Sentiment Analysis for Arabizi: Application to Algerian Dialect

Asma Chader[1], Dihia Lanasri[1,2], Leila Hamdad[1], Mohamed Chemes Eddine Belkheir[1] and Wassim Hennoune[1]

[1]*Laboratoire de la Communication dans les Systèmes Informatiques (LCSI), Ecole Nationale Supérieure d'Informatique (ESI), BP 68M, 16309, Oued-Smar, Algiers, Algeria*

[2]*Research & Development & Innovation Direction, Brandt, Algiers, Algeria*

Keywords: Sentiment Analysis, Algerian Dialect, Arabizi, Social Networks, Supervised Classification.

Abstract: Sentiment Analysis and its applications have spread to many languages and domains. With regard to Arabic and its dialects, we witness an increasing interest simultaneously with increase of Arabic texts volume in social media. However, the Algerian dialect had received little attention, and even less in Latin script (Arabizi). In this paper, we propose a supervised approach for sentiment analysis of Arabizi Algerian dialect using different classifiers such as Naive Bayes and Support Vector Machines. We investigate the impact of several preprocessing techniques, dealing with dialect specific aspects. Experimental evaluation on three manually annotated datasets shows promising performance where the approach yielded the highest classification accuracy using SVM algorithm. Moreover, our results emphasize the positive impact of proposed preprocessing techniques. The adding of vowels removal and transliteration, to overcome phonetic and orthographic varieties, allowed us to lift the F-score of SVM from 76 % to 87 %, which is considerable.

1 INTRODUCTION

Nowadays, sentiment analysis is a topic of great interests in both research and industry. With billions of comments and reviews produced every day, available information over internet and social media is constantly growing and mining user's opinions has become a key tool in different applications such as marketing and politics. Such opinions can be used for a better profiling of the users, understanding their expectations and criticisms to adapt brands services, discovering public opinions about different policies or even predicting election results (Medhat et al., 2014). Therefrom, automatic extraction and analysis of opinions from created content, known as sentiment analysis (SA) or opinion mining (OM), emerged as a new challenging thematic of Natural Language Processing (NLP). SA field can be seen as a classification task of whether an opinion is positive, negative or neutral (Medhat et al., 2014). One of its biggest challenges remains language. In social media particularly, most users communicate with abbreviations and unstructured dialectal texts that vary significantly from region to region and from a culture to another.

Recent studies focused gradually on Arabic language sentiment analysis (Al-Ayyoub et al., 2019). However, they are generally restricted to Modern Standard Arabic (MSA), which is regulated and standardized but only used in formal communication. Most Arabic social media texts are actually written in dialects and often mixed with foreign languages (e.g. French or English). This leads to another phenomenon, even more challenging: the non standard Romanization of Arabic, called Arabizi, which consists on using Latin alphabet, numbers or punctuation to write an Arabic word. For instance, the word "3jebni", romanized form of "عجبني" in Arabic, meaning "I like it", is composed of Latin letters and a number (used to symbolize Arabic letter "ع" that have no phonetic Latin equivalent, but his shape seems as the number). In literature, only limited work addressed Arabizi sentiment analysis compared to Arabic script, in particular for Algerian dialects (AlgD). The reason behind this is complexity of analysis due to the absence of specific and predefined rules to write Arabizi, the frequent misspellings and adapted expressions used in addition to the variety of Algerian accents that differ from one region to another.

We aim in this research to determine sentiment of Algerian Arabizi. We apply a machine leaning approach to automatically classify social media mes-

Chader, A., Lanasri, D., Hamdad, L., Belkheir, M. and Hennoune, W.
Sentiment Analysis for Arabizi: Application to Algerian Dialect.
DOI: 10.5220/0008353904750482
In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019)*, pages 475-482
ISBN: 978-989-758-382-7

sages written in AlgD in both Latin and Arabic script (after transliteration). To handle the specific aspects of AlgD, very different from other Arabic dialects, we propose and evaluate new steps as a part of preprocessing, namely, phonetic grouping and removing vowels. We collect and annotate (manually and using a tailored built-in crowdsourcing tool) three datasets to enable evaluation.

This paper is organized as follows: In Section 2, related work is presented. Section 3 describes our sentiment analysis approach. Section 4 presents different datasets and evaluation results. Finally, Section 5 concludes the paper with some future directions.

