Deception Detection in Arabic Texts Using N-grams Text Mining

Jorge Cabrejas, Jose Vicente Martí, Antonio Pajares, and Víctor Sanchis
{jorcabpe, jvmao82, apajares71, vicsanig}@gmail.com

Universitat Politècnica de València,
Camino de Vera s/n, 46022 Valencia, Spain

Abstract In this paper, we present our team participation at Author Profiling and Deception Detection in Arabic (APDA) - Task 2: Deception Detection in Arabic Texts. Our analysis shows that the accuracy of a unigram method outperforms both bigrams and hybrid modeling based on unigram and bigrams. We show that the accuracy of the unigram modeling can achieve performance above 76%.

Keywords: Arabic · deception · n-grams

1 Introduction

Social networks such as Facebook, Twitter or Instagram are excellent communication channels between customers and different brands. According to statistics, approximately 92% of business-to-business marketers in North America use social networks as part of their marketing tactics [1]. For companies, this is the paramount importance as a means of customer acquisition. On the other hand, customers generally send thousands of messages to show their personal opinions in social networks either to criticize or recommend, for instance, a hotel or restaurant. Many times these views are unbiased and help companies to improve their mistakes. However, many others are full of hatred and resentment and sometimes they cause damage to the reputation of the company.

Being aware of the importance of social networks in the commercialization of the companies, it is key to detect when a comment in social networks is real or when it is suspected of deception and tries to damage the reputation of the brand. This problem is constant throughout the world, regardless of the country. However, for instance, the Arabic language suffers from a lack of Natural

Language Processing (NLP) techniques that does not happen in the English language. In particular, one of the first contributions can be found in [2] where authors try to detect spam opinions in the Yahoo!-Maktoob social network through dictionaries of linguistic polarity and machine learning classification techniques. However, there have only been some articles in the literature that aim to detect the irony [3], analyze the sentiment [4] or profile authors in Arabic texts [5].

In this paper, we present our participation in Author Profiling and Deception Detection in Arabic (APDA) - Task 2 carried out together with the FIRE 2019 Forum for Information Retrieval Evaluation 12-15 December 2019, Kolkata, India [6]. In the article, we show that word n-grams (hereinafter referred to n-grams) are appropriate methods to detect when a tweet or a new is false or true.

2 Task Description

Given a set of Arabic messages on Twitter or new headlines, the goal of the task is to predict when a given message is deceptive when it was written trying to sound authentic to mislead the reader. Therefore, the target labels are *Truth* or *Lie*. The organizers provided [1] 1443 new headlines (APDA@FIRE.Qatar-News-corpus.training.csv) and 532 tweets (APDA@FIRE.Qatar-Twitter-corpus.training.csv) to train our model. Both files have identically the same fields, that is, an ID (identifier of each text), a label (to describe if a text is true or false), and a text (words to analyze). It is important to say that some tweets or news were repeated throughout the file and we decided to delete them so as not to overtrain the model with those messages. Similarly, organizers also provided 370 tweets (APDA@FIRE.Qatar-Twitter-corpus.test.csv) and 241 new headlines (APDA@FIRE.Qatar-News-corpus.test.csv). In this case, the files contain only the ID and the text but not the label, which has to be predicted with a statistical model.

3 Deception detection algorithm

In this section, we analyze the n-grams methods that can be found in the literature in their standalone version (unigrams or bigrams) or hybrid (unigrams and bigrams) for text processing.

3.1 n-Grams

An n-gram model consists of a probabilistic method that attempts to predict the x_i item based on the items $x_{i-(n-1)}$, ..., x_{i-1}, that is:

$$P_{x_i} = P(x_i|x_{i-(n-1)}, ..., x_{i-1}). \quad (1)$$

[1] https://www.autoritas.net/APDA/corpus/

In our case, an item is a word but it could be a syllable, a set of letters, phonemes, etc. For instance, for $n = 1$, the probability of the word *example* is written along with the word *for* in English is very high. However, the probability of the word *can* is written together with the word *would* is zero, $P_{can} = P(can|would) = 0$.

The probabilistic method proposed in this paper is based on the next steps:

1. We divide L sentences of Twitter or headlines with the following delimiters: *spaces* ( ), *dots* (.), *question marks* (?), *exclamation marks* (!), *parentheses* (()), *semicolons* (;) or *commas* (,).
2. We eliminate a set of words (stop words) that are common in the Arabic language.
3. We remove punctuation marks, numbers, hashtags, and extra whitespaces.
4. We get all combinations of n words from all sentences. For example, in English, the sentence *In summer the temperatures are high in middle east* is divided in the following combinations: *in summer*, *summer the*, *the temperatures*, *temperatures are*, *are high*, *high in*, *in middle*, *middle east*.
5. We calculate the P_{x_i} probability of all combinations.
6. Given the large number of combinations, we select the T combinations with the highest probability.
7. We build the final dataset with L rows and T columns.
8. Finally, with machine learning, we classify tweets or headlines.

Note that this approach has been carried out for the training and test data.

