EvolutionTeam at CLEF2020 – CheckThat! lab : Integration of linguistic and sentimental features in a fake news detection approach

Ibtissam Touahri and Azzeddine Mazroui

Department of Computer Science, Faculty of Sciences, University Mohamed First,
Oujda, Morocco
{ibtissamtouahri555 and azze.mazroui}@gmail.com

Abstract. Misinformation is a growing problem around the web. The spread of such a phenomenon may impact public opinions. Hence fake news detection is indispensable. The first step for fact-checking is the selection of check worthy tweets for a certain topic, then ranking sentences from related web pages according to the carried evidence. Afterward, the claim will be verified according to evident sentences. At CLEF2020 – CheckThat! lab, three tasks run in Arabic, namely check-worthiness on tweets, evidence retrieval, and claim verification that corresponds respectively to task1, task3, and task4. We participated in the three tasks. We integrated manual sentiment features as well as named entities to detect fake news. The integration of sentiment information in the first task caused result degradation since there may be an overlap between check worthy and not check worthy tweets. For the second task, we explored the effect of sentiment presence and we used cosine similarity as a similarity measure between the claim and a specific snippet. The third task is a classification task based on sentiment and linguistic features to compute the overlap and the contradiction between the claim and the detected check worthy sentences. The results of task1 and task3 leave large room for improvement, whereas the results of task 4 are promising since our system reached 0.55 of F1-measure.

Keywords: Fact-checking · sentiment features · unsupervised approach.

1 Introduction

Interaction with others online through social media has become indispensable. Social media are not only a way to communicate but also a warehouse of news used by people who seek news and information from social media rather than news organizations. People publish their opinions as if they were facts which can mislead the orientation of the public opinion and have negative effects on the psychology of the people. A high proportion of people is exposed to misleading or false claims. For example during coronavirus pandemic, claims about COVID-19 appeared without trustful reference. Many stories have been found such as the theory that the spread of COVID-19 is caused by 5G technology. The spread of such claims affected people understanding of the pandemic.

Fake news has known explosive growth in recent years especially on social media where a large amount of data is uncontrolled. The extensive spread of this phenomenon may impact individuals and society negatively.

In recent years, fake news appear to mislead the orientation of public opinion for commercial and political purposes. Facts are ignored when shaping public opinion, since appealing to emotions works better as it has a potential impact on the person. Social media publish fake news to affect reader psychology and hence increase readership. With offensive and deceptive words, social media users can get affected by these fake news easily, which brings tremendous effects on society.

The identification of fake news is hard and time consuming. To improve information trustworthiness, we should build systems to detect fake news in real time. Thus, many studies addressed the automation of fake news detection process to facilitate their verification among which [7], [1] and [5].

In the following, we describe our participation in CLEF2020 – CheckThat! lab. The paper is organized as follows, we present previous works, afterward we define the tasks in which we participated, then we describe the external resources used by our system as well as the system approach and we give the obtained results for the classification task (task 4).

2 Previous works

This paper investigates the principal approaches used to define news factuality. In the following, we present some previous works that aimed to detect fake news.

Hansen et al. [5] presented an automatic fact-checking system to detect fake news based on a neural ranking model to check sentence worthiness. The model represents each word in a sentence by both its embedding and syntactic dependencies aiming to capture both semantic and the role of syntax to affect the semantic of terms in the same sentence. The check worthiness ranking is based on a neural network model trained on large a amount of unlabelled data through weak supervision.

Shu et al. [9] presented a review of detecting fake news on social media, they addressed their effect on psychology and social theories. They reported the representative datasets, existing algorithms from a data mining perspective, and the evaluation metrics used to detect fake news. They discussed the challenges of this task, related research areas, and future research directions for fake news detection on social media.

Zafarani et al. [11] presented a paper that introduces the characteristics of fake news that differentiate it from similar concepts such as misinformation to present fake news detection strategies systematically.

Atanasova et al. [1] presented an overview of task1 of the CheckThat! Lab 2019. They reported that eleven teams out of 47 participating teams submitted runs. From the evaluation results, the best performing approaches used logistic regression and neural networks. The best system achieved a mean average precision of 0.166 . The obtained results need improvement, and hence the authors released all datasets and scoring scripts to enable further research in check-worthiness estimation.
Hasanain et al. [7] presented an overview of Task 2 at CheckThat! Lab 2019. The authors provided an annotated Arabic dataset to detect fake news. They used normalized discounted cumulative gain (nDCG) for ranking and F1 for classification. They reported that four teams submitted runs. They released all the datasets and the evaluation scripts from the lab to enable further researches.
Haouari et al. [6] presented their participation in Task 2 of CLEF-2019 CheckThat! Lab. Their runs achieved the best performance in subtasks A and B. Whereas the runs of subtasks C and D, achieved the median performance among participating runs. Subtask B is a classification task, hence they proposed a classification model that uses source popularity features as well as named entities. Their model achieved an F1 score of 0.31. For subtask C, they used BOW and named entities to train a model, they achieved an F1 score of 0.4. For subtask D, they proposed a classification model based on sentiment features.
Ghanem et al. [3] presented their participation at CheckThat!- 2019 lab - Task 2 on Arabic claim verification. They proposed a cross-lingual approach to detect claims factuality. Their approach achieved 0.62 as F1 in subtask-D.

