Automatic Verification of Political Claims based on morphological features

Ibtissam Touahri and Azzeddine Mazroui

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

Abstract. Fact-checking aims to identify the veracity of the enormous amount of data generated on the web. It enhances and automates human effort in detecting harmful misinformation. In this paper, we present a description of the system with which we participated in three subtasks A, B, and D of Task2 at CLEF 2019 Check That lab. For the subtask A, we propose an unsupervised approach that extracts information from the important parts of a document to rank web pages based on their usefulness, and subsequently verify the target claim. Hence, we assign weights for important document fields in order to compute field scores that are employed besides field lengths to determine the overall score of a given web page. For the subtask B, we adopt a supervised approach that generates a model using an SVM classifier from training data, then we apply it on test data to determine the usefulness degree of a web page. Subtask D aims to tag claim factuality as true or false. Thus, we use lemma, the morphology form that represents many inflected terms, in order to find the claim terms in target web pages. We proceed to detect stance, then we determine whether analyzed expressions confirm or contradict a given claim.

Keywords: Fact-checking · relevance · stance.

1 Introduction

The old fashioned rhythm to check claims veracity can't be kept by Fact checkers as it is less effective in term of detection time due to its slowness in comparison to the need of momentary detection of fake news in a large amount of information on the web. Thus the need for an automatic system for Fact-checking is indispensable.
CLEF-2019 CheckThat! Lab [7] aims at determining evidence and factuality for given claims by proposing two tasks. The first task [2] runs in English and it determines the prioritized claims for Fact-checking within a piece of text. Whereas, the second task is in Arabic and it identifies evidence and factuality. The second task [8] is divided into four subtasks where subtask A aims to rank web pages based on their usefulness for verifying a given claim. Subtask B classifies web pages based on the degree of their usefulness to verify a given claim, thus each page is tagged as either very useful, useful,

not useful or not relevant. Subtask C extracts passages from useful pages in order to find the ones that are useful for claim verification. Whereas, subtask D targets identifying claim factuality as true or false. CheckThat! Lab released both training and test data for the aforementioned subtasks. In this paper, we present our approaches to deal with Fact-checking problem related to the second task.

For subtask A, we rank web pages based on the overall score computed by addressing weights of document fields namely, meta-data, title, and body text. Subtask B is a classification task, thus we extract features from web pages using documents related information which is URL data and page content as well as the extraction of named entities. In order to meet the need of subtask D, which is identifying whether a claim is true or false, we proceed to detect stance by looking for terms from web pages that match claim terms, then we determine whether an expression confirms or contradicts the claim.

The remainder of this paper is organized as follows. We start by situating our approaches in comparison to related work. Then, we describe our system by giving details related to each subtask, and after we give the obtained official evaluation results. We conclude by giving some perspectives.

2 Related work

In the literature, many works pointed at generating methods and approaches that help for Fact-checking. Jaradat et al. in [9] presented an online claim rank system that supports Arabic besides English language. The system was trained on political debates. Moreover, it works with other kinds of data. Baly et al. in [3] described a unified corpus for stance detection and Fact-checking. Their system noticed the main challenge when using only stance detection to predict facts as their system illustrates that for true claims the major part of documents supports them, however, a major part supports also false claims. Thus, the insufficiency of stance detection to predict claim factuality. At CLEF-2018 CheckThat! Lab, Authors of [10] proposed a learning to rank approach that gathers NLP features, namely sentiment analysis and named entities. Afterward, they extracted relevant segments from a set of web pages in order to predict claim veracity.

3 Task 2: Evidence and Factuality

This task aims to identify useful web pages and passages which assist in the human Fact-checking task by following the three first subtasks, the last subtask aims to find supporting and contradicting information within given web pages, thus, identifying claim factuality either true or false.

3.1 System approach

Our system approach addresses both web pages and Arabic language specificities, thus it proceeds to unify input data in term of format and content.

