteration to a translation task. The major drawback of these approaches is that they depend on a parallel corpus. The used corpus is usually constructed manually.

2.2 The research works inspiring our lexicon-based approach

The proposed sentiment lexicon construction approach is firstly inspired by the group of approaches using the automatic translation of an existing English lexicon (Mohammad et al., 2016a; Salameh et al., 2015; Mohammad et al., 2016b; Abdul-Mageed and Diab, 2012; Abdulla et al., 2014). The majority of these approaches are based on Google translate. However, Google translate deals with MSA only (i.e. Google translate is not adequate for translating dialects). Moreover, the Arabic/English dictionaries are covering MSA and some dialects such as Egyptian and Levantine. Limited resources are dedicated to Maghrebi dialects such as Tunisian, Moroccan or Algerian dialects. Hence, we opt to use Glosbe API [2], which is an online API offering the translation from/to MSA and almost all dialects. This API is open-source (i.e. no fees are required for using it). The proposed approach is also inspired by the work using a semi-automatic construction (El-Beltagy, 2016) where the authors manually review the constructed lexicon.

For handling morphological aspects of Arabic dialects, some works relying on stemming tools, dedicated to MSA. For example, the work of Mataoui et al. (Mataoui et al., 2016) used the Khoja stemmer (Khoja and Garside, 1999) for stemming the DALG, which is designed for MSA. In our work, we treat agglutination by employing an algorithm that supports the originality of the studied dialect (DALG), principally related to its prefixes, suffixes, and negative pronouns. The work of Al-Twairesh et al. (Al-Twairesh et al., 2017) also inspires the proposed approach. This work is relying on sentiments words for automatically annotating large corpus in Saudi dialects. However, in contrast to this work, our approach is not only concentrating on sentiment words, but it is also based on a sentiment algorithm for handling opposition, Arabic morphology and negation.

2.3 The research works inspiring our corpus-based approach

The works that firstly inspire our proposed (Pak and Paroubek, 2010; Hogenboom et al., 2013; Yadav and Pandya, 2017) are not dedicated to Arabic but other languages (English and Dutch) The main idea of these works is to use emoticons for automatically tag a large corpus. Hence, the proposed contribution also exploits the presence of emoticons to determine the sentiment of messages. However, it can be

[1] https://en.wikipedia.org/wiki/Arabic_chat_alphabet

[2] https://en.glosbe.com/

seen that all emoticons are not appropriate for determining sentiment. Hence, our proposed approach also considers emoticons for annotation but not all emoticons, only the emoticons with the strongest sentiment (either positive or negative). Our approach of constructing corpus is also inspired by the work of Gamal et al. (Gamal et al., 2019) that they relied on a sentiment lexicon to automatically annotate a sentiment corpus. However, their algorithm relies only on the positive and negative words count. For these authors, if the number of positives words is higher than or equal twice the number of negatives words than the message is considered as positive, and the same philosophy is applied for the negative messages. In contrast to these authors, we developed more sophisticated algorithms handling Arabic agglutination, opposition and negation. We also consider a set of heuristics, including the number of words.

Our contribution is also inspired by the work of Medhaffar et al. (Medhaffar et al., 2017), which is the unique work, to the best of our knowledge, focusing on Arabic and Arabizi at the same time. However, in contrast to this work, we used a more voluminous corpus (which was constructed automatically), and we propose a transliteration step. Finally, our contribution is also inspired by the approach proposed by Duwairi et al. (Duwairi et al., 2016). Hence, we firstly define and apply a transliteration step. However, in contrast to this work, our contribution is dealing with ambiguities treatment (especially vowels ambiguities), and our corpus sentiment is constructed automatically, so it is more voluminous than the corpora which the authors constructed manually.

3 Methodology

3.1 The proposed algorithm

The general algorithm proposed and developed in the context of this work is presented in Algorithm 1.

It can be seen from Algorithm 1 that the proposed steps are executed in the following order :

1. The first step is to manually construct some resources including the list of the identifiers of some famous Algerian pages, the list of positive/negative emoticons and expressions, the list of prefixes/suffixes and finally the list of negation/ opposition terms. This step is illustrated by the function MANUALRESCONSTRUCTION().
2. The second step is to automatically extract comments from Facebook pages (using the collected identifiers). This step is illustrated by the function COMMENTSEXTRACTION($Facebook_{key}$).
3. The third step is to automatically construct the Algerian sentiment lexicon by relying on an existing English sentiment lexicon. This step is illustrated by the function AUTOMATICARLEXCONSTRUCT(Eng_{lex}).
4. The fourth step is to review the constructed lexicon manually. This step is illustrated by the function MANUALLEXREVIEW(Alg_{lexV1}).
5. The fifth step is to automatically annotate each message from the corpus (extracted from Facebook). This step is illustrated by the function ANNOTATE($Alg_{lexV2}, m, L_{emp}, L_{emn}, L_{exp}, L_{exn}, L_{pr}, L_{sf}, L_{neg}, L_{op}, pos, neg$).
6. The sixth step is to transliterate each message in the used Arabizi corpus. This step is illustrated by the function ARABIZITRANSLITERATE(CORPUS). For translteration we rely on the same algorithm proposed and used by Guellil et al. (Guellil et al., 2018c, 2020a, 2018a)
7. The last step is to classify the sentiment (written with Arabic script) in both corpora (the initially Arabic one and the transliterated one). This step is illustrated by the function SENTIMENTCLASS($corpus, Senti_Alg$).