2 RELATED WORKS

This section outlines research related to Arabic sentiment analysis field with focus on dialectal studies that we present with respect to the three main approaches used in mining sentiment, namely, lexicon based, machine learning based and hybrid approach.

Several studies were interested in Arabic and its dialects sentiment analysis with supervised approaches (using annotated datasets to train classifiers). For instance, (Cherif et al., 2015) used SVM to classify MSA Arabic reviews and comments on hotels (collected from Trip Advisor website) into five categories: excellent, very good, middling, weak and horrible. Likewise, (Hadi, 2015) examined different classifiers on Arabic corpus of 3700 tweets manually annotated by native speakers in positive, negative and neutral. The conducted experiments showed that SVM outperformed k-NN , NB and Decision Tree (DT) in terms of accuracy. In another work,(Abdul-Mageed et al., 2014) presented "SAMAR" system for subjectivity and sentiment analysis of both MSA and Egyptian dialect. They built a sentiment lexicon consisting of 3982 adjectives (labelled as positive, negative or neutral) and used SVM-light toolkit for classification. They reported an accuracy of 73.49 % for dialect, affecting negatively the performance. Few work addressed Arabizi dialects. There are for example (Zarra et al., 2017) in Maghrebi and (Medhaffar et al., 2017; Ali et al., 2018) in Tunisian dialects. In (Medhaffar et al., 2017), authors used several ML classifiers (Multi-Layer Perceptron, SVM, NB) to determine the polarity of comments constructing the TSAC a Tunisian corpus dedicated to sentiment analysis.

On the other side, lexicon based approaches present the advantage of not requiring a manually annotated corpus. In such approaches, a list of words with annotated opinion polarities, called lexicon, is usually created then used to predict the polarity of text. First studies were essentially conducted on MSA Arabic; we refer for example to those of (Al-Ayyoub et al., 2015; Badaro et al., 2014; Bayoudhi et al., 2015). Thereafter, more attention has been drawn on dialectal analysis. In (Abdulla et al., 2014), authors applied a lexicon-based algorithm to tweets and comments in both MSA and Jordanian dialect. To construct their lexicon (4000 words), they expand a seed of 300 words using synonym and antonym relations. As to AlgD, (Mataoui et al., 2016), starting with an existing Arabic and Egyptian lexicons, built three AlgD lexicons (a keywords lexicon, a negation words lexicon, and intensification words lexicon). These three lexicons are then enriched by a dictionary of emoticons and another of common phrases. They tested their approach under different configurations and reported a 79.13 % accuracy.

Finally, the hybrid approach combines lexicon-based and machine learning approaches. (Mustafa et al., 2017) proposed a hybrid approach to SA. Its lexicon-based phase, extracts polarities of data using a look-up table stemming technique to annotate the training corpus, while the supervised phase uses the annotated data to train SVM and NB sentiment classifier; which achieved an accuracy of 96 % on the MIKA (Ibrahim et al., 2015) corpus extended with Egyptian dialect. Similarly, (Bettiche et al., 2018) presented a hybrid approach to determine polarity of Arabizi AlgD comments. They reported 93.7 % in terms of F-score.

As can be noticed from related work, different approaches have been used to analyze different Arabic dialects. However, only few research concerns with Algerian dialect. We propose a supervised approach to Arabizi AlgD sentiment analysis by training and testing different classifiers (SVM, NB, DT).

3 OUR APPROACH

Our approach aims to analyze sentiments of a given AlgD document retrieved from social networks. To achieve this goal, we propose a process composed mainly of 6 major steps (see figure 1): i) *Data collection* to gather Algerian comments from different social networks. ii) *Language detection* allowing to identify the Algerian comments among the other languages. iii) *Data annotation* consisting on associating a polarity label to each comment in the Algerian dataset. iv) *Data preprocessing* considered as the main step in our approach presented as a pipeline aiming to treat the Algerian dataset and generate a valid & clean one, ready to be exploited by the machine learning models. v) *Data representation* used to gen-

erate a vector from a text entry. vi) *Data classification* in which the classifier is defined basing on machine learning models in order to detect the class of messages {positive, negative or neutral}. The major steps presented in this approach are familiar in sentiment analysis domain applied on regular languages like French, English, classic Arabic. But the particularity of AlgD, that we can resume in syntactic & semantic variation of Algerian vocabulary due to the variety of Algerian regions (Oran, Kabyle, Sahara...) each one having its specific properties and accents, motivates us to adapt the process mainly the preprocessing step, where we propose new modules (Transliteration, Phonetic grouping, Removing vowels) very useful and powerful to prepare an Algerian text even written in Latin or Arabic characters that will serves as an entry corpus to the classifier model. The details about our process are given in what follows:

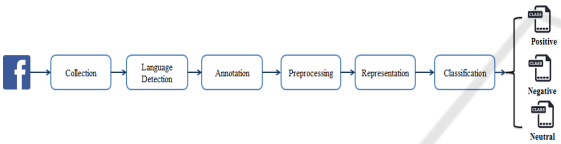

Figure 1: Our Algerian dialect sentiment analysis approach.

1- Data Collection: Before starting the process of sentiment analysis & development of machine learning models, a dataset composed of an important size of AlgD messages even written in Arabic or Latin characters should be prepared. Many ways exists to get an Algerian dataset like: extracting comments from social networks, using an existing datasets like Wach-t7ass (Mataoui et al., 2016).

2- Data Language Detection: As we note, comments published on Algerian social networks are more often a mix of dialect, classical Arabic, French, English and even Tamazight. To rely to our defined problem consisting on detecting sentiments of AlgD comments, we have to start by a language detection, which is necessary for our system. For this, our solution eliminates all comments which are in regular language (French, English, Spanish and classical Arabic). This implies that we keep just AlgD text.

3- Data Annotation: The objective of the annotation step consists on associating a polarity label (positive, negative, neutral) to each message, which represents its sentiment. To annotate the different collected comments of AlgD dataset, we propose to use manual method. According to the diversity of the AlgD due to the different regions and accents, we opted for manual annotation which is more reliable than the automatic one. To accelerate the process of annotation, we propose to use crowd sourcing method basing on a double annotation, where two people annotate the same comments in order to be sure of the given label. As the process is really complex and tiring, we propose to use an annotating application available in https://github.com/chakki-works/doccano to help annotators to accelerate this task.

4- Data Preprocessing: This step is the core of our approach of AlgD sentiment analysis. Before starting the classification of messages into positive, negative or neutral, a clean dataset should be provided to the machine learning model in order to get a powerful classification model with higher score. As we noticed, most comments retrieved from social networks are not prepared and not clean, with semantic and syntactic variation because they are written by different people of different intellectual levels. Furthermore, the richness of the AlgD leads to a variety of words and variety of meaning for the same word according to each region. Moreover, we identified other problems in comments like: spelling errors, presence of: links, hashtags, gifs, stickers, special characters, etc., which must imperatively be removed and these messages should be deduplicated in order to have a minimal, unified, valid and clean corpus ready to be exploited. Our aim is to keep maximum of variant informative vocabulary in order to enrich our corpus. In what follows we will detail the pre-processing steps (see figure 2) and focus on our contribution:

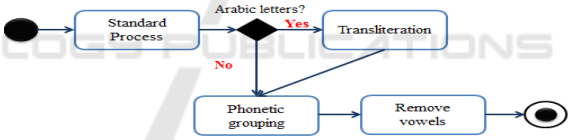

Figure 2: Preprocessing pipeline.

4.1. Tokenization: This first step permits decomposing a string of characters (message or comment) into words called "Tokens". Tokenization is even more important in the sentiment analysis than in other areas of the NLP, because feelings' information are often poorly represented. A single group of punctuation marks such as ">:-(" could tell the right feelings, in this case is "I'm upset". Considering importance of extracting tokens, several functions are implemented in different languages that allow us to do this step.

4.2. Remove Noise: This step includes a set of tasks that allow to remove the noise from the Algerian dataset (capital letters, accents, links, hashtags ...), detailed in what follows:

4.2.1. Remove Hypertext Links: In this step, all hypertext links (URLs) are eliminated using regular expressions and algorithmic functions. This allows reducing the size of our vocabulary.

4.2.2. Remove Hashtags: Hashtags are omnipresent

in sentiment analysis area, since it can reorient the polarity of a sentence easily. For instance, *"win nroh, ew khsarli TV # 3ayitouna"*, *# 3ayitouna* is negative. Internet users tend to use Hashtags to explain the subject that their message or comment deals with. But in our case it does not help much to keep them since we rely on an automatic learning approach hence the need to remove all hashtags of our corpus.