Information extraction In order to rank web pages based on their usefulness, we proceed as follows. We assume that the most important parts in a document are meta-data, title, and body text, with an advantage to the first two fields over the third, thus using Jsoup that is a java library that parses HTML documents, we extract the three contents from each web page. It is worth noting that even word and text documents were converted into HTML format in order to unify the system input.

Preprocessing The aim of preprocessing is giving a standard representation that addresses Arabic language specificities to both claim and web page content, thus claim, meta-data, title and body text are preprocessed by removing Latin characters, punctuation marks and digits in order to keep Arabic characters only, Moreover, we remove stop words that don't carry any important information as they have the same function wherever they are used. The used stop words list contains 157 terms.

SubstaskA We assign weight for document fields, thus we give three points to both of meta-data and title content and one point to body content as we consider that meta-data have an advantage over the body as they are considered a communication way between SEO and search engines, the title too is distinguished as it appears to the user before the body.

$$Sp = F_W \times T_{Occ} \tag{1}$$

We compute the score points (Sp) for each field using formula (1), where, the score is the product of the given field weight ($\mathrm{F_W}$) multiplied by the occurrence of the field terms that belong to the claim.

$$Os = \frac{M_{Sp} + T_{Sp} + BT_{Sp}}{M_L + T_L + BT_L} \tag{2}$$

We compute the overall score (Os) of a given web page by using formula (2), where, $\mathrm{M_{Sp}}$, $\mathrm{T_{Sp}}$, and $\mathrm{BT_{Sp}}$ are score points of meta-data, title, and body text respectively and $\mathrm{M_L}$, $\mathrm{T_L}$, and $\mathrm{BT_L}$ are their lengths. After computing the overall score for each web page, we rank web pages by sorting obtained values in descending order.

SubstaskB We consider subtask B a classification task that classifies web pages based on their usefulness degree for claim verification. Thus, we proceed to extract two categories of features, where each one contains four features.

Category1 uses document fields as features, thus we have four features which are normalized score points of URL, meta-data, title, and body text. It is worth noting that the URL takes the same weight as meta-data and title.

$$NSp = \frac{F_{Sp}}{F_L} \tag{3}$$

The normalized score points of each field are calculated using formula (3). Where, $\mathrm{F_{Sp}}$ value determines the score points of a specific field either URL, meta-data, title or body text divided by the field length ($\mathrm{F_L}$).

Category2 gives named entities count using Farasa that is an Arabic text processing tool that offers many functionalities, among which named entity tag [6]. For a given term Farasa gives either LOC tag if the entity is a location, PERS if the entity identifies a person or ORG when the entity is an organization unless it gives none, in other words, none tag is given when the entity doesn't present any of the aforementioned tags. Thus the sum of tagged and non-tagged entities count presents document length. The system gathers features of category1 and category2 to give a standard representation to all web pages. Hence, in summary, we have eight features, namely, normalized score points of URL, meta-data, title, and body text besides named entity features, that are LOC, PERS, ORG tags count and the count of terms that were not tagged by Farassa. The system starts by training a model using the extracted features from training web pages and SVM classifier, then it labels test data based on the generated model. SVM classifier has been chosen as it had a considerable improvement over Naïve Bayes (NB) and K-Nearest Neighbors (KNN).

SubstaskC The subtask C points at defining the useful passages within a web page, thus, we proceed to extract passages that constitute the web page, then, we compute cosine similarity between the claim and each passage, afterward, we sort passages in descending order by cosine similarity value after removing their redundancy, then we consider the first five passages in the sorted list as useful.