3 Tasks description

CLEF2020 – CheckThat! lab proposed many tasks among which three tasks that run in Arabic. We have participated in the three tasks, namely task1, task3 and task4. In the following, we describe each task according to its presentation by the lab organizers.

3.1 Task1 : Tweet Check-Worthiness

The organizers gave a set of topics and their corresponding potentially-related tweets. This task aims to verify whether a tweet is check worthy. A tweet is considered check worthy if it carries harmful content or it is of interest to a large audience. This task is a ranking task that aims to rank the tweets according to their check-worthiness for the topic. The official measure used for evaluation is P@30 for the Arabic dataset.

3.2 Task3 : Evidence Retrieval

For this task, the organizers presented a set of topics and the corresponding claims and a set of text snippets extracted from potentially-relevant webpages. The task aims to return for a given claim a ranked list of evidence snippets that support or refute the claim, namely the ones that are useful in verifying it.

3.3 Task4 : Claim Verification

The task presents a dataset that contains 201 check-worthy claims related to 12 topics. For these topics, a set of potentially related web pages is given. The task aims to use the data to predict claims veracity. The task is a classical binary classification task that uses true or false tags to tag a specific claim according to its veracity. Precision, recall, and F1-measure are used as evaluation measures and the macro-averaged F1 is used as the official measure.

4 External resources

This section aims to describe the resources used by our system besides the ones presented by the task organizers. We constructed four lexicons, namely sentiment, offense, sarcasm, and named entities lexicons. The lexicons are described in the following:
Sentiment lexicon: the lexicon contains 9858 sentimental terms. This lexicon is a combination of many resources which are:
Lexicon1(SemEval [1]): a lexicon that contains sentimental terms and their corresponding sentiment intensity.
Lexicon2 (MPQA [2]: the Arabic version of the original MPQA lexicon that contains sentimental terms.
Lexicon3 (ENGAR): is an English sentiment lexicon created by [8] and then translated into Arabic by the authors of this paper.
Lexicon4: a sentimental lexicon extracted from a corpus collected from Hespress [3] Facebook page.
The lexicons have been verified semantically by the authors of this paper. We give in Table 1 the statistics of sentiment lexicons.

Table 1. Statistics of sentiment lexicons

Lexicon	Lexicon1	Lexicon2	Lexicon3	Lexicon4	Total	Total unique
Statistics	980	4166	3504	1778	10428	9858

Offense lexicon: the offensive lexicon is sharper than the negative sentiment lexicon. We constructed a lexicon by extracting offensive terms from the offensive corpus collected by the organizers of the offensive language detection shared task [4]. The lexicon contains 1120 offensive terms.
Sarcasm lexicon: the lexicon contains 148 sarcasm indicators extracted manually from the ironic corpus that was created by [4].
Named entities lexicon : the lexicon contains the names of religions, countries and

[1] http://www.saifmohammad.com/WebPages/SCL.html
[2] http://www.purl.org/net/ArabicSA
[3] https://fr-fr.facebook.com/Hespress
[4] https://sites.google.com/site/offensevalsharedtask/

known personalities. The terms of this lexicon were collected by google queries and then were expanded by the authors of this paper.
a) Religion lexicon: contains 9 religions which give 104 terms of the corresponding adjectives and nouns.
b) Nationality lexicon: contains 194 countries. We enhanced the names of the countries by the corresponding nationalities.
c) Named entities: contains named entities extracted from the offensive corpus. The terms target religions (مسلم), countries (قطر), backgrounds (فرس), sports teams (زمالك), political parts (حوثي), genders (نسوان), famous personalities (سيسي). The lexicon contains 216 terms.
We give in Table 2 examples of the mentioned lexicons.

Table 2. Statistics and examples of lexicon terms

Lexicon	Sentiment	Offense	Sarcasm	Named entities
Size	9858	1120	148	514
Example	اجمل	الخونه	ههه	إيران

The presence of sentiment and sarcasm terms within an expression may indicate that the expression is an opinion not a fact. The cause behind using offense lexicon is that it may define harmful expressions. We use named entities to define trustful sources since a text that contains named entities tends to be more factual. The lexicons have been created for sentiment analysis purposes, they have not been made publically available yet. We use them as external resources that cover a large set of sentiment terms.

