Deception Detection in Arabic Tweets and News

F. Javier Fernández-Bravo Peñuela

Polytechnic University of Valencia, Spain

Abstract. The project Arabic Author Profiling for Cyber-Security (ARAP)[1] aims at preventing cyber-threats using Machine Learning. To this end, they monitor social media to early detect threatening messages and, in such a case, to profile the authors behind. Profiling potential terrorists from messages shared in social media may allow detecting communities whose aim is to undermine the security of others. One of this framework's main challenges is recognizing false positives, such as potential threatening messages that are actually deceptive, ironic or humorous. This paper focuses on the goal of detecting deceptive messages, which are intentionally written trying to sound authentic. This task is performed on two different genres of Arabic texts: Twitter messages and news headlines.

Keywords: deception detection, text classification, natural language processing

1 Introduction

Author profiling is a research field of growing interest within the area of Natural Language Processing. It enables to deduce some characteristics and traits about an individual by just analyzing a text written by her. In a world of social media and fake news where millions of anonymous users can spread and disseminate messages which hold the potential to influence the masses, the possibility of determining a person's gender, age, native language, or even language variety enables a way to infer the author profile behind theses messages. This allows to identify security issues, such as threatening messages, and the nature of the individual or organizations responsible for them.

The coming of such techniques reveals itself as specially useful in territories in conflict or regions suffering of warfare, aiding in the task of identifying supporters or members of terrorists groups and parties, so that attacks and other violent acts can be prevented and thus avoided. [10]

In the case of Arabic-speaking countries, this constitutes a particular difficulty, due to the existence of a notable lack of research in the field of Natural Language Processing on the Arabic language [9], since most of the effort until now has been dedicated to the English language. The fact that the Arabic written language lays on its own alphabet and does not fit the left-to-right direction of writing common in the occidental countries provides additional problems, all

[1] http://arap.qatar.cmu.edu

of which must be handled in order to attain a proper analysis and processing of texts written on the Arabic language or any of its dialects.

Furthermore, another interesting challenge arises in the problem of determining the truthfulness of a statement just from the way it is written and expressed. This entails classifying a message as true or deceptive (a false statement which is intentionally written pretending to seem true) [5], taking into account linguistic twists which may imply false positives, such as irony or humor [9]. All these constructions must be detected and handled for the sake of the system's accuracy. This paper focuses on the goal of detecting deceptive messages, which are intentionally written trying to look authentic. This task is performed on two different genres of Arabic texts: Twitter messages and news headlines.

The present work describes the process of designing and implementing a classifier whose goal is to decide whether a message is either true or deceptive, based on the application of Natural Language Processing techniques for decomposing the text and arranging it in the form of a vector of features, along with the construction of a Machine Learning model trained upon two different datasets of Arabic written texts: Twitter messages and news headlines. This task is carried out in the context of the APDA challenge [7] held in conjunction with the FIRE 2019 Forum for Information Retrieval Evaluation.

This document first outlines the aforementioned challenges and difficulties found in the problems of author profiling and deception detection, focusing on their application in the analysis of Arabic written texts. Next, it focuses on how to deal with these problems, detailing how each one of them was overcome in the design of the proposed classifier for detecting deceptive messages: which techniques and features were applied on the corresponding stages of development, which were considered but later discarded for different reasons, and which were retained and assimilated in the final model. Last, some conclusions regarding the attempt in providing a solution to the problem of deception detection, with its assorted eventualities and the results reached are highlighted.

2 Challenges

This section addresses three main challenges found when analyzing written texts from Arabic media (both press and social). First of all, how these texts' structure can be processed in the context of an automated task, emphasizing how its differences to languages based on the Latin alphabet can be handled. Next, the result of this process is applied to the problems of author profiling and the more definite problem of deception detection, which are briefly described.