SubstaskD As subtask D aims to identify either a claim is true or false, we proceed to determine web pages terms that match claim terms, thus we use Alkhalil analyzer [5] that doesn't address context, as a result, it gives all potential lemmas of a given term. We use it over all claim terms and for each analyzed term we obtain a list of corresponding lemmas. However, for web pages content, we use Alkhalil lemmatizer [4] that handles the context, thus it gives one equivalent lemma for each analyzed term within a piece of text. We detect all terms that match between claim and web page content by comparing lemmas. Alkhalil analyzer was used for claim text instead of Alkhalil lemmatizer due to claim short length, thus the use of Alkhalil lemmatizer may not give the correct corresponding lemmas. Supporting and contradicting expressions identification is performed by using a list of negation words. The negation words list contains 36 terms manually constructed by browsing different sources. For instance, we have the claim حماية أجهزة أبل قوية بحيث لا تتعرض للفيروسات، the first sentence in Table 1 matches the claim, however, the second contradicts it as ليست قوية contradicts قوية and تتعرض contradicts لا تتعرض. In other words, sentence one supports the claims as none of its terms are negated, whereas, the second sentence contradicts the claim as two of its terms among the ones that match claim terms are negated. Thus, wherever our system finds contradicting sentences, it tags the claim as false, otherwise, it gives it true tag.

Table 1. Stance and relation between claim and sentences

N	Sentence	W1	W2	W3	W4	W5	W6	W7
1	حماية أجهزة أبل قوية بحيث لا تتعرض للفيروسات	S	S	S	S	S	S	S
2	حماية أجهزة أبل ليست قوية بحيث تتعرض للفيروسات	S	S	S	C	S	C	S

Subtask D runs two times at different evaluation cycles, thus we describe our approach for both of them. We apply the aforedescribed process to predict claim factuality for the two cycles, where:

Cycle1 uses unlabelled web pages to estimate claim factuality. Our approach for this cycle is gathering all useful passages from unlabelled pages related to a specific claim, then generating their corresponding in context lemmas. In other words, we use five passages from each unlabelled web page related to the claim. For this cycle, we sent one run.

Cycle2 uses useful web pages to estimate claim factuality. Our approach is the same as for cycle1 except that we use useful passages from useful pages related to the claim only instead of using all pages. For this cycle, we sent three runs, run1 that uses the given test passages from useful web pages that are related to 32 claims besides useful passages selected by our approach that are related to 27 claims to detect factuality of the given 59 claims, the useful passages identified by our approach were added to cover the 27 remainder claims, run2 that uses the given test passages from useful web pages that are related to 32 claims only, and run3 that gathers the five selected passages from each useful web page related to a given claim and it tests factuality of 57 claims.

3.2 Evaluation results

SubtaskA In order to rank web pages, we follow an unsupervised method that is based on weighted document fields. Table 2 shows the results in the official evaluation, the results reached for this task are between 0.19 and 0.5, where our system achieved an AP value of 0.48. In the official evaluation, our run is ranked 3rd (out of 7 runs from 3 teams). However, all runs were outperformed by the baseline. Hence, our system needs further development in order to rank pages by usefulness.

Table 2. Subtask A official results

Our system	Min	Median	Max
0.48	0.19	0.45	0.5

SubtaskB In this task, we point at classifying test data based on a model generated using training data and SVM classifier. Table 3 shows the obtained results in the official evaluation, where our system achieved an accuracy value of 0.79 when testing two classes per claim, 0.78 when testing two classes over all claims and 0.58 when testing four classes over all claims. For the 4-way classification (very useful vs. useful vs. not useful vs. not relevant) our system outperformed the baseline with a slight improvement. However, it was not the case for the 2-way classification (useful vs. not useful), and this may be due to training our system on a limited set of useful pages.

Table 3. Subtask B official results

Accuracy	Our system	Min	Median	Max
2 classes per claim results	0.79	0.5	0.785	0.79
2 classes over all claims	0.78	0.49	0.78	0.78
4 classes over all claims	0.58	0.24	0.58	0.60

SubtaskD In this task, we aim to detect claim factuality as true or false. This task runs at two different cycles the one that uses all web pages to detect claim factuality and the one that uses only useful web pages. Table 4 shows the official evaluation results for subtask D. Our team participates with one run for cycle1 and three runs for cycle2. For cycle1 our system reached 0.53 of accuracy, for cycle2, it reached 0.46 for both run1 and run2 and 0.6 for run3. However, run2 and run3 can't be compared to the overall results as they predict only claims that have related useful passages and useful web pages. In the official evaluation, our system outperformed the baseline in both cycles. For cycle1 our run was the only evaluated run. However, in cycle2 our run reached the minimum accuracy. Thus, the use of a supervised approach in order to improve our system is essential.

