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: text classification, deception detection, arabic text mining, natural language processing

1 Introduction

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 [2] 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 text mining 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, where its assorted eventualities and the results reached are highlighted.

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

2 Challenges

Three main challenges are found when analyzing written texts from Arabic media (both press and social). First of all, differences to languages based on the latin alphabet and with widely different from the gramatical point of view must be handled, so that texts written on these languages can be processed in the context of an automated task.

Next, the result of this process in the current task must be applied to the problem of deception detection, which will be addressed below in section 3.

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 [1], 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 has to be constructed and trained using the portion of the dataset available to the APDA challenge contestants [1]. 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.

The tasks carried out to reach this goal are outlined in the following subsections.

3.1 Preprocessing

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 word contained into the English and Arabic languages stop-words lists of common words with empty semantic meaning was 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.

3.2 Feature extraction

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.

3.3 Classifier model training

Once the vectorized form of the text collection was complete, the data frame containing the bag of words representation for each dataset was used to train a classifier[2], 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 every 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.

[2] In case of having unbalanced classes, weighting and penalization mechanisms can be included in the classifier, so that the predictions made will be more accurate and representative for the general population.

3.4 Accuracy evaluation

Once the classifier model was built, 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 built 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. The original F-score reached on both datasets was quite low for a classification with just two possible categories. These results, along with the attained in further stages, are displayed in table 1.

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

Once the definitive implementation was complete, having tuned the support vector machine's hyperparameters, extracted additional useful feature and included supplementary mechanisms (which were commented in subsection 3.2), classification on the Twitter and news datasets respectively improved in 16% and 12%.

Although an even higher improvement would be feasible, the results show that the application of some of the techniques introduced actually improves the classifier's accuracy. In the case of the classification on the Twitter dataset, it was evaluated before implementing the extraction of relevant features (just keeping the n most frequent words). The inclusion of some relevant features based on the text characteristic themselves, combined with common natural language processing techniques, provides a significant improvement on the results attained.

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 (a false statement which is intentionally written pretending to seem true).

The extraction of features related to the nature of the text itself has proven useful for increasing the model's accuracy and might indeed be decisive on the results attained. Particularly, when analyzing Twitter texts, characteristic traits from this media were extracted, such as *hashtags*, user mentions, and *emojis*. This led to the construction of a different classifier for each media, trained with their corresponding vectors of features.

In a world of social media and fake news, this effort is oriented to provide a solution to the problem of determining the truthfulness of a statement just from the way it is written and expressed. Further efforts aim to take into account linguistic twists which may imply false positives, such as irony or humor.

References

1. Rangel, F., Charfi, A., Rosso, P., Zaghouani, W.: Detecting deceptive tweets in arabic for cyber-security
2. 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)