2.1 Analysis of Arabic written texts

Advances in author profiling are constrained by the availability of representative training data, since having a large amount of tagged data remains necessary to build a Machine Learning model capable of attaining reliable results. Collecting this data requires a huge effort for retrieving raw text and annotate it by means of human operators. Due to a lack of research efforts targeting author profiling on the Arabic language [3], most of the existing corpora are intended to English or other European languages, all of them widely different from Arabic regarding their morphology and syntax. [9] [11]

The complexity of the Arabic language at the various levels of its linguistic representation (phonology, orthography, morphology, and syntax) makes a challenging task the work of building Natural Language Processing applications and tools targeting it [13]. Arabic morphology uses prefixes, infixes, and suffixes, not only for inflection but also to concatenate words. Moreover, Arabic is a language with clear diglossia, which causes its spoken form to be quite different from the written form of the language [9]. While modern standard Arabic has a clearly defined set of orthographic standards, the various dialects historically related and which co-exist with it have no official orthographies, which enables a given sentence or event the same word to be written and spelled in multiple ways in different Arabic dialects. [12]

A great advance in the processing of the Arabic language is the Arap-Tweet dataset, a large-scale and multi-dialectal corpus of tweets from eleven regions and sixteen countries in the Arab world representing the major Arabic dialectal varieties, which takes advantage on the opportunity that Twitter offers to gather large amounts of informal language texts from many individuals, whose main traits can be retrieved from the API Twitter itself provides. The data provided by the corpus has been collected, processed, normalized and annotated according to their dialectal variety, the gender of the user and the age within three categories, making it an invaluable resource for further author profiling research on Arabic, based on these and other features, as well as the development of new Natural Language Processing tools. [13]

2.2 Author profiling

Author profiling enables to deduce some characteristics and traits of an individual just by the analysis of her written texts. The linguistic profile of a person having written a text allows us to provide some valuable background information about its author, such as the demographic characteristics of the individual as well as her cultural and social context. [9]

In the case of the Arap-Tweet dataset [13], its authors focused on annotating each data entry with four separate labels related to different traits of the author: gender, age, Arabic dialect, and whether the user is a native Arabic speaker or not. Also, since guessing the age of an individual is not a task easy at all, age intervals are established, where individuals are classified in one range by the corpus dataset taggers. When constructing a Machine Learning model, the corresponding interval is often calculated by means of probability distributions, choosing the slot that maximizes this probability.

The Arabic language used in social and online media is a mix of MSA (*Modern Standard Arabic*) and other regional dialectal varieties. The variation from one region to another poses many challenges to Natural Language Processing applications, which must recognize this situation when studying online texts written in Arabic. Although there are many similarities amidst the Arabic dialects, there is often a difference at the levels of lexicon, morphology and phonology.

In the case of the ARAP project, this technique is used for developing profiling resources and tools that can be exploited in the context of cyber-security for profiling potential terrorists from messages shared in social media, with the clear proposal of detecting communities whose aim is to undermine the security of others. More specifically, author profiling in the context of the mentioned project could be useful for forensic investigations to narrow the set of potential authors of a threatening message. [12]

2.3 Deception detection

A deceptive opinion can be defined as a fictitious opinion with the intention to sound authentic in order to mislead the reader [9], usually a short text written by an unknown author using a not very well-defined style [8]. In this case, based on the analysis of texts, indicators of deception can not be obtained from physiological and gestural behaviors, and all assessment must forcibly come from written expression. Cagnina and Rosso [2] detail some Natural Language Processing and Machine Learning techniques which pose particularly appropriate for detecting opinion spam.

The framework at the ARAP project addresses deception detection in Arabic in order to detect potentially threatening message profiling their authors and discarding those messages that do not really represent potential threats [5]. Of course, this implies becoming aware of false positives, such as apparently threatening messages that are actually deceptive, ironic or humorous, as displayed on the workflow show on Figure 1 on the next page.

This implies real-time retrieval and analysis of such messages and their authors. Due to the lack of valid tagged and annotated training data, its authors opted for a language-independent approach, proposing the *Low Dimensionality Statistical Embedding* (LDSE) [6] to represent documents on the basis of the different use of the words depending on the classification classes available. The key concept is a weight representing the probability of a term to belong to one of the classes into which be classified (in this case, credible and non-credible, when the user lies or not). So, the distribution of weights for a given document should be closer to the weights of its corresponding class.

This approach was verified to be suitable for deception detection in Arabic and competitive with existing approaches based on word embeddings, such as Continuous Bag of Words and Skip Grams.