3.2 The used models for classification

For classification, we use two kinds of algorithms, shallow and deep. For both classifications, we extract features with word embedding techniques. With shallow classification,

Algorithm 1 Sentiment analysis of Arabic/ Arabizi messages

Input:
Eng_{lex} : English lexicon,
$ArTest_{corp}[]$: List of Arabic sentiment corpora,
$ArabiziTest_{corp}[]$: List of Arabizi sentiment corpora,
$ArabiziTrTest_{corp}[]$: List of Arabizi transliterated sentiment corpora,
$Facebook_{Key}$: A key for accessing RestFB API
Output:
Alg_{lexV1} : Algerian Lexicon V1,
Alg_{lexV2} : Algerian lexicon V2,
Ar_{corp1} ,Ar_{corp2} : Large Arabic corpora,
$Senti_{Alg}$: Automatic annotated Algerian (Arabic) corpus
L_f: List identifiant of Facebook pages,
L_{emp}: List of positive emoticons,
L_{emn}: List of negative emoticons,
L_{exp}: List of positive expressions,
L_{exn}: List of negative expressions,
L_{pr}: List of prefixes, L_{sf}: List of suffixes,
L_{neg}: List of negation terms,
L_{op}: List of opposition terms[3]

$Senti_{Alg} \leftarrow \emptyset$
$L_f, L_{emp}, L_{emn}, L_{exp}, L_{exn}, L_{pr}, L_{sf}, L_{neg}, L_{op} \leftarrow$ MANUALRESCONSTRUCTION()
Ar_{corp1} ,$Ar_{corp2} \leftarrow$ COMMENTSEXTRACTION($Facebook_{key}$)
$Alg_{lexV1} \leftarrow$ AUTOMATICARLEXCONSTRUCT(Eng_{lex})
$Alg_{lexV2} \leftarrow$ MANUALLEXREVIEW(Alg_{lexV1})
for each $m \in Ar_{corp2}$ **do**
  polarity$\leftarrow$ ANNOTATE($Alg_{lexV2}, m, L_{emp}, L_{emn}, L_{exp}, L_{exn}, L_{pr}, L_{sf}, L_{neg}, L_{op}$)
  add $m, polarity$ in $Senti_{Alg}$
end for
for each $corpus \in ArTest_{corp}$ **do**
  SENTIMENTCLASS($corpus, Senti_{Alg}$)
end for
for each $corpus \in ArabiziTest_{corp}$ **do**
  $Corpus_{tr} \leftarrow$ ARABIZITRANSLITERATE($corpus, Ar_{corp1}$)
  ADD($ArabiziTrTest_{corp}, Corpus_{tr}$)
  SENTIMENTCLASSIFICATION($Corpus_{tr}, Senti_{Alg}$)
end for

we use Word2vec algorithm. While we use fastText for deep classification. For Word2vec, we used a context of 10 words to produce representations for both CBOW and SG of length 300. For classification we use five Algorithms such as: GaussianNB (GNB), LogisticRegression (LR), RandomForset (RF), SGDClassifier (SGD, with loss='log' and penalty='l1') and LinearSVC (LSVC with C='1e1'). For deep learning classification, we first used the model presented by Attia et al. (Attia et al., 2018) with five layers using 300 filters and a width equal to 7. To enrich this model, our approach also uses the CBOW and SG of FastText for calculating the weights of embedding_matrix. Also, our approach used other deep learning algorithms, such as LSTM and Bi-LSTM. Table 1 gives more details about the configuration and architecture of the layers of our models on the training corpus. For all the models, we use an epoch equal to 100 with an early_stopping parameter. This parameter is used for stopping the iteration in the absence of improvements (for handling overfitting). This parameter allows stopping the models after 20 epochs (on average). Adam optimiser is used in all the deep learning experiments.

4 Evaluation and Discussion

4.1 Dataset

For the experiments part, the following dataset were used:

1) A large corpus (Ar_corpus2), extracted in November 2017, and containing 15,407,910 messages with 7,926,504 written in Arabic letters.

2) ALG_Senti(Guellil et al., 2018b, 2020a) is an annotated sentiment corpus which was automatically constructed based on ALGLex_V2(Guellil et al., 2020b) and on the sentiment algorithm that we proposed and implemented. The annotation process also considers other features such as the sentiment score of the message and the number of positives/negatives words recognised in the lexicon. The final corpus contains 127,004 positive messages and 127,004 negative ones.

3) TSAC[4] (Medhaffar et al., 2017) is a Tunisian sentiment corpus. This corpus is the unique corpus in the research literature, to the best of our knowledge, containing both Arabic and Arabizi. For testing our approach on other corpus presented in the research literature, we propose to transliterate the Arabizi part of TSAC into Arabic, using our transliteration approach.

