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, 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 [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.

[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 or this process in the current task must be applied to the problems of author profiling and the more definite problem of deception detection. These 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 [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 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 Subsection 3.1), the F-Score obtained improved to 0.70 on the news dataset and 0.77 on the Twitter dataset.

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)