Apart from the Credibility Analysis of Arabic Content on Twitter (CAT), known as the Credibility corpus [4], two new corpora were also created in the mentioned work: the Qatar Twitter corpus (tweets referring to the Qatar Blockade and the Qatar World Cup) and the Qatar News corpus (short contents such as headlines and/or excerpts from well-known Arabic newsletters); both of them were used in the present work in order to perform deception detection on Arabic texts of various natures.

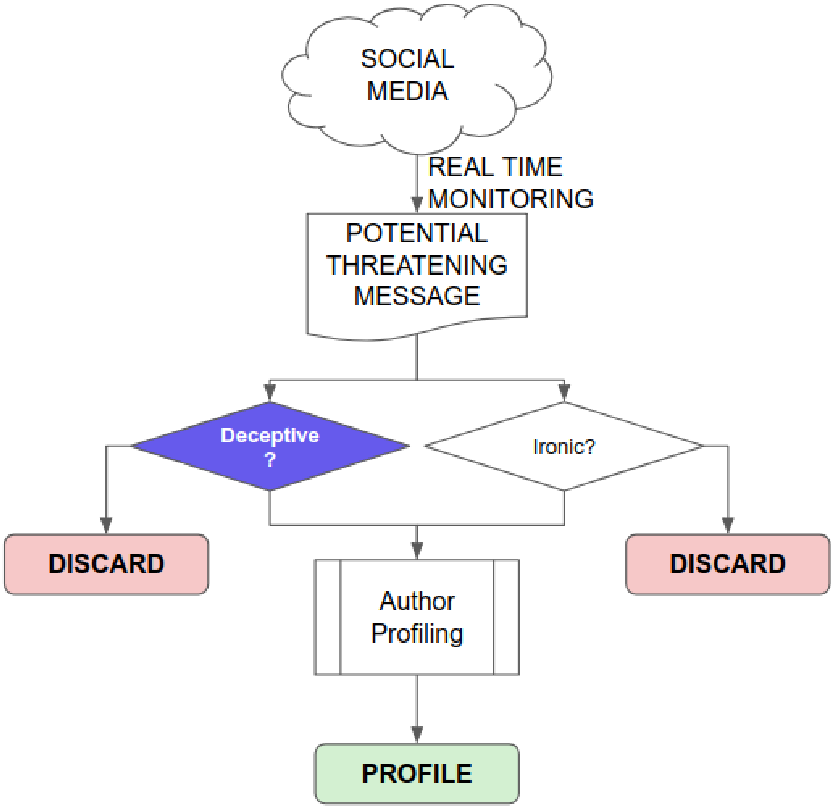

Fig. 1. Workflow of the Arabic author profiling for cyber-security project. Extracted from [5]

3 Deception detection in Arabic written texts

The focus of the task is on deception detection in Arabic on two different genres: Twitter and news headlines. Both dataset's origin are the corpora created in [5], which contain 1444 news headlines (679 true statements and 765 deceptive statements) and 532 Twitter messages (259 true publications and 273 deceptive publications). They reach a large variety of topics and both classes are balanced enough, so that they are supposed to be representative of the whole populations generalizable by the experiment.

The documents contained in each dataset must be transformed into representation which allows its processing, such as a vector of features. A Machine Learning model will be constructed and trained using the portion of the dataset available to the APDA challenge contestants [5]. This model will be used for making classification predictions for the texts in the evaluation portion of the dataset. Finally, the test results' quality will be measured by computing the F-Score which compares the classification predictions to the actual classes for the evaluation data subset.

3.1 Approach to Natural Language Processing from raw text

As referred in Subsection 2.1, the complexity of the Arabic language at the various levels of its linguistic representation makes a challenging task the work of building Natural Language Processing tools and applications targeting it. [13]

When arranging a Natural Processing pipeline for a collection or stream of written texts, usually the first stage is that of normalization. The most common technique to be applied is converting text to lowercase before doing anything with its words, so that the distinction amidst differently capitalized words is ignored. Another normalization task involves identifying non-standard words, including numbers, abbreviations, and dates, and to map any such tokens to a special vocabulary. This helps to control the growth of the vocabulary and improves the accuracy of many language modeling processes.