4) SANA_Alg[5] (Rahab et al., 2019) is an Algerian annotated sentiment corpus. This corpus includes 513 messages that were manually annotated.

5) ASTD/QCRI/ARTwitter[6] (Altowayan and Tao, 2016) is an Arabic corpus including 1,589 tweets from astd(Nabil et al., 2015), 1, 951 tweets from ArTwitter (Abdulla et al., 2013) and 754 from QCRI(Mourad and Darwish, 2013)

[4]https://github.com/fbougares/TSAC

[5]http://rahab.e-monsite.com/medias/files/corpus.rar

[6]https://github.com/iamaziz/ar-embeddings/tree/master/datasets

4.2 Experimental results

The aim of this work is to classify an Arabic message into positive/negative automatically. More particularly to use a language model and the resources constructed for one dialect for classifying the sentiments of an another dialect (and MSA). Hence, for validating our approach , we applied it on four corpora annotated manually by natives speakers. Two of these corpora are in Algerian dialect (Senti_Alg(Guellil et al., 2018b,a) and Sana_Alg(Rahab et al., 2019)), one of them is in MSA (ASTD_QCRI_ArTwitter)(Altowayan and Tao, 2016) and the last one in Tunisian dialect (TSAC)(Medhaffar et al., 2017). Two of these corpora include both Arabic and Arabizi (Senti_Alg and TSAC) and the others are dedicated to Arabic script. Our purpose behind the different experiments is not only to validate our approach but to also highlight its adaptability to MSA and other dialects written with both scripts Arabic and Arabizi. For doing so, we apply the following steps:

1. For Senti_Alg, we focus on both sides, Arabic and Arabizi. For Arabizi part, we investigate the results using both automatic and manual transliteration.
2. As Sana_Alg and ASTD_QCRI_ARTwitter use only Arabic script, no need for the transliteration process.
3. As TSAC_Test represents a combination between Arabic and Arabizi messages, for each experiment on TSAC, we use both, the initial test (Initial_test) corpus and the test corpus obtained after applying the proposed translietration system (Transliterated_test).

The different experiments and the obtained results are presented in the following sections.

4.2.1 Results on Algerian dialect

Results on the Arabic side of Senti_Alg (Senti_Alg_test_Arabic) The experiments done on `Senti_Alg_test_Arabic` show that the best results are obtained using CBOW of Word2vec model combined with GNB classifier (F1= 87.77). For deep learning classification, the combination of FastText, CBOW and MLP gives the best results (F1-score=

Model	Layers	output shape	params
CNN	embedding_1(default/CBOW/SG)	(None,12,300)	41789100
	conv1d_d	(None,12,300)	630300
	global_max_pooling1d_1	(None,300)	0
	dropout_1	(None,300)	0
	dense_1	(None,600)	180600
	dense_2	(None,2)	1202
MLP	embedding_1(default/CBOW/SG)	(None,12,300)	41789100
	dense_1	(None,12,64)	19264
	global_max_pooling1d_1	(None,64)	0
	dropout_1	(None,64)	0
	dense_3	(None,600)	39000
	dense_4	(None,2)	1202
LSTM	embedding_1(default/CBOW/SG)	(None,12,300)	41789100
	lstm_1	(None,12,64)	93440
	global_max_pooling1d_1	(None,64)	0
	dropout_1	(None,64)	0
	dense_1	(None,600)	39000
	dense_2	(None,2)	1202
Bi-LSTM	embedding_1(default/CBOW/SG)	(None,12,300)	41789100
	bidirectional_1	(None,12,128)	186880
	global_max_pooling1d_1	(None,6128)	0
	dropout_1	(None,128)	0
	dense_1	(None,600)	77400
	dense_2	(None,2)	1202

Table 1: Deep learning models architecture

80.99). However, the results obtained using Word2vec model combined with shallow classifiers outperform those obtained using FastText model combined with deep learning classifiers. It can also be seen from this Table that CBOW model results generally outperform the results returned by using the SG model. More details are presented (in the Appendices, section 7) in the Table 1)

Results on the Arabizi side of Senti_Alg (Senti_Alg_test_Arabizi) obtained on the Arabizi side of Senti_Alg, that we named `Senti_Alg_test_Arabizi`. However, as our language model and training corpus is in Arabic script, the corpus `Senti_Alg_test_Arabizi` was firstly transliterated. For showing the efficiency of our transliteration system, we transliterate this corpus in both ways, automatically (for obtaining `Senti_Alg_test_trauto`) and manually (for obtaining `Senti_Alg_test_trmanu`). The best results for the corpus `Senti_Alg_test_trauto` were obtained using SG of Word2vec combined with SGD classifier (F1= 75.23). For deep learning classification, the combination of FastText, CBOW and CNN gives the best results for the corpus `Senti_Alg_test_trauto` (F1-score= 69.78). However, the results obtained using Word2vec model combined with shallow classifiers outperform those obtained using fastText model combined with deep learning classifiers. The results obtained by using the corpus transliterated manually (`Senti_Alg_test_trmanu`) are better than those obtained on the corpus transliterated automatically (`(Senti_Alg_test_trauto`). However, the improvement between both transliterations is non-consequential (0.9, less than 1 point For F1-score). This small improvement rate highlights the quality of the proposed transliteration system. More details are presented (in the Appendices, section 7) in the Table 2)