4 Experiments and Results

To carry out the Task 2, we use a unigram modeling to predict whether a tweet or headline is misleading or not, and we compare its performance with the following methods:

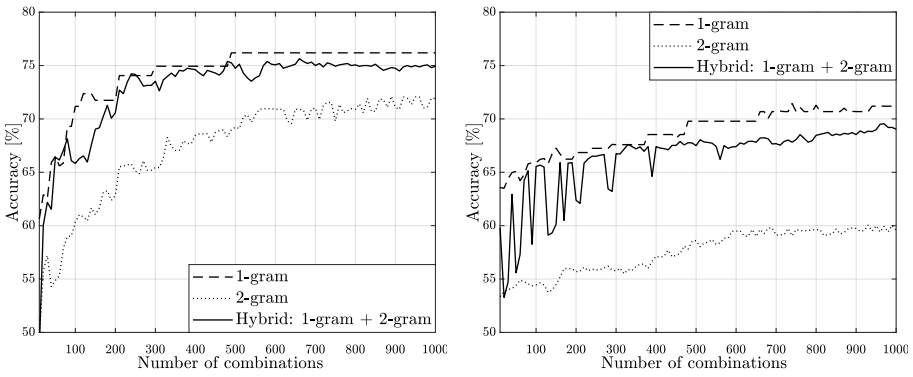
1. Bigrams.
2. A hybrid method that uses unigram and bigrams.

We use 751 common words to remove them from the texts to analyze [2]. Besides, instead of using the probability as a metric of deception, we use the number of times each combination appears. Finally, we have tested several machine learning algorithms with the qdap, tm, XML, splitstackshape, caret, and RWeka libraries of the R programming language. In particular, we have tested k-Nearest Neighbour, Neural Networks, Random Forest, and Super Vector Machine (SVM). We verified that SVM is the learning algorithm that offers the best performance with a 10-fold cross-validation experiment and three repetitions.

Figure 1 shows the performance evolution of the proposed methods based on n-grams to detect the deception in Arabic. This performance evolution is calculated for Twitter (right figure) and new headlines (left figure). We show the accuracy with the number of combinations T explained in Section 3.1. From

[2] https://github.com/mohataher/arabic-stop-words/blob/master/list.txt

the figures, we can conclude that there is a significant performance gap between Twitter and the headline texts. In general, as the number of combinations increases, the accuracy increases for both texts. However, 500 combinations could be good enough to get a good performance at a reasonable cost. We also highlight an erratic performance behavior in the figures with few combinations (see the hybrid method unigram and bigrams). Therefore, it is key to select correctly which combinations can be removed. From both figures, we can conclude that the unigram method is the best algorithm to detect deception in Arabic texts. Table 1 shows the best accuracy values that can be achieved with the proposed methods. Note that the Arabic language is a relatively complex language from the NLP point of view. For instance, some prepositions are joined with some words: كأس العالم (the world cup) and لكأس العالم (for the world cup). This fact causes a performance loss in the n-grams that should be considered. This problem can be solved by using handmade rules that separate prefixes and suffixes from words [5].



(a) Twitter **(b)** New headlines

Figure 1: Accuracy of the proposed methods for deception detection.

Table 1: Maximum accuracy [%] of the proposed methods for deception detection.

	Unigram	**Bigram**	**Hybrid method**
Twitter	76.19	72.12	75.64
New headlines	71.45	60.05	69.55

5 Conclusion and Future Work

In this paper, we presented our participation in Author Profiling and Deception Detection in Arabic (APDA). We worked on a unigram approach and compared its performance with bigrams and a hybrid method that used unigrams and bigrams. Our proposed approach achieved good performance compared to the other two methods. In particular, we obtained an accuracy of 76.19% for Twitter texts and 71.45% for new headlines. As future work, we could include letters as items instead of words to deal with the complexity of the Arabic morphology where some prepositions are joined with words.

References

[1] Statista. Social media and user-generated content. https://www.statista.com/topics/2057/brands-on-social-media/, 2019. [Online; accessed 20-aug-2019].

[2] Heider Wahsheh, Mohammed Al-Kabi, and Izzat Alsmadi. Spar: A system to detect spam in arabic opinions. 12 2013.

[3] Jihen Karoui, Farah Banamara Zitoune, and Veronique Moriceau. Soukhria: Towards an irony detection system for arabic in social media. *Procedia Computer Science*, 117:161–168, 2017.

[4] Hossam S. Ibrahim, Sherif Mahdy Abdou, and Mervar H. Gheith. Sentiment analysis for modern standard arabic and colloquial. *International Journal on Natural Language Computing (IJNLC)*, 4, 2015.

[5] Paolo Rosso, Francisco Rangel, Irazu Hernandez Farias, Leticia Cagnina, Wajdi Zaghouani, and Anis Charfi. A survey on author profiling, deception, and irony detection for the arabic language. *Language and Linguistics Compass*, 12:1–20, 2018.

[6] Francisco Rangel, Paolo Rosso, Anis Charfi, Wajdi Zaghouani, Bilal Ghanem, and Javier Sánchez-Junquera. Overview of the track on author profiling and deception detection in arabic. *In: Mehta P., Rosso P., Majumder P., Mitra M. (Eds.) Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings. CEUR-WS.org*, Kolkata, India, December 12-15.