One way to go further is to strip off any affixes, a task known as stemming, and another beyond further step is to make sure that the resulting form is a known word in a dictionary, a task known as lemmatization. Once again, the application of these last two techniques is dependent on the availability of the necessary linguistic resources for the corresponding language. Two of the most popular stemmers are the Porter and Lancaster's ones, but they are intended only to the analysis of the English language. Even in this case, stemming is not a well-defined task. [1]

After the normalization stage, the next fundamental task usually applied is tokenization, the process of cutting a string into identifiable linguistic units that constitute a piece of language data. The very simplest method for tokenizing text is to split on whitespace, but usually other splitter characters are also used, such as newline characters, tabs, or any concatenation of them. Often a better approach is to make use of regular expressions and, instead of selecting splitters, select every full sequence of alphanumeric characters as a single word, thus covering a wider range of cases, stripping punctuation characters, and just keeping the relevant portions of the string.

No single solution works well when approaching the tokenization task, and it is critical to decide what counts as a token depending on the application domain. An additional issue is the presence of contractions, which would probably be better normalized into the different words composing the comprised form. In this case, the existence of tokenizers supporting the Arabic alphabet reveals itself as a critical matter. Another additional issue is the (own) lack of knowledge about the Arabic alphabet and language, which avoids to take advantage from any form of expert judgment when evaluating the tokenization results.

A subsequent stage is tagging: the process of classifying words into their lexical categories, also named "part-of-speech tagging" or "POS tagging" [1]. Usually tokens in a tagged corpora are represented using a tuple consisting on the token and the tag. Of course, the process of automatically tagging the tokens composing a corpora into nouns, verbs, adjectives, adverbs, and other grammatical classes depends on specific support for the language by the Natural Language Processing toolset used, which must provide lookup tables and dictionaries which support homonyms associated to different grammatical categories. A more advanced approach is the application of n-gram tagging, which does not consider just the current token in isolation, but the larger context of the part-of-speech tags of the preceding tokens. The analysis for determining the most probable categories relies on morphological, syntactic, and semantic clues. Once again, the existence or absence of tools supporting the target language poses crucial, along with the ability to introduce expert knowledge while the implementation of these techniques is still being tuned.

The next stage, and the overall goal of this work, is classification, the task of choosing the correct class label for a given input. In this case, the task is to build a supervised classifier, based on training corpora containing the correct label for each input. During training, a feature extractor is used to convert each input value to a feature set. Pairs of feature sets and labels are fed into the Machine Learning algorithm to generate a model. During prediction, the same feature extractor is used to convert unseen inputs to feature sets. These feature sets are then fed into the model, which generates predicted labels.

The first step is creating a classifier is deciding what features of the input are relevant, and how to encode them. Selecting relevant features and deciding how to encode them for a learning method can have an enormous impact on the learning method's ability to extract a good model. There are usually limits on the number of features that should be used with a given learning algorithm, since providing too many features could produce the phenomenon known as overfitting, where the model generated from the training data does not generalize well to new examples. Successive evaluations of the error by dividing the development dataset into two subsets (training and dev-test) are a good way to refine the set of features selected, especially when combined with data partitioning methods like cross-fold validation, that helps to obtain a generalizable model which does not suffer of overfitting.

Once a feature extractor for documents has been defined, so that the classifier will know which aspects of the data it should pay attention to, such a classifier can be fed with training data later and used to label new entries. Three of the most common classifiers to be used are decision trees, naive Bayes classifiers, and support vector machines. Decision trees are often fairly easy to interpret when the set of features is small, since they generate conjunctions of conditional rules. Naive Bayes classifiers assign likelihood estimates to each label, computed by weighting the input features values. Support Vector Machines, in addition to linear classification, can also perform a non-linear classification by implicitly mapping their inputs into high-dimensional feature spaces. When part-of-speech tagging is used, Hidden Markov Models act like consecutive classifiers and typically reach good results. [1]

Finally, in order to decide whether a classification model is accurately capturing a pattern, how trustworthy that model is and for what purposes it can be used, it must be evaluated. Mos evaluation techniques calculate a score for a model by comparing the labels that it generates for the inputs in an evaluation set with the correct labels for those inputs. However, it is very important for the test set being distinct from the training corpus, since reusing the training set for evaluations would be misleading, would not show the model's ability to generalize, and could be masking severe overfitting.