4.2.3. Remove Repeated Letters: In this step, the process deletes the repeated letters (more than twice) in a word by reducing them into a single letter. Although we are convinced that the repeated letters in a word of opinion carry a stronger feeling,i.e, they play the role of a feeling intensifier; for Example: 'Raw3aaaaaaa' is positive word stronger than 'Raw3a'; we must delete this intensification since the approach for which we opt is not lexicon-based and our goal is to build a homogeneous clean corpus.

4.2.4. Remove Stop Words: A stop word is a useless and non-significant word appearing in a text, hence the need to eliminate it from our corpus. For this purpose, for languages such as English, French or Modern Standard Arabic, there are lists of well-known stop words. These lists/tools are freely available like NLTK[1]. Nevertheless, there is no defined or elaborated resource for AlgD stop words to consider hence the obligation to create them. The difficulty lies in identifying all the stop words or words without added meaning in order to eliminate them later from our corpus. In Algeria we have several accents, and this requires to colaborate with a substantial human resource and experienced in linguistics and dialects to encompass all these stop words. To facilitate our work, we have followed an approach aiming to create a general void list for the dialect and which can be used as a reliable source for future works on the given dialect language.

4.3. Remove Isolated Single Letters: The purpose of this step is similar to stop words one, except that this part does not require a predefined list, since here we delete all the words of a single letter, because they did not bring any useful information to this analysis.

4.4. Treatment of Capitalized Words: In order to unify our corpus and reduce the vector of features called also the dimension, our process transforms all the words into lowercase in order to ensure the presence of a single instance of each word in our dataset.

4.5. Dis-accentuation: The aim of this step is to eliminate accentuation from some Latin letters like (é, à) and substitute them with (e, a) in order to keep single instance of each word in the dataset.

4.6. Stemming: Stemming or rooting is a really important step in the process of sentiment analysis since it allows reducing the size of the corpus to the maximum. Plenty of stemmers are developed like: Snowball[2] used to root French or English words and keep their Stems, basing on Porter algorithm [3] or "Arabic-Stemmer"[4] used for stemming classic Arabic text, incorporated into the NLTK library in Snowball pack. Otherwise, there is no stemmer elaborated for AlgD words, mainly because dialect does not belong to any regular language and especially with syntactic and grammatical rules allowing rooting. So we ended by ignoring this step in our contribution.

These next steps are specific to treat messages written in AlgD after going through the previous steps (*Basic preprocessing*). As we have already mentioned, the problem with Algerian dialect is that it doesn't rely to any orthographic rules, thus several spellings can be observed in the corpus for the same word. Example: We3lach, We3lech, waalach, 3lach, Elach, oualah, olah, wealache, etc. To overcome this problem, we propose including three steps: transliteration, phonetic grouping and vowels removing. More details are given in what follows:

4.7. Transliteration: We noticed that AlgD comments on social networks can be written in two forms: i)Arabic dialect is the standard normal Arabic writing format. ii)Arabizi dialect which is a form of writing an Arabic text based on Latin letters, numbers and punctuation rather than Arabic letters. In the literature, the difficulties related to sentiment analysis in Arabizi have been underestimated, mainly because of the complexity of Arabizi. To deal with the two cases, we propose the transliteration step aiming to transform a word written in Arabic form to Latin one. Some works and systems are elaborated in Arabic-Latin-Arabic transliteration such as Din 31635, Buckwalter, Qalam[5], etc. Our process is based on the Qalam system for Arabic-Latin transliteration thanks to the similarities of the latter to the Algerian dialect (ش = sh, خ = kh, ذ = dh, ث = th) Qalam is a system of Arabic-Latin-Arabic transliteration between Arabic writing and the Latin alphabet embodied in ASCII (American Standard Code for the exchange of information). Even though, Qalam presents some limitations because some words are written differently in both cases, eg "FOORT" and "فور", transliteration of "فور" gives us 'FOR' and not 'FORT'.