Table 4. Subtask D official results

Accuracy	Our system	Min	Median	Max
Cycle1	0.53	0.53	0.53	0.53
Cycle2	Run1 0.46 Run2 0.46 Run3 0.6	0.46	0.54	0.63

4 Conclusion

In this paper, we present a description of our participation in CLEF-2019 CheckThat! lab. We start by describing the followed approach to rank web pages based on their usefulness, thus we use information related to document content by giving weights to document parts according to their importance. Then, we classify web pages by usefulness level. The classification task is performed by constructing a model using training data and SVM classifier, then applying the model on the test data. Afterward, we extract useful passages from the given pages using cosine similarity, then, we select the most important passages that belong to the first positions in the sorted list in descending order according to cosine similarity value. The extraction of useful passages helps to meet the need of subtask C and subtask D at the same time. However, our participation points only at subtask D. For this task, we detect stance by defining terms that match between claim and page content, then by using a list of negation words, we detect supporting and contradicting parts to define claim factuality. Our system reached promising results, however, further work with various approaches and several experiments is needed. Thus we intend to improve the supervised approach of subtask B by extracting more significant features as well as using supervised approaches for both subtask A and subtask D instead of unsupervised ones, besides using not only lemmas but synonyms and antonyms as well to meet the need of subtask D.

References

1. Working Notes of CLEF 2019 – Conference and Labs of the Evaluation Forum. CEUR Workshop Proceedings, CEUR-WS.org, Lugano, Switzerland (2019)
2. Atanasova, P., Nakov, P., Karadzhov, G., Mohtarami, M., Da San Martino, G.: Overview of the CLEF-2019 CheckThat! Lab on Automatic Identification and Verification of Claims. Task 1: Check-Worthiness. [1]
3. 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)
4. Boudchiche, M., Mazroui, A.: Approche hybride pour le développement d'un lemmatiseur pour la langue arabe. In: 13th African Conference on Research in Computer Science and Applied Mathematics, Hammamet, Tunisia. p. 147 (2016)
5. Boudchiche, M., Mazroui, A., Bebah, M.O.A.O., Lakhouaja, A., Boudlal, A.: Alkhalil morpho sys 2: A robust arabic morpho-syntactic analyzer. Journal of King Saud University-Computer and Information Sciences **29**(2), 141–146 (2017)
6. Darwish, K.: Named entity recognition using cross-lingual resources: Arabic as an example. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 1558–1567 (2013)
7. Elsayed, T., Nakov, P., Barrón-Cedeño, A., Hasanain, M., Suwaileh, R., Da San Martino, G., Atanasova, P.: Overview of the CLEF-2019 CheckThat!: Automatic Identification and Verification of Claims. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. LNCS, Lugano, Switzerland (September 2019)
8. Hasanain, M., Suwaileh, R., Elsayed, T., Barrón-Cedeño, A., Nakov, P.: Overview of the CLEF-2019 CheckThat! Lab on Automatic Identification and Verification of Claims. Task 2: Evidence and Factuality. [1]
9. Jaradat, I., Gencheva, P., Barrón-Cedeño, A., Màrquez, L., Nakov, P.: Claimrank: Detecting check-worthy claims in arabic and english. arXiv preprint arXiv:1804.07587 (2018)
10. Yasser, K., Kutlu, M., Elsayed, T.: bigir at clef 2018: Detection and verification of check-worthy political claims. Cappellato et al.[5]