The proportion of the development dataset which is dedicated to training and the proportion dedicated to evaluation must be carefully delimited, and the total data must be shuffled to increase entropy (as long as there are no sequential nor temporal relations between the data entries which should be taken into account when building the model), and it is often a good idea to apply techniques such as cross-fold validation, as mentioned beforehand, especially if the training dataset is not very large.

Measures like accuracy, precision, and recall are commonly used, but it is often a better idea to use a measure like F-score, which is defined as the harmonic mean of precision a recall, both combined to give a single score. Confusion matrices are usually quite illustrative in case the number of possible classes remains small.

3.2 Walkthrough of the task resolution

In order to process the raw text collection from the dataset and build a classifier capable of differentiate whether tweets and news are true or deceptive, the first step was to apply some normalization tasks on the text entries, as follows:

- All letters capitalization was turned to lowercase.
- Numbers were removed.
- White spaces and other splitter characters were first collapsed and then removed.
- Every words contained into the English and Arabic languages stop-words lists of common words with empty semantic meaning were removed from the text.
- Punctuation symbols were removed.
- Words from the English and Arabic languages stop-words lists were removed again. The reason for performing this elimination twice was having found out that removing words from the stop-words list before and after removing punctuations actually contributed to a better cleansing of the text processed, thus increasing the final accuracy attained by the classifier.

Next step was to retrieve a data frame containing the 1000 most frequent words occurring in the text collection. This data frame, which corresponds to the text collection vocabulary, was used to generate the bag of words representation of the text collection, a two-dimensional matrix which relates words from the vocabulary to their number of occurrences on each of the dataset's documents.

This bag of words is actually a vectorized representation of the text's features, which, along with the class each document belongs to, can be used to train a classifier a build a Machine Learning model with the ability to decide the most probably category for new unseen documents.

At this point, the bag of words was enriched in different ways for the news and Twitter datasets (constructing separated classifiers for every one). A new feature was added to both data frames containing the number of words in the document, and three new features were added to the Twitter data frame detailing the number of *hashtags*, user mentions, *emojis*. These three are some traits characteristic in Twitter's texts and it is believed that their frequency might be related to either true or deceptive messages, so that they make a good discriminatory factor for differentiating and classifying messages.

Once the vectorized form of the text collection is complete, a classifier could be trained. Before performing this task, the probability distribution for both classes (true and deceptive) was computed from the training data, in order to determine whether the classes, which are supposed to be representative for the general population, are balanced. They were actually found to be balanced, so no additional action was performed for this reason. However, in case of having training data whose population has unbalanced tag categories, weighting mechanisms could be included in the classifier to compensate this situation. These weighting mechanisms can also be used if the penalization for erroneous classification differs between classes, so that the classifier will take this into account when predicting and it will not be unconditionally slanted to the most probable class.

The data frame containing the bag of words representation for each dataset was used to train a classifier, taking Support Vector Machines as the Machine Learning technique of choice, due to their good performance on classification where many features are used (as it happens to be the case of text classification). K-fold cross-validation was used to select the classifier which provides the highest accuracy and displays the best ability to generalize. The collection was split into four folds, three of which were used for training on each iteration, while the other one was left for evaluation.

Since the result classes for the test dataset are unknown, the F-Score obtained by the implementation of the k-fold cross-validation technique was used to check which combination of parameters, features, and techniques produced the highest accuracy on classification. This accuracy result data from evaluation are displayed in Section 4.

Once the classifier model was build, the bag of words representations for the test datasets (news and tweets) were constructed, applying on them the same normalization techniques on the documents as in their documents as in the training data and using the vocabulary from the training stage. By doing this, both vectorized representations are equivalent to the training ones regarding order and indexing, and the classifier build from training data can be used to generate predictions for new data.

Finally, predictions were generated for the vectorized representation of the test data, and the identifier and predicted class for each document were stored into text files, ready to be submitted for evaluation. Since different models were trained for classifying news and tweets, the corresponding classifier was used for generating predictions on test data files.

