IRIT at CheckThat! 2018

Romain Agez[1], Clement Bosc[1], Cedric Lespagnol[1],
Josiane Mothe[2][0000−0001−9273−2193], and Noemie Petitcol[1]

[1] Université P. Sabatier de Toulouse, UPS, France
FirstName.LastName@univ-tlse3.fr
[2] IRIT, UMR5505, CNRS & Université de Toulouse, France
Josiane.Mothe@irit.fr

Abstract. The 2018 CLEF CheckThat! is composed of two tasks: (1) Check-Worthiness and (2) Factuality. We participated to task (1) only which purpose is to evaluate the check-worthiness of claims in political debates. Our method to achieve this goal is to represent each claim by a vector of five computed values that correspond to scores on five criteria. These vectors are then used with machine learning algorithms to classify claims as check-worthy or not. We submitted three runs using different machine learning algorithms. The best result we achieved using the official measure MAP ranks our primary run the 12th over the 16 submitted runs. Our first contrastive run is ranked 2nd with the Mean Precision@1 measure.

Keywords: Information retrieval · fact checking · information nutritional label.

1 Introduction

The CLEF CheckThat! pilot task aims at predicting which claims in political debates should be prioritized for fact-checking [6].

To achieve this goal, the task organizers released several textual transcripts of political debates with each sentence being annotated according to whether it is check-worthy or not.

This paper describes the participation of the Université de Toulouse team (official name RNCC) at CLEF 2018 CheckThat! pilot task for check-worthiness.

We preprocessed the data by representing each sentence corresponding to a transcription of what a speaker said in the debate by a vector containing the score of this sentence for five different criteria. We then trained three classifiers using these vectors to submit three different runs.

The remaining of this paper is organized as follows: Section 2 gives a description of the pilot task. Section 3 details the model we developed and the submitted runs. Then Section 4 details the results we obtained. Finally, Section 5 concludes this paper.

2 Task Description

2.1 Objectives

The Check-Worthiness task aims to predict which statements in a political debate should be fact-checked. Indeed, nowadays, information objects are spreading faster and faster on the Internet and especially on social networks. This spreading is named the *virality* of the information [1].

During a political debate, any of the statements made by the participants can be reused without checking its factuality and it even can become viral. CheckThat! aims at providing journalists with a list of statements members of the debate made, that should be checked before they are reused by others.

2.2 Dataset

There are two datasets : one to train the model and one to test it. Both sets consists of political debates transcribed into texts.

They are annotated so that each line indicates the sentence number, the speaker, the transcription of the sentence that the speaker said. The training dataset has in addition a label that indicates whether this sentence is to be fact-checked or not The training set contains three political debates while the test set contains seven debates [6].

2.3 Evaluation metric

The task has been evaluated according to different measures. The official measure is MAP which calculates the usual mean of the average precision. Then, other measures were used as Mean Reciprocal Rank which allows to obtain reciprocals of rank of the first relevant document as well as Mean Precision at x which performs the average of x best candidates. Details on the measures used can be found in the task overview [6].

Evaluations are carried out on primary and contrastive runs. Primary run corresponds to the results file of the participant's main model ; the decision of the main run was the participant's decision. Contrastive runs match the secondary models the participant used.

3 Method and runs

We computed five of the criteria from the Information Nutritional Label for Online Documents proposed by [3]. These criteria and the methods used to calculate their score are as follows:

- **Factuality and Opinion** : Determines whether a sentence represents a fact or a personal opinion. We use a Multi-layer Perceptron classifier, using LBFGS gradient descent [8]. The datasets to train the neural network come

from various Wikipedia[3] articles for factual sentences and from Opinosis[4] for opinion sentences. The features used to classify a sentence are fine-grained part-of-speech tags extracted with spaCy[5].
- **Controversy** : Determines the degree of controversy in a text. We count the number of controversial issues in the text based on the Wikipedia Article List_of_controversial_issues[6]. For each issue referenced in the wiki article, we also take in account the anchor text labels[7] to find the synonyms and other appellations of the issues in all of the Wikipedia database. For example : Donald Trump is in the list of controversial issues. Other names can link to his Wikipedia page such as "45th President of America". These names are called anchor text labels and will be recognized as a controversial issue.
- **Emotion** : Determines the intensity of emotion in a sentence. We use the list of $2,477$ emotional words and valuation from AFINN[8] [7] (ex : abusive = -3, proud = 2). We sum the absolute value of the positive and negative valuations of the emotional words found in the sentence and we divide it by the total number of words in the sentence :

$$(\sum posWordValue + \sum |negWordValue|)/totalNumberWords$$

- **Technicality** : Determines the degree of technicality in a text. We count the number of domain-specific terms in the text. For that, we use NLTK[9] [2] to tag all the words of the text (adjective = JJ, name = NN, etc.). Then, we use the RE library[10] to match, from tags, with a regular expression defined in [5] which identifies the *terminological noun phrases* (NPs). They represent domain-specific terms in the text. We extract all the NPs of the text and keep those which appear more than once. We then calculate the ratio of the number of these NPs on the number of words in the text.