5 System approach

5.1 Text extraction and preprocessing

We extract texts of tweets, claims, and snippets from the given JSON object using regular expressions. For task 4, instead of using Jsoup that is a java library that parses HTML documents as in [10] to extract the content of relevant web pages, we use the text snippets that were extracted from these pages. We give a standard representation to the extracted text, we preprocessed the claims and the text snippets extracted from potentially relevant webpages by removing all characters other than the Arabic letters. We tokenize each text into terms using space delimiter. Hence, each claim and text snippet will be represented by a set of terms.

5.2 Task1

In this task, we aim to rank the top 500 tweets related to each topic according to their check-worthiness. In the following, we use the resources created by our

system to rank tweets besides other features. We use different features namely the title and the description of the topic, sentiment, offense and named entities features. We weigh each feature using a weight that represents its importance.
F_1: represents the weighted intersection between the topic title and tweet text. We give the title the weight 3 as it is the most important part of the content.
F_2: is the weighted intersection between the topic description and the tweet text. We give the description the weight 2 since it contains important information.
F_3: represents the occurrence of named entities in the tweet text. Texts that contain named entities are check worthy as they represent a trustful source of information. We give this feature the weight 1.
F_4: represents the occurrence of offense lexicon terms in the tweet text. The offense lexicon is an indicator of the presence of harmful content. We give this feature the same weight as named entities.
F_5: represents the weighted occurrence of sentiment terms in the tweet text. The text that contains sentiment lexicon tends to be an opinion not a fact. This feature is given a negative weight -1.
All the features are given a positive weight except sentiment features since check-worthy tweets tend to be facts rather than opinions, hence we give a negative weight to the present sentiment terms. We give positive weight to offense feature since from the definition, check-worthy tweets are the ones that carry harmful content. Since the title and description are related to the topic, then each tweet is given a score based on the product of the mentioned features weights and their intersection with the tweet text. In other words, whenever a topic title or description term matches tweet text term, we increment the value by 3, 2 or 1 according to the corresponding weight for each feature. The same for other features. The score is then divided by the sum of feature sizes which gives a normalized score.

Table 3 gives an example of initial values. Using Formula (1) we compute the normalized score for each tweet. L_i is the length of each lexicon. The statistics of each lexicon are given in Table 2.

Table 3. Example of initial values

Feature	Topic title	Topic description	Sentiment	Offense	Named entities
Size	12	36	100	50	10
Weight	3	2	-1	1	1
Intersection with the tweet	2	5	3	1	2

$$\frac{\sum_{n=1}^{5} F_i}{\sum_{n=1}^{5} L_i} = \frac{6+10-3+1+2}{12+36+100+50+10} = \frac{16}{208}; F_i = Intersection_i \times Weight_i \quad (1)$$

This approach showed degraded results in comparison to the approach [10] that uses only document parts as features to rank web pages. In the official results it reached only 0.28 using P@30 which was under the baseline. The reason for this may be the intersection between check worthy and less check worthy terms that match the extracted features which means that they may characterize both of which or rather the selected features match more less check worthy tweets which made the ranking difficult.

5.3 Task3

We aim to rank a list of text snippets based on their evidence namely their usefulness for fact checking. We extract a text snippet and compare it with a tweet text. Then we affect a score to each snippet based on the results of (2) that gathers cosine similarity between the tweet and the text snippet and weighs using a negative weight the intersection between a specific snippet and the sentiment lexicon. In other words if five sentiment terms are present in the text snippet, then the intersection is five. The negative weight is given to differentiate between facts and opinions. We rank the top 100 evidence text snippets corresponding to each tweet based on the relation (2) and also using only on the cosine similarity score. When multiple snippets have equal cosine similarity score we break the tie by considering both of which if they are ranked with the top 100. Using cosine similarity only shows better results than adding sentiment information. This may be explained by the fact that sentiment terms may appear in a factual text without the intention of the holder to express an opinion. However the obtained results for this task were degraded by reaching 0.05 only using P@10 metric.

$$Score = cosineSimilarity - 0.5 \times intersection \tag{2}$$

5.4 Task4

In this task, we aim to classify claims as true or false. We compare each claim with the text snippets extracted from the relevant web pages. We calculate factuality based on a snippet information using two values identical and opposite, we define each of which by:
Identical: represents the concordance between the claim terms and the text terms.
Opposite: is a negative value that represents the number of claim terms which opposite appear in the text.
In order to define the negated terms we use a list of negation words. If a claim term matches a text term then based on table 4, we can define whether they are identical.