4.8. Phonetic Grouping: This step has two essential parts: i) unification of letters and ii) Phonetic group-

[1]https://www.nltk.org/

[2]https://snowballstem.org/

[3]http://snowball.tartarus.org/algorithms/porter/stemmer.html

[4]https://www.arabicstemmer.com/

[5]http://qalam.info/

ing. The process begins with the substitution of numbers, because several words in AlgD contain numbers that are pronounced as letters for example (kh=5, h=7, k=9, ou=2, t=6). After unifying the words containing numbers, the process starts the phonetic grouping. The idea is to exploit phonetic regency to maximize its use in the categorization of comments, resulting a resource that can be used later to correct new texts. Our aim is to group words that are pronounced in the same way, for example: ['Insh'allah', inch'allah', 'in'cha'alla] At the end of this step, a phonetic dictionary is created where each word is associated to a list of its corresponding phonetic codes. Subsequently, this dictionary will be used to substitute certain words by their most common form in the dictionary. For example ['Insh'allah', inch'allah', 'in'cha'alla'] are associated to ANCHALA. The substitution by the most common form participates to reduce the noise generated by the diversity of scripts for the same word in Algerian dialect.

4.9. Removing Vowels: After analyzing how users on social networks interact and communicate through messages and comments, we realized that the majority of users write dialect words regardless of any grammatical or spelling rules, especially when writing vowels. For example, they do not differentiate between the 'a' and the 'e', example: 'nhabak' or 'nhebek'.

Because vowels are not informative contrary to consonants, we propose in our process to delete all vowels from a given word written in Latin, in order to fill the absence of stemmers for the Algerian dialect. for example: 'nchallah, inchaleh' are associated to "NCHL". This step allows reducing the size of the corpus vocabulary by grouping words that represent the same instance.

5. Data Presentation: In this phase, the process transforms the documents (treated comments and messages generated from the previous steps) into vectors in order to be understandable by the classifier, using 3 vectorizers :CountVectorizer [6] , TF vectorizer (term frequency), TF-IDF vectorize (term frequency – inverse document frequency)[7]

6. Classification: In this phase, the prediction model is created, based on a supervised classification, using the annotated and treated corpus. To accomplish this task, we used the most popular supervised classification algorithms in the natural language processing tasks, which are :Naive Bayes, Support vector machine and Decisional trees For this, the dataset is decomposed into training subset,test subset and validation subset. The model should be parametrized in order to find the best combination of parameters giving the best score for each algorithm.

[6]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[7]https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

4 EXPERIMENTATIONS & RESULTS

In this section, we will present our experimentation methods and obtained results.

4.1 Data Collection

The lack of Algerian datasets in literature, motivates us to propose 4 methods to collect Algerian comments from social networks. To this end, we followed these ways:

1. Consume an existing Algerian dataset "Wacht7ass" created in 2016 (Mataoui et al., 2016). It contains 12.612 comments retrieved from different Algerian Facebook pages. The comments are written in Latin or Arabic characters.

2. Collect comments & posts from Facebook pages on which we are admins like BrandtDZ[8]. We have written a python program based on Facebook Graph API[9] to extract comments and posts published in these pages. We gathered about 2.569 Algerian documents since January 2018 until April 2019.

3. Publish a Google Form that we created in order to collect comments with their polarities (positive, negative or neutral). Our aim was to collect maximum of annotated data from Algerian people, for this we sponsored our form in many facebook pages and we collected around 1.400 annotated comments.

4. Use Facepager[10] open source application designed to collect public data from social networks even without any access token. All data are stored in a SQLite database and can be exported into csv files. Basing on this tool, we have retrieved more than 20.000 comments from different Algerian Facebook pages like: Cevital, Jumia, El Bilad, Djezzy, etc. Majority of text is written in Arabic characters. This dataset is used for tests.

More details about these datasets are given in table 1.

For language detection, we have written a python program to automate this step. while, we used Alphabet detector[11] python library to detect Latin and

[8]www.facebook.com/BRANDT.DZ

[9]https://developers.facebook.com/docs/facebook-login/access-tokens/

[10]https://github.com/strohne/Facepager

[11]https://pypi.org/project/alphabet-detector/

Table 1: Dataset sizes.

	Wach-t7ass	Google form	BrandtDZ page
NB documents	12.611	1.400	2.569
NB characters	312.396	37.080	816.511
NB words	68.986	7.390	179.271

Arabic characters of each comment. The repartition of these characters are resumed in Fig. 3, most comments are in Arabizi.