Results on SANA_Alg The experiments done on `SANA_Alg` show that the best results were obtained using CBOW of word2vec combined with GNB classifier (F1= 81.00). For

Test_corpora	Metrics	Best_results	Embedding	Models	Classifier
Senti_Alg_test_Arabic	P	93.50	Word2vec	CBOW	GNB
	R	90.40	Word2vec	CBOW	SGD
	F1	87.77	Word2vec	CBOW	SGD
Senti_Alg_test_trauto	P	82.18	Word2vec	CBOW	GNB
	R	82.00	Word2vec	SG	SGD
	F1	75.23	Word2vec	SG	SGD
Senti_Alg_test_trmanu	P	85.47	Word2vec	CBOW	GNB
	R	77.20	Word2vec	SG	SGD
	F1	76.13	Word2vec	SG	SGD
SANA_Alg	P	82.74	Word2vec	SG	SGD
	R	96.67	Word2vec	SG	GNB
	F1	81.00	Word2vec	CBOW	GNB
TSAC_Test	P	88.97	Word2vec	CBOW	SGD
	R	95.82	Word2vec	SG	SGD
	F1	77.29	Word2vec	SG	LSVC
TSAC_Test_Tr	P	91.66	FastText	CBOW	Bi-LSTM
	R	91.65	Word2vec	SG	SGD
	F1	91.59	FastText	SG	MLP
ASTD/QCRI/ARTwitter	P	77.65	Word2vec	CBOW	SGD
	R	85.61	Word2vec	CBOW	GNB
	F1	80.58	Word2vec	CBOW	GNB

Table 2: Synthesis of the best obtained results

deep learning classification, the combination of FastText, CBOW and LSTM gives the best results (F1-score= 63.29). However, the results obtained using Word2vec model combined with shallow classifiers outperform those obtained using FastText model combined with deep learning classifiers. The CBOW model results generally outperform the results returned by using the SG model. More details are presented (in the Appendices, section 7) in the Table 3).

4.2.2 Results on Tunisian dialect

For the experiments, we use both versions of the Tunisian corpus. We denote the version in its current state (before transliteration) as TSAC_test. We denote the version after proceeding to the transliteration as TSAC_Test_Tr. To compare the sentiment analysis results obtained before and after transliteration step, we divide Table 4 into two parts: the first one illustrates the sentiment classification results obtained on TSAC_test and the second one, the results obtained on TSAC_Test_Tr. For the experiments done on both corpora, it can be seen that the best results were obtained using SG of Word2vec combined with SGD classifier (F1= 73.69). For deep learning classification, the combination of FastText, CBOW/SG and CNN gives the best results for the corpus TSAC_Test (F1-score= 52.88). For the corpus TSAC_Test_Tr, the best results were obtained using SG of Word2vec combined with SGD classifier (F1= 75.24). For deep learning classification, the combination of FastText, CBOW and CNN gives the best results for the corpus TSAC_Test_Tr (F1-score= 62.98). More details are presented (in the Appendices, section 7) in the Table 4)

4.2.3 Results on MSA

Experiments using ASTD/QCRI/ARTwitter The experiments done on `ASTD/QCRI/ARTwitter` show that the best results obtained using CBOW of word2vec combined with GNB classifier (F1= 80.58). For deep learning classification, the combination of FastText, CBOW and CNN gives the best results (F1-score= 64.03). However, the results obtained using Word2vec model combined with shallow classifiers outperform those obtained using FastText model combined with deep learning classifiers. More details are presented (in the Appendices, Section 7) in the Table 6)

5 Synthesis and corpus validation

5.1 Synthesis

The best results obtained from the different experiments and that we discussed in Section 4.2 are summarised in Table 2.

For Algerian dialect, the corpora that we used (i.e. `Senti_Alg_test_Arabic`, `Senti_Alg_test_trauto` and `Senti_Alg_test_trmanu`) were presented and used in many research papers (Guellil et al., 2017, 2018b,a; Imane et al., 2019). We based on the issues of each presented research work to improve the results presented in this paper (where the best F1= 87.77% for the Arabic side and F=76.13% for the Arabizi side, after transliteration). The best results obtained on SANA_Alg are up to 81.00% (for F1-score). This result outperforms the results presented in the research literature, where the F1-score presented by Rahab et al. .(Rahab et al., 2019) was up to 75%. Hence, our approach and corpus lead to an improvement of 6% on this corpus.