$$(\sum NPs)/totalNumberWords$$

3.1 Models

Each of our three runs uses its own model to compute a check-worthiness score. For each of our models, we preprocessed the data using the criteria previously described. We computed the score of each of these criteria for each sentence that

[3] https://www.wikipedia.org/

[4] http://kavita-ganesan.com/opinosis/

[5] spaCy is a library for Natural Language Processing in Python. It provides NER, POS tagging, dependency parsing, word vectors and more.
https://spacy.io/

[6] https://en.wikipedia.org/wiki/Wikipedia:List_of_controversial_issues

[7] https://en.wikipedia.org/wiki/Anchor_text

[8] http://www2.imm.dtu.dk/pubdb/p.php?6010

[9] Natural Language ToolKit, https://www.nltk.org/

[10] Regular Expression, https://docs.python.org/3/library/re.html

has to be evaluated for check-worthiness. These sentences are then represented by a vector containing five features, one for each criterion score.

For our primary and first contrastive runs, we decided to use the Support Vector Machine in sklearn [11]. We used an RBF kernel for the primary run and a linear kernel for the first contrastive run. For our second contrastive run we used the random forest classifier in sklearn.

To train our models, we used the three annotated debates provided by the clef2018-factchecking github repository[12]. We decided to use a 3-cross validation with the 3 datasets as our validation method. Following the guideline of the task, we trained our models on two of them and tested them on the third.

To obtain a score of check-worthiness, we computed the probability for each sentence to be check-worthy using the classifiers. The score of a sentence was then set to the probability obtained for this sentence divided by the highest probability computed, so that the scores are between 0 and 1.

4 Results

Seven teams submitted runs to this task for a total of 16 runs for the task 1.

Table 1 presents the results of our three runs.

Table 1. Results for each of our runs. Values in parenthesis correspond to the ranks of our runs over the 16 that were submitted.

Name	Model used	MAP	MRR	Mean Precision@1
primary	SVM with RBF kernel	.0632 (16)	.3775 (9)	.2857 (6)
cont. 1	SVM with linear kernel	.0886 (12)	.4844 (5)	**.4286 (2)**
cont. 2	Random Forest Classifier	.0747 (15)	.2198 (15)	.0000 (14)

Overall, our first contrastive run obtained better results than our primary run; that was unexpected since non linear kernel have been shown to work better in information retrieval applications. Our first contrastive run has been ranked twelfth according to the main measurement (Man Average Precision) , but obtained better rank when considering the other measure : it is ranked fifth according to the Mean Reciprocal Rank and second according to the Mean Precision@1.

5 Conclusion and perspectives for future works

To improve our models, we could add weights for each of the criteria. Indeed, for our models we decided that the value returned by each criterion would represent 20% of the final result. Thereafter, we could consider that the *factuality* and *opinion* criteria would have a higher weighting than the other criteria. The latter

[11] http://scikit-learn.org/stable/modules/svm.html

[12] https://github.com/clef2018-factchecking/clef2018-factchecking/tree/master/data/task1/English

allows to determine whether a sentence represents a fact or a personal opinion, so sentences that represent a fact would have a higher score and vice versa if it is an opinion.

We could also complete the representations of the texts by using content-based components like it is done in [4]. While the objective is different (virality prediction), some of the features may also be useful for the task tackled by CheckThat!. Finally, we could test these models on other datasets such as social networks. For example, we could consider a Twitter-based dataset where each tweet would have a score indicating its worthiness for fact checking taking into account hashtags and tweet sources.

References

1. Berger, J., Milkman, K.L.: What makes online content viral? Journal of marketing research **49**(2), 192–205 (2012)
2. Bird, Steven, E.L., Klein, E.: Natural language processing with python
3. Fuhr, N., Giachanou, A., Grefenstette, G., Gurevych, I., Hanselowski, A., Jarvelin, K., Jones, R., Liu, Y., Mothe, J., Nejdl, W., et al.: An information nutritional label for online documents. In: ACM SIGIR Forum. vol. 51, pp. 46–66. ACM (2018)
4. Hoang, T.B.N., Mothe, J.: Predicting information diffusion on twitter–analysis of predictive features. Journal of Computational Science (2017)
5. Justeson, J.S., Katz, S.M.: Technical terminology: some linguistic properties and an algorithm for identification in text. Natural language engineering **1**(1), 9–27 (1995)
6. Nakov, P., Barrón-Cedeño, A., Elsayed, T., Suwaileh, R., Màrquez, L., Zaghouani, W., Gencheva, P., Kyuchukov, S., Da San Martino, G.: Overview of the CLEF-2018 lab on automatic identification and verification of claims in political debates. In: Working Notes of CLEF 2018 – Conference and Labs of the Evaluation Forum. CLEF '18, Avignon, France (September 2018)
7. Nielsen, F.Å.: Afinn (mar 2011), http://www2.imm.dtu.dk/pubdb/p.php?6010
8. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)