Table 4. Concordance between claim and text terms

Claim term Negated		Text term Negated		Concordance
Yes	No	Yes	No	
*		*		Identical
*			*	Opposite
	*	*		Opposite
	*		*	Identical

We use snippets extracted from potentially relevant Web pages. For a given Web page, if a snippet supports the claim as none of its terms are negated, whereas, a second snippet contradicts the claim as one or many of its terms among the ones that match claim terms are negated. Then the claim is false at the current Web page level. According to the relation (3), if the factuality is greater than 0, then the claim is true. Unless, the factuality will be negative which means the presence of snippets opposite to the claim. Thus, wherever our system finds contradicting snippets, it tags the claim as false, otherwise, it gives it true tag. This threshold has been chosen since according to Baly et al. in [2] a major part of documents which represent snippets can support true claims, however, a major part can support also false claims this means that even when enlarging the threshold we will encounter the mentioned constraint and hence we chose the presence of opposite snippets as an indicator. Then the factuality of a claim according to the potentially relevant Web pages is the major score of true and false values of the initial factuality $\mathrm{Factuality}_{\mathrm{Initial}}$ calculated using each Web page. In other words if the $\mathrm{Factuality}_{\mathrm{Initial}}$ is true according to two Web pages and false according to three Web pages then the factuality of the claim is false.

$$Factuality_{Initial} = Identical \times \mathit{opposite} \tag{3}$$

In table 5 we give the results of Task4 using the mentioned criteria.

Table 5. Task 4 official results

Class	Accuracy	Precision	Recall	F1
FALSE	0.9273	0.1250	0.1667	0.1429
TRUE		0.9682	0.9560	0.9620
Average		0.5466	0.5613	0.5524

The second test is based on the following criterion, if the factuality is false according to the aforementioned criteria or a sarcasm indicator is present in the text, then the claim is false. The presence of sarcasm features augments the probability of the analyzed sentence to be fake. The second test uses a list of negation terms that contains 189 terms. Adding sarcasm features doesn't generate any improvement, which may be due to the weak intersection between sarcasm lexicon and the analyzed text and hence we don't report the results of the corresponding test.

6 Conclusion

In this paper, we presented our participation in CLEF2020 – CheckThat! lab. We aimed to build a sentiment aware fake news detection system. We participated in three tasks, we based approaches on various mathematical dependencies. We enhanced the used approach by adding sentiment features to define the impact of it on the detection of fake news. The challenge of this paper wasn't the integration of sentiment features only, but also we aimed to base our system on an unsupervised approach to overcome the difficulties of datasets collection and annotation and also to reduce the time of fact-checking taken when building supervised models. The obtained results in the classification task are promising, however, there is a large room for improvement when it comes to ranking tasks.

References

1. Atanasova, P., Nakov, P., Karadzhov, G., Mohtarami, M., Da San Martino, G.: Overview of the clef-2019 checkthat! lab: Automatic identification and verification of claims. task 1: Check-worthiness. In: CLEF (Working Notes) (2019)
2. Baly, R., Mohtarami, M., Glass, J., Màrquez, L., Moschitti, A., Nakov, P.: Integrating stance detection and fact checking in a unified corpus. arXiv preprint arXiv:1804.08012 (2018)
3. Ghanem, B., Glavas, G., Giachanou, A., Ponzetto, S.P., Rosso, P., Pardo, F.M.R.: Upv-uma at checkthat! lab: Verifying arabic claims using a cross lingual approach. In: CLEF (Working Notes) (2019)
4. Ghanem, B., Karoui, J., Benamara, F., Moriceau, V., Rosso, P.: Idat at fire2019: Overview of the track on irony detection in arabic tweets. In: Proceedings of the 11th Forum for Information Retrieval Evaluation. pp. 10–13 (2019)
5. Hansen, C., Hansen, C., Alstrup, S., Grue Simonsen, J., Lioma, C.: Neural check-worthiness ranking with weak supervision: Finding sentences for fact-checking. In: Companion Proceedings of the 2019 World Wide Web Conference. pp. 994–1000 (2019)
6. Haouari, F., Ali, Z.S., Elsayed, T.: bigir at clef 2019: Automatic verification of arabic claims over the web. In: CLEF (Working Notes) (2019)
7. Hasanain, M., Suwaileh, R., Elsayed, T., Barrón-Cedeno, A., Nakov, P.: Overview of the clef-2019 checkthat! lab: Automatic identification and verification of claims. task 2: Evidence and factuality. In: CLEF (Working Notes) (2019)
8. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the 14th international conference on World Wide Web. pp. 342–351 (2005)
9. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter **19**(1), 22–36 (2017)

10. Touahri, I., Mazroui, A.: Automatic verification of political claims based on morphological features. In: CLEF (Working Notes) (2019)
11. Zafarani, R., Zhou, X., Shu, K., Liu, H.: Fake news research: Theories, detection strategies, and open problems. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 3207–3208 (2019)