For Tunisian dialect, it can be seen that the results obtained by using the corpus transliterated `(TSAC_Test_Tr)` are relatively better than those obtained on the initial corpus `(TSAC_Test)` (without transliteration). Medhaffar et al. (Medhaffar et al., 2017) obtained an F1-score up to 78% for TSAC_Test corpus. Our best results by using our approach on the corpus (Senti_Alg) is up to 75.24% (F1-score). The results are then comparable to the results obtained by the authors (even with a corpus constructed automatically and dedicated to Algerian dialect). However, our transliteration system drastically improves the results. The results are up to 91.59% after transliterating both the training and the testing corpus (by using TSAC_train for the training). Hence an improvement of 14% was observed on this corpus. Another interesting observation is that, except for the training corpus, all the approach and corpora used for TSAC corpus are the same that we used for our other experiments. The vast corpus used for training Word2vec and fastText dedicated to Algerian dialect. The language model used for extracting the best candidate transliteration was also dedicated to Algerian dialect. Finally, concerning MSA, we opt for using the corpus ASTD/QCRI/ArTwitter (Altowayan and Tao, 2016). The best results obtained by Altowayen et al. (Altowayan and Tao, 2016) are up to 79.62% (for F1-score). It can be seen from Table 6 that the best results that we obtained are up to 80.58% (for F1-score). Moreover, This corpus is dedicated to MSA with a focus on Egyptian dialect (for ASTD). Hence, our approach and corpus, which are dedicated to Algerian dialect, outperforms the results presented for corpora dedicated to MSA and Egyptian dialect.

5.2 Manual corpus validation

To validate the constructed corpus automatically, we focus on a sample containing 3,048 messages (1,488 positives and 1,560 negatives). Afterwards, we manually review this sample. The messages that are correctly annotated were kept, and the messages which were wrongly annotated were corrected. Our first observation is that, among the 3,048 messages that are manually reviewed, 85.17% (2,596 messages) were correctly annotated. To the best of our knowledge, this corpus is the first manually checked annotated sentiment corpus that handles DALG as well as MSA. For showing the efficiency of the manually reviewed corpus, we present Table 3. Almost all the results were improved with the corpus, which was reviewed manually. The best F1 on Senti_Alg_test_Arabic is now up to 90.16 (it was up to 87.77 with Senti_Alg_auto). The best F1 on Senti_Alg_test_trauto is now up to 80.95 (it was up to 75.23 with Senti_Alg_auto). The best F1 on Senti_Alg_test_trmanu is now up to 83.10 (it was up to 76.13 with Senti_Alg_auto). The best F1 on ASTD/QCRI/ARTwitter is now up to 81.75 (it was up to 80.58 with Senti_Alg_auto). The decrease for SANA_Alg is insignificant, where the best F1 was up to 81.00, and now it is up to 80.97.

Concerning the experiments on Tunisian corpus (TSAC), It can be seen from Table 3 the reviewed corpus outperforms the results obtained by the Algerian corpus constructed automatically. For the corpus constructed auto-

Test_corpora	Metrics	Best_results	Embedding	Models	Classifier
Senti_Alg_test_Arabic	P	95.43	Word2vec	SG	LR
	R	91.60	Word2vec	SG	SGD
	F1	90.16	Word2vec	SG	SGD
Senti_Alg_test_trauto	P	88.21	Word2vec	SG	LR
	R	78.40	fastText	CBOW	Bi-LSTM
	F1	80.95	Word2vec	SG	LR
Senti_Alg_test_trmanu	P	90.73	Word2vec	SG	LR
	R	82.00	Word2vec	SG	SVC
	F1	83.10	Word2vec	SG	SGD
SANA_Alg	P	91.36	Word2vec	CBOW	SGD
	R	93.75	Word2vec	SG	GNB
	F1	80.97	Word2vec	SG	LR
TSAC_Test	P	88.94	Word2vec	CBOW	SGD
	R	92.82	Word2vec	SG	SGD
	F1	75.61	Word2vec	SG	SGD
TSAC_Test_Tr	P	87.14	Word2vec	SG	LR
	R	87.41	Word2vec	CBOW	SGD
	F1	80.69	Word2vec	SG	SGD
ASTD/QCRI/ARTwitter	P	89.29	Word2vec	SG	SGD
	R	88.36	Word2vec	SG	GNB
	F1	81.75	Word2vec	SG	LR

Table 3: Synthesis of the best-obtained results on the manually reviewed corpus

matically, F1 was up to 73.69 (for TSAC_Test) and up to 75.24 (for TSAC_Test_TR). By using the manually reviewed corpus, F1 is up to 75.61 (for TSAC_Test) and up to 80.69 (for TSAC_Tr). These results outperform the results presented in the research literature ((Medhaffar et al., 2017)), where the best-presented F1 was up to 78.00. Hence, the manual reviewing of a corpus which was initially constructed automatically outperforms all the results presented in the research literature.

6 Conclusion

The major contribution in this paper is the new perspectives that it opens:

- Automatic training corpus construction.
- Using one language model trained for one dialect to MSA and either to other dialects.
- Moreover, using the training corpus of one dialect to others (which is a case of transfer learning).
- Stop handling Arabizi as it is. Transliteration is crucial for improving the results.

Moreover, only simple word embedding models were used (word2vec and fastText). It was for showing the efficacy of the approach even with the fastest models. However, in the future, we are planning to improve this approach with the most recent models such as BERT(Devlin et al., 2018) or ELMO(Peters et al., 2018).

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.

Nawaf Abdulla, Salwa Mohammed, Mahmoud Al-Ayyoub, Mohammed Al-Kabi, et al. 2014. Automatic lexicon construction for arabic sentiment analysis. In *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*, pages 547–552. IEEE.