4 Results

First of all, a prototype implementation was built, including just the components strictly needed for basic tokenization, vectorization, and classification (without including other manipulations on the text neither the extraction of additional features), so that a baseline score could be obtained and later improved. a F-Score of 0.58 was reached on the news dataset, while a score of 0.61 was reached in the case of the Twitter dataset, both of them being quite low for a classification with just two possible categories

Once the definitive implementation was completed, having tuned the Support Vector Machine's hyperparameters, extracted additional useful features and included supplementary mechanisms (which were commented on Section 3.2), the F-Score obtained improved to 0.70 on the news dataset and 0.77 on the Twitter dataset.

Table 1. Classification results on news and tweets

Model	**News**	**Twitter**
Baseline	0.58	0.61
Intermediate model	0.70	0.64
Final model with ad-hoc Twitter features	0.70	0.77

Although an even higher improvement would be feasible, the results reached show that the application of some of the techniques introduced actually improves the classifier's accuracy. In the case of classification on the Twitter dataset, it was evaluated before implementing the extraction of relevant features (just keeping the n most frequent words), reaching a F-Score of 0.64. The inclusion of some relevant features based on the text characteristics themselves (number of *hashtags*, user mentions...), combined with the common Natural Language Processing techniques, provides a significant improvement on the results attained, which were raised to the final 0.77 on this dataset.

5 Conclusions

The present work has described the process of designing and implementing a classifier whose goal is to decide whether a message is either true or deceptive. The tasks of author profiling and deception detection have been analyzed, also detailing the additional difficulties found when dealing with Arabic written texts.

The process of building the mentioned classifier has been addressed, first from the perspective of decomposing, transforming, and arranging the text into a vector of features via Natural Language Processing techniques, and next continuing across some Machine Learning methods capable of being trained with the data resulting from the first stage and building a model which can be used to carry out predictions on new data entries. Last, the results obtained by the classifier have been displayed, detailing how the importance of the set of features extracted from the text can indeed be decisive on these results.

References

1. Bird, S., Klein, E., Loper, E.: Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc." (2009)
2. Cagnina, L., Rosso, P.: Classification of deceptive opinions using a low dimensionality representation. In: Proceedings of the 6th workshop on computational approaches to subjectivity, sentiment and social media analysis. pp. 58–66 (2015)
3. Celli, F., Pianesi, F., Stillwell, D., Kosinski, M.: Workshop on computational personality recognition: Shared task. In: Seventh International AAAI Conference on Weblogs and Social Media (2013)
4. El Ballouli, R., El-Hajj, W., Ghandour, A., Elbassuoni, S., Hajj, H., Shaban, K.: Cat: Credibility analysis of arabic content on twitter. In: Proceedings of the Third Arabic Natural Language Processing Workshop. pp. 62–71 (2017)
5. Rangel, F., Charfi, A., Rosso, P., Zaghouani, W.: Detecting deceptive tweets in arabic for cyber-security
6. Rangel, F., Franco-Salvador, M., Rosso, P.: A low dimensionality representation for language variety identification. In: International Conference on Intelligent Text Processing and Computational Linguistics. pp. 156–169. Springer (2016)
7. Rangel, F., Rosso, P., Charfi, A., Zaghouani, W., Ghanem, B., Sanchez-Junquera, J.: Overview of the track on author profiling and deception detection. In: Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019) (2019)
8. Rosso, P., Cagnina, L.C.: Deception detection and opinion spam. In: A Practical Guide to Sentiment Analysis, pp. 155–171. Springer (2017)
9. Rosso, P., Rangel, F., Farías, I.H., Cagnina, L., Zaghouani, W., Charfi, A.: A survey on author profiling, deception, and irony detection for the arabic language. Language and Linguistics Compass **12**(4), e12275 (2018)
10. Russell, C.A., Miller, B.H.: Profile of a terrorist. Studies in conflict & terrorism **1**(1), 17–34 (1977)
11. Zaghouani, W.: Critical survey of the freely available arabic corpora. arXiv preprint arXiv:1702.07835 (2017)
12. Zaghouani, W., Charfi, A.: Arap-tweet: A large multi-dialect twitter corpus for gender, age and language variety identification. arXiv preprint arXiv:1808.07674 (2018)
13. Zaghouani, W., Charfi, A.: Guidelines and annotation framework for arabic author profiling. arXiv preprint arXiv:1808.07678 (2018)