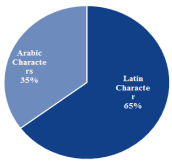

Figure 3: Comments characters.

4.2 Data Annotation

To achieve this step, we used crowd sourcing method. For the dataset gathered from google form, it was annotated by default as explained before. While, for the other datasets we used an open source platform developed for annotation "DOCCANO" [12] where we defined our labels (positive, negative and neutral) in order to help the annotators group composed of 22 people. Each two people work together on the same comments for more reliability. The results of the data annotation are given in Fig. 4

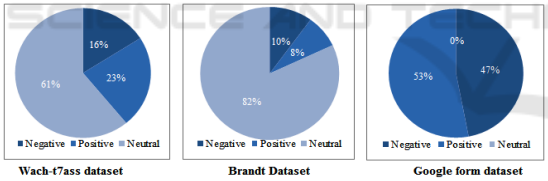

Figure 4: Data annotation results.

4.3 Data Preprocessing

As already explained, the aim of this step is to get a valid and clean dataset with maximum informative vocabulary and without any redundancy. During the life cycle of a natural language processing project, more than 50 % of time is spent on this step, which gives us insights about its importance. The results of preprocessing are presented in figure 5.

After the deduplication of comments and removing of tags, we pass from 29.422 to 16.785 comments.

We notice that the majority of comments are tags and they are duplicated that's why it's important to deduplicate them.

[12]https://github.com/chakki-works/doccano

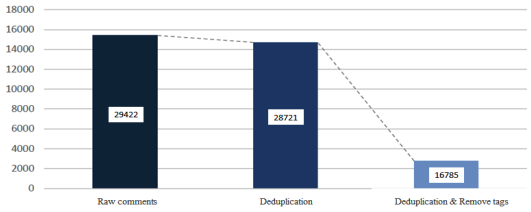

Figure 5: Results of deduplication.

We also used CLTK [13] for transliteration and Phonetics[14] for phonetic grouping. At the end of the preprocessing pipeline, the process generates a corpus of 23.618 unique words. The following barplot (Fig. 6) shows the reduction of the vocabulary size (dimensionality reduction), where:
Basic preprocessing = tokenization + remove noise + remove isolating single letters + Treatment of capitalized words + Dis-accentuation.

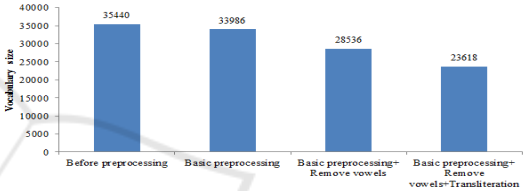

Figure 6: Vocabulary before and after preprocessing.

We thus reduced the dimensionality of our model by 33 %, we pass from 35.440 to 23.618 features. This reduction is due to grouping words representing the same instance and deleting meaningless words such as stop words. The barplots in Fig. 7 represent most frequent words among the generated corpus "Before pre-processing" and "After pre-processing" .

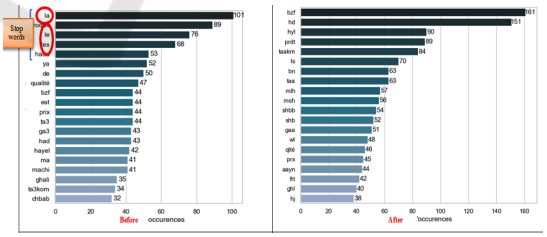

Figure 7: Most frequent words before & After preprocessing.

Before Preprocessing: This part represents the most frequent words in our corpus before pre-processing. We clearly notice the empty words being at the top of the ranking, in addition to the presence of two words that represent the same instance, "hada" and "had".
After Preprocessing: This part represents the most frequent words in the corpus after the pre-processing.

[13]http://cltk.org/
[14]https://pypi.org/project/phonetics/

We notice that the list is represented mostly by adjectives which describe an opinion (hyl, bn, mlh, etc). We also notice that the word "bzf", meaning "a lot", occured 44 times before pre-processing and 161 time after pre-processing. This means that all the words similar to "bzf" are grouped (bezaf, bazaf, baazef, bezzaaf, etc.)