Nawaf A Abdulla, Nizar A Ahmed, Mohammed A Shehab, and Mahmoud Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In *Applied Electrical Engineering and Computing Technologies (AEECT), 2013 IEEE Jordan Conference on*, pages 1–6. IEEE.

Mohamed Al-Badrashiny, Ramy Eskander, Nizar Habash, and Owen Rambow. 2014. Automatic transliteration of romanized dialectal arabic. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 30–38.

Nora Al-Twairesh, Hend Al-Khalifa, AbdulMalik Al-Salman, and Yousef Al-Ohali. 2017. Arasenti-tweet: A corpus for arabic sentiment analysis of saudi tweets. *Procedia Computer Science*, 117:63–72.

Rahma M Alahmary, Hmood Z Al-Dossari, and Ahmed Z Emam. 2019. Sentiment analysis of saudi dialect using deep learning techniques. In *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–6. IEEE.

A Aziz Altowayan and Lixin Tao. 2016. Word embeddings for arabic sentiment analysis. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 3820–3825. IEEE.

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.

Mohammed Attia, Younes Samih, Ali El-Kahky, and Laura Kallmeyer. 2018. Multilingual multi-class sentiment classification using convolutional neural networks. In *LREC*.

Gaétan Baert, Souhir Gahbiche, Guillaume Gadek, and Alexandre Pauchet. 2020. Arabizi language models for sentiment analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 592–603.

Fady Baly, Hazem Hajj, et al. 2020. Arabert: Transformer-based model for arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15.

Naaima Boudad, Rdouan Faizi, Rachid Oulad Haj Thami, and Raddouane Chiheb. 2017. Sentiment analysis in arabic: A review of the literature. *Ain Shams Engineering Journal*.

Kareem Darwish. 2013. Arabizi detection and conversion to arabic. *arXiv preprint arXiv:1306.6755*.

Kareem Darwish. 2014. Arabizi detection and conversion to arabic. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 217–224.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rehab M Duwairi, Mosab Alfaqeh, Mohammad Wardat, and Areen Alrabadi. 2016. Sentiment analysis for arabizi text. In *Information and Communication Systems (ICICS), 2016 7th International Conference on*, pages 127–132. IEEE.

Samhaa R El-Beltagy. 2016. Nileulex: A phrase and word level sentiment lexicon for egyptian and modern standard arabic. In *LREC*.

Hady ElSahar and Samhaa R El-Beltagy. 2015. Building large arabic multi-domain resources for sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 23–34. Springer.

Ali Farghaly and Khaled Shaalan. 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4):14.

Donia Gamal, Marco Alfonse, El-Sayed M El-Horbaty, and Abdel-Badeeh M Salem. 2019. Twitter benchmark dataset for arabic sentiment analysis. *International Journal of Modern Education and Computer Science*, 11(1):33.

Badaroand Gilbert, Jundiand Hussein, Hajj Hazem, El-Hajj Wassim, and Habash Nizar. 2018. Arsel: A large scale arabic sentiment and emotion lexicon.

Imane Guellil, Ahsan Adeel, Faical Azouaou, Fodil Benali, Ala-eddine Hachani, and Amir Hussain. 2018a. Arabizi sentiment analysis based on transliteration and automatic corpus annotation. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 335–341.

Imane Guellil, Ahsan Adeel, Faical Azouaou, and Amir Hussain. 2018b. Sentialg: Automated corpus annotation for algerian sentiment analysis. *arXiv preprint arXiv:1808.05079*.

Imane Guellil, Faical Azouaou, Fodil Benali, Ala Eddine Hachani, and Marcelo Mendoza. 2020a. The role of transliteration in the process of arabizi translation/sentiment analysis. In *Recent Advances in NLP: The Case of Arabic Language*, pages 101–128. Springer.

Imane Guellil, Faical Azouaou, Fodil Benali, ala-eddine Hachani, and Houda Saadane. 2018c. Approche hybride pour la translitération de l'arabizi algérien : une étude préliminaire. In *Conference: 25e conférence sur le Traitement Automatique des Langues Naturelles (TALN), May 2018, Rennes, FranceAt: Rennes, France*. https://www.researchgate.net/publication/326354578_Approche_Hybride_pour_la_transliteration_de_l%27arabizi_algerien_une_etude_preliminaire.

Imane Guellil, Faical Azouaou, and Francisco Chiclana. 2020b. Arautosenti: Automatic annotation and new tendencies for sentiment classification of arabic messages. *Social Network Analysis and Mining*, 10(1):1–20.

Imane Guellil, Faical Azouaou, Houda Saâdane, and Nasredine Semmar. 2017. Une approche fondée sur les lexiques d'analyse de sentiments du dialecte algérien.

Imane Guellil, Faical Azouaou, and Alessandro Valitutti. 2019a. English vs arabic sentiment analysis: A survey presenting 100 work studies, resources and tools. In *2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE.

Imane Guellil, Houda Saâdane, Faical Azouaou, Billel Gueni, and Damien Nouvel. 2019b. Arabic natural language processing: An overview. *Journal of King Saud University-Computer and Information Sciences*.

Nizar Y Habash. 2010. Introduction to arabic natural language processing. *Synthesis Lectures on Human Language Technologies*, 3(1):1–187.

Salima Harrat, Karima Meftouh, and Kamel Smaïli. 2017. Maghrebi arabic dialect processing: an overview. In *ICNLSSP 2017-International Conference on Natural Language, Signal and Speech Processing*.

Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska de Jong, and Uzay Kaymak. 2013. Exploiting emoticons in sentiment analysis. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 703–710. ACM.

Guellil Imane, Darwish Kareem, and Azouaou Faical. 2019. A set of parameters for automatically annotating a sentiment arabic corpus. *International Journal of Web Information Systems*.

Shereen Khoja and Roger Garside. 1999. Stemming arabic text. *Lancaster, UK, Computing Department, Lancaster University*.

M'hamed Mataoui, Omar Zelmati, and Madiha Boumechache. 2016. A proposed lexicon-based sentiment analysis approach for the vernacular algerian arabic. *Research in Computing Science*, 110:55–70.

Jonathan May, Yassine Benjira, and Abdessamad Echihabi. 2014. An arabizi-english social media statistical machine translation system. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas*, pages 329–341.

Salima Medhaffar, Fethi Bougares, Yannick Esteve, 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.

Saif Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. 2016a. Sentiment lexicons for arabic social media. In *LREC*.

Saif M Mohammad, Mohammad Salameh, and Svetlana Kiritchenko. 2016b. How translation alters sentiment. *Journal of Artificial Intelligence Research*, 55:95–130.

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.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Hichem Rahab, Abdelhafid Zitouni, and Mahieddine Djoudi. 2019. Sana: Sentiment analysis on newspapers comments in algeria. *Journal of King Saud University-Computer and Information Sciences*.

Mohammad Salameh, Saif Mohammad, and Svetlana Kiritchenko. 2015. Sentiment after translation: A case-study on arabic social media posts. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 767–777.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2016. A simple but effective approach to improve arabizi-to-english statistical machine translation. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 43–50.

Payal Yadav and Dhatri Pandya. 2017. Sentireview: Sentiment analysis based on text and emoticons. In *Innovative Mechanisms for Industry Applications (ICIMIA), 2017 International Conference on*, pages 467–472. IEEE.

7 Appendices

In this section, more details about the obtained results on each model, corpus are given.

Model	Type	ML Algo	Arabic		
			P	R	F1
Word2vec	CBOW	GNB	**93.50**	74.80	83.11
		LR	82.09	88.00	84.94
		RF	85.07	75.20	79.83
		SGD	85.28	**90.40**	**87.77**
		LSVC	82.71	88.00	85.27
	SG	GNB	90.34	74.80	81.84
		LR	85.10	86.80	85.94
		RF	85.59	76.00	80.51
		SGD	84.62	88.00	86.27
		LSVC	85.32	86.00	85.66
FastText	CBOW	CNN	80.03	80.00	79.99
		MLP	81.04	81.00	80.99
		LSTM	79.65	79.60	79.59
		Bi-LSTM	79.92	79.60	79.54
	SG	CNN	78.44	78.20	78.15
		MLP	79.63	79.60	79.59
		LSTM	79.04	79.00	78.99
		Bi-LSTM	76.84	76.80	76.79

Figure 1: Results on the Arabic side of Senti_Alg (Senti_Alg_test_Arabic)

Model	Type	ML Algo	Senti_Alg_test_trauto			Senti_Alg_test_trmanu		
			P	R	F1	P	R	F1
Word2vec	CBOW	GNB	**82.18**	57.20	67.45	**85.47**	58.80	69.67
		LR	69.81	74.00	71.84	75.00	76.80	75.89
		RF	73.06	64.00	68.23	72.20	64.40	68.08
		SGD	69.10	79.60	73.98	73.71	74.00	73.85
		LSVC	70.04	72.00	71.01	75.70	76.00	75.85
	SG	GNB	79.50	63.60	70.67	85.41	63.20	72.64
		LR	68.40	73.60	70.91	73.08	76.00	74.51
		RF	72.29	66.80	69.44	76.47	67.60	71.76
		SGD	69.49	**82.00**	**75.23**	75.10	**77.20**	**76.13**
		LSVC	69.08	72.40	70.70	72.76	74.80	73.77
FastText	CBOW	CNN	69.85	69.80	69.78	73.65	73.60	73.58
		MLP	67.64	67.60	67.58	71.81	71.80	71.80
		LSTM	68.69	68.60	68.56	70.01	70.00	70.00
		Bi-LSTM	68.95	68.80	68.74	71.93	71.40	71.22
	SG	CNN	68.52	68.20	68.06	73.29	72.60	72.40
		MLP	68.25	68.20	68.18	71.37	71.20	71.14
		LSTM	69.42	69.40	69.39	72.60	72.60	72.60
		Bi-LSTM	68.84	68.80	68.78	70.60	70.60	70.60

Figure 2: Results on the Arabizi side of Senti_Alg (Senti_Alg_test_Arabizi) after transliteration