4.4 Presentation and Classification

In order to evaluate our model, we followed the strategy described in section 'Classification'. In what follows:

- Type of preprocessing used is: (S : Data without pre-processing ; P : Data with basic preprocessing (noise removal); PV : P + Vowels removal; PVT : PV + Transliteration).
- Data Presentation (count, TF, TF-IDF)
- Classification algorithm (Naive Bayes, SVM : Support Vector Machine, DT : Decision trees)

For these tests, the data used are the combination of all collected corpuses (Google Form, Wach-t7ass, BrandtDZ dataset) to get a voluminous dataset. We used Cross validation where the dataset is divided into training subset, test subset and validation subset. The model should be parametrized aiming to find the best combination of parameters giving the best score for each algorithm using Grid-Search. For evaluating our model, we used two appropriate metrics: Accuracy metric and the F-score.

Our tests and experimentations showed that SVM outperforms other algorithms and returns the best scores even accuracy or F-score. As detailed in figure 8, the SVM model reaches 83,28 % with TF-IDF representation and basic presprocessing + vowel removing. We argue that the preprocessing specially the added steps of removing vowels and phonetic grouping are really important for enhancing the scores. while the transliteration has a big impact on creation of consistent vocabulary.

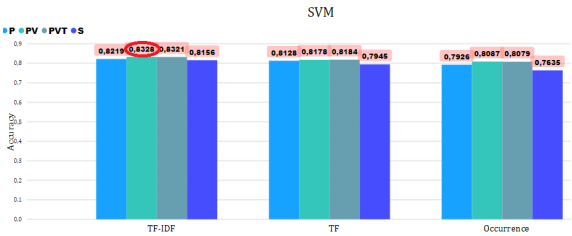

Figure 8: SVM accuracy results.

Even, in figure 9 the obtained results according to F-score support our previous ones; where we reach 87,06 % with occurence representation. We notice also, that values of F-score and accuracy are close, this confirms that the obtained results are effective and there is no overfitting phenomenon.

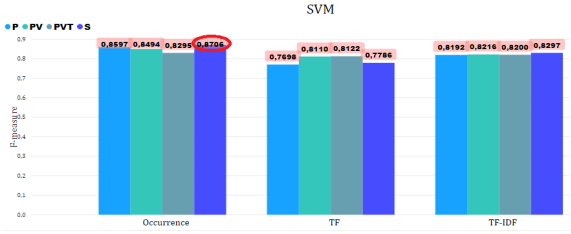

Figure 9: SVM F-measure results.

Figure 10 generated with matplotlib is a 3-dimensional array that describes the variation of the accuracy and F-score of the model according to the type of processing , the type of algorithm and the type of representation.

To simplify the presentation of our results we have converted our 3d arrays to 2d arrays by joining the third dimension (type of pre-processing) on the second dimension (classification algorithm). Our strategy is to use multiple combination of the 3-uples.

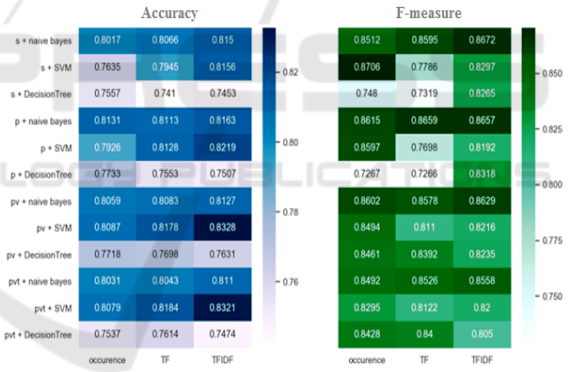

Figure 10: Classification models evaluation.

We note that the value of accuracy varies from 76 % to 83 % and the value of F-score from 76 % to 87 %. Among the algorithms, we note that Decision Tree gives the worst results compared to other algorithms. We also notice the stability of the results given by Naive bayes according to the type of pre-processing. However, as explained before, SVM gives better accuracy score with the TF-IDF representation (83 %), and better F-score with the occurrence representation (87 %). Finally we notice that the new preprocessing steps have a positive impact on the value of accuracy and F-score. The treatment of dialect text allowed us to lift the F-score of SVM to 87 % which is considerable. The added preprocessing steps : Transliteration, vowels removing and phonetic grouping have an impact in enhacing the results

but also a big importance in constructing the Algerian vocabulary.