Model	Type	ML Algo	SANA_Alg		
			P	R	F1
Word2vec	CBOW	GNB	81.17	80.83	**81.00**
		LR	76.23	70.83	73.43
		RF	71.54	77.50	74.40
		SGD	80.28	72.92	76.42
		LSVC	74.44	69.17	71.71
	SG	GNB	62.37	**96.67**	75.82
		LR	79.09	72.50	75.65
		RF	72.05	76.25	74.09
		SGD	**82.74**	67.92	74.60
		LSVC	78.80	71.25	74.84
FastText	CBOW	CNN	62.26	62.30	62.28
		MLP	59.65	60.00	59.64
		LSTM	63.42	63.22	63.29
		Bi-LSTM	62.08	62.07	60.29
	SG	CNN	60.76	61.15	60.22
		MLP	57.57	57.93	57.62
		LSTM	60.37	60.23	60.29
		Bi-LSTM	59.28	59.77	58.87

Figure 3: Results on SANA_Alg

Model	Type	ML Algo	TSAC_Test			TSAC_Test_Tr		
			P	R	F1	P	R	F1
Word2vec	CBOW	GNB	**80.31**	36.71	50.38	**81.25**	65.76	72.69
		LR	61.38	88.82	72.60	70.06	77.35	73.53
		RF	58.75	78.82	67.32	70.01	64.41	67.10
		SGD	61.81	89.29	73.05	71.72	77.88	74.68
		LSVC	61.38	87.88	72.28	70.27	76.76	73.38
	SG	GNB	60.92	89.76	72.58	71.51	76.76	74.04
		LR	75.45	41.76	53.77	72.37	76.41	74.33
		RF	59.15	77.59	67.12	67.28	60.35	63.63
		SGD	62.02	**90.76**	**73.69**	71.44	**79.47**	**75.24**
		LSVC	75.09	38.82	51.18	72.48	75.76	74.09
FastText	CBOW	CNN	59.69	56.62	52.88	63.84	63.32	62.98
		MLP	58.86	55.97	52.06	62.65	62.50	62.39
		LSTM	59.12	55.79	51.36	63.01	62.21	61.61
		Bi-LSTM	57.36	55.29	51.93	62.08	61.91	61.78
	SG	CNN	57.70	55.79	52.88	61.45	61.38	61.33
		MLP	55.87	54.21	50.71	62.21	62.15	62.10
		LSTM	56.33	42.4	50.11	61.09	60.85	60.65
		Bi-LSTM	57.25	54.88	50.87	61.36	61.15	60.96

Figure 4: Results on TSAC_Test by using Senti_Alg as training

Model	Type	ML Algo	TSAC_Test			TSAC_Test_Tr		
			P	R	F1	P	R	F1
Word2vec	CBOW	GNB	78.65	32.29	45.79	82.39	57.24	67.55
		LR	65.08	89.47	75.35	84.76	84.76	84.76
		RF	62.51	83.76	71.59	87.13	78.82	82.77
		SGD	64.34	92.12	75.76	82.55	89.06	85.68
		LSVC	85.70	42.65	56.95	84.65	87.29	85.95
	SG	GNB	76.39	31.59	44.69	82.70	56.53	67.16
		LR	65.66	**89.88**	**75.89**	87.26	87.41	87.33
		RF	83.45	33.82	48.14	87.53	78.88	82.98
		SGD	65.46	89.65	75.67	83.76	**91.65**	87.53
		LSVC	**88.10**	43.53	58.27	86.64	88.12	87.37
FastText	CBOW	CNN	75.53	66.50	63.25	89.94	89.65	89.63
		MLP	75.29	67.21	64.36	90.81	90.76	90.76
		LSTM	75.71	67.41	64.55	91.52	91.41	91.41
		Bi-LSTM	77.53	67.44	64.16	**91.66**	91.21	91.18
	SG	CNN	75.85	66.85	63.69	91.58	91.47	91.46
		MLP	76.13	67.50	64.57	91.65	91.59	**91.59**
		LSTM	75.78	66.91	63.80	90.85	90.71	90.70
		Bi-LSTM	77.10	65.59	61.50	91.39	91.03	91.01

Figure 5: Results on TSAC_Test using TSAC_train_Tr

Model	Type	ML Algo	Arabic		
			P	R	F1
Word2vec	CBOW	GNB	76.11	**85.61**	**80.58**
		LR	71.93	75.31	73.58
		RF	69.79	68.00	69.28
		SGD	**77.65**	73.45	75.49
		LSVC	70.90	74.66	72.73
	SG	GNB	66.33	92.59	77.29
		LR	71.67	77.64	74.54
		RF	71.09	67.12	69.05
		SGD	71.12	82.58	76.42
		LSVC	71.40	77.08	74.13
FastText	CBOW	CNN	64.24	64.11	64.03
		MLP	62.65	62.65	62.65
		LSTM	61.40	61.09	60.81
		Bi-LSTM	62.97	62.88	62.81
	SG	CNN	63.30	63.27	63.26
		MLP	60.83	60.81	60.78
		LSTM	60.58	60.43	60.30
		Bi-LSTM	60.48	60.41	60.34

Figure 6: Results of ASTD/QCRI/ArTwitter