For more details, see table2 , which summarises the results of each dataset individually.

Table 2: Best classification results of each dataset.

		Accuracy	F-measure
wacht7ass	PV+SVM+TF-IDF	81,63 %	84,47 %
G-Form	PVT+Naive bayes+TF-IDF		80,87 %
G-Form	PVT+SVM+TF	78,32 %	
Brandt	PVT+SVM+TF-IDF	94,24 %	90,67 %

For instance, some results labled Facebook comments are given at https://drive.google.com/file/d/1oFmoETRYys8ZHjcZcQCZqubIJ3Ceex66/view?usp=sharing

5 CONCLUSIONS

In this paper, we presented a supervised approach for sentiment analysis in Algerian dialect written in Latin script, which gave interesting results despite the many specific aspects of the dialect and complexity of Arabizi analysis. We report results from an extensive empirical evaluation assessing the effects of classifiers, the effects of presentation types (count, TF, TF-IDF) and those of novel contributions in preprocessing phase, notably, vowels removing. Three data sets were annotated with their respective sentiment labels using crowdsourcing in this experiment. We achieved an F-score of 87 % and an accuracy of 83 % using this approach. Results revealed also that SVM outperforms the other classifiers. Finally, the preprocessing allowed us to impove f-score of SVM by 9,20 %, which is considerable and shows the relevance of our prior premises.

Our work can be improved in various directions. First, we will test other models (random forest, gradient-boosted trees, Latent Dirichlet Allocation model). We could also explore other characteristics and feature such as emoji interpretation and Irony/Sarcasm detection or other areas of opinion mining field, notably, subjectivity analysis and rumor detection.

REFERENCES

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014). Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

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

Al-Ayyoub, M., Essa, S. B., and Alsmadi, I. (2015). Lexicon-based sentiment analysis of arabic tweets. *IJSNM*, 2(2):101–114.

Al-Ayyoub, M., Khamaiseh, A. A., Jararweh, Y., and Al-Kabi, M. N. (2019). A comprehensive survey of arabic sentiment analysis. *Information Processing & Management*, 56(2):320–342.

Ali, C. B., Mulki, H., and Haddad, H. (2018). Impact du prétraitement linguistique sur l'analyse des sentiments du dialecte tunisien. In *Actes de la conférence Traitement Automatique de la Langue Naturelle, TALN 2018*, page 383.

Badaro, G., Baly, R., Hajj, H., Habash, N., and El-Hajj, W. (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.

Bayoudhi, A., Ghorbel, H., Koubaa, H., and Belguith, L. H. (2015). Sentiment classification at discourse segment level: Experiments on multi-domain arabic corpus. *JLCL*, 30(1):1–24.

Bettiche, M., Mouffok, M. Z., and Zakaria, C. (2018). Opinion mining in social networks for algerian dialect. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 629–641. Springer.

Cherif, W., Madani, A., and Kissi, M. (2015). Towards an efficient opinion measurement in arabic comments. *Procedia Computer Science*, 73:122–129.

Hadi, W. (2015). Classification of arabic social media data. *Advances in Computational Sciences and Technology*, 8(1):29–34.

Ibrahim, H. S., Abdou, S. M., and Gheith, M. (2015). Mika: A tagged corpus for modern standard arabic and colloquial sentiment analysis. In *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*, pages 353–358. IEEE.

Mataoui, M., Zelmati, O., and Boumechache, M. (2016). A proposed lexicon-based sentiment analysis approach for the vernacular algerian arabic. *Res. Comput. Sci*, 110:55–70.

Medhaffar, S., Bougares, F., Estève, Y., and Hadrich-Belguith, L. (2017). Sentiment analysis of tunisian dialects: Linguistic ressources and experiments. In *Proceedings of the third Arabic natural language processing workshop*, pages 55–61.

Medhat, W., Hassan, A., and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

Mustafa, H. H., Mohamed, A., and Elzanfaly, D. S. (2017). An enhanced approach for arabic sentiment analysis. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 8(5).

Zarra, T., Chiheb, R., Moumen, R., Faizi, R., and Afia, A. E. (2017). Topic and sentiment model applied to the colloquial arabic: a case study of maghrebi arabic. In *Proceedings of the 2017 international conference on smart digital environment*, pages 174–181. ACM.