Checking fake news on web browsers: an approach using collaborative datasets

Anderson C. Charles[1][0000-0002-4886-5949] and Jonice de Oliveira Sampaio[2][0000-0002-2495-1463]

[1,2] Institudo de Informática
Universidade Federal do Rio de Janeiro - UFRJ
andersoncordeironf@gmail.com, jonice@dcc.ufrj.br

Abstract. Rumors are a constant reality related to information sharing on social networks. The increase of interactions, encouraged by social media, facilitates the dissemination of non-validated content. Sometimes, promoting misinformation and causing irreparable damages. The dynamism of on line activities transforms the process of evaluating the accuracy of a message in a lonely task, user-dependent and often tricky. The lack of reliable and centralized data sources that can be used as a reference for content verification, as well as the lack of tools to support this process, makes it harder to verify facts quickly. This article presents the creation of a collaborative dataset of fake Brazilian news, an API to enable the validation of contents and how this environment was used in the development of an extension for the Google Chrome browser, giving rise to a solution that allows checking a text selected by the user.

Keywords: rumor, information sharing, facts, fake news, data, datasets, browser, fake, validation, disinformation.

1 Introduction

In recent years, the growth of social media has intensified human interaction on the internet, allowing fast information dissemination, which reaches a diverse and broader public. Despite the notorious social contribution brought by social technologies and tools, the difficulty in dealing with information overload contributed to the growing emergence of unverified content (rumors).

Contents are transmitted continuously without a veracity or provenance validation. The speed of communication turns ineffective any process of veracity assessment. Also, conflicts resulting from socio-political divergences drive the practice of producing doubtful content envisioning attention focus and the discord promotion. For example, during the presidential election in the United States, fake news in Facebook has surpassed the content of the mainstream media [1], demonstrating its influence on democratic issues and how important it is to fight disinformation. Moreover, producing fake news has become a profitable business [2].

Rumor identification is a critical challenge [3]. A rumor can be defined as "an unverified claim that starts at one source (or more) and begins to spread over time to several nodes in the network" [4]. Rumors can be classified into three types according to their purpose [5]. First, the news that comes from tabloids, which produce news about scandals, famous people or crimes, with the objective audience. In this scenario, many news stories are created, fake or exaggerated [6].

The second type is hoaxes, which are rumors created in social media, using real facts to be improperly validated, generating more confidence about its message. Unlike a simple joke or material to generate profit through entertainment, hoaxes can cause real damages to someone else [7]. The latter type is the humor news, which relies on deadpan humor and irony to imitate genuine news sources, and has wide reach. Regardless of the type of rumor, content verification is a task that confronts the dynamism and speed of social media. Despite the existence of websites and blogs dedicated to checking facts, finding explanations about a rumor is still a task that requires the use of search engines, which do not consider the impartiality and reliability of a data source when presenting its results, which may increase the misinformation.

The purpose of this article is three-fold, describing: a) the creation of a collaborative dataset about fake news, b) an extension (plug-in) for Google Chrome web browser, which allows the fast verification of content selected by the user, c) an API that can be reused by other applications. This approach used as referential the main Brazilian sites, but the process and the code source can be applied in any language and domain.

2 Related Work

Currently, several techniques have emerged to detect rumors on the internet. In [8], authors used supervised machine learning techniques to classify a topic on Twitter as "news" or "personal talk". Those classified as "news", they were reclassified as "credulous" or "non-credulous". The classification is a supervised process, having a phase of manual annotation (done by human beings)and a learning algorithm extracts the patterns. This work concludes that "news" usually has links to the source and a larger spreading tree. "Credulous" news is propagated by authors with a high quantity of posts, which is originated from one or a few users on the network and has large amounts of retweets. On the other hand, the "non-credulous" news is propagated by ones who use a lot of emoticons, have few friends and have little time of life on Twitter.

In [9],the authors developed a platform called *Hoaxy* to collect news - from different social media and news sites - through crawlers and APIs. After the collection, the platform only tracks updates through RSS. As future work, the authors indicate the development of an interactive web interface for news analysis.

The *Truthy* system [10] is a service designed to track political memes and detect false information in this context. Its framework is responsible for collecting, analyzing, and tracking memes. Other systems have been proposed such as *RumorLens* [11] and *Twitter-Trails*[12], both solutions allow users to explore the rumor spread with an interactive interface, where the user inputs a rumor for searching.

In [13], who after analyzing language and machine learning approaches, proposes a hybrid model combining these two approaches, applying this method to network analysis to create a possible false news detection system. In their experiments, the authors demonstrate that linguistic processing must be constructed in multiple layers and that such tools should be designed to aid in human judgment, but never to replace it.

Author of [14] evaluates the growing number of keywords in times of crisis, relating the location of a given event with the emergence of messages about it and, using machine learning, analyzed the proposed technique in social media during sporting events and adapted to situations of such as the occurrence of earthquakes. Their experiences have shown that, given to social crisis scenario or any type of event that is beyond normality, identifying important words and verifying their origins can help in assessing the credibility of information.

[15] used as source data Wikipedia articles to identify false or unconfirmed content. First, researchers looked at the impact of fake articles by measuring how long they survive before they are eliminated and how many referrals they receive from external links. In addition, using classification tasks, they have found characteristics capable of identifying textual structures and differences between terms that help in the identification of false content. Finally, the authors carry out experimentation involving human beings and conclude that the developed classifier surpasses them, by a great margin, in the task of classifying a content as false or not.

The authors of [16] address the issue of the dissemination of rumors in an epidemiological way, in which users can be infected if they are exposed to this type of information. This methodology has strong relevance when studying the rise and spread of rumors in social media. Epidemiological models can serve as a basis for studies on the relationship between credulity and diffusion in the face of unconfirmed news.

Automatic veracity identification remains a difficult task for machines because it requires a high level of abstraction and creativity. We believe that solutions that involve collaborative knowledge creation - such as crowdsourcing strategies and access to open data sources - are useful to this type of challenge. Our proposal uses these collaborative data as a premise.

3 Fakepedia: the proposal

In our approach, we collect and store in a unique dataset, rumors published in various portals specializing in content verification, so that they are centralized and organized. We have also developed an API to make it easier to interact with the dataset and created an extension for the Google Chrome browser that demonstrates the use of the environment.

As can be seen in Figure 1, the integration between these modules consists of the FakePedia architecture and can be summarized as follows: crawlers download news published by specialized sites, create .xml files and execute the script responsible for Vector Model . Vector Model uses existing .xml files to create .csv files that will serve as a query base, which are indexed (index.py) and stored in the dataset. When a user performs a query in their browser, the extension calls the API that executes the search.py script. This script performs word processing operations on the query and searches the terms in the dataset, based on similarity and *tf-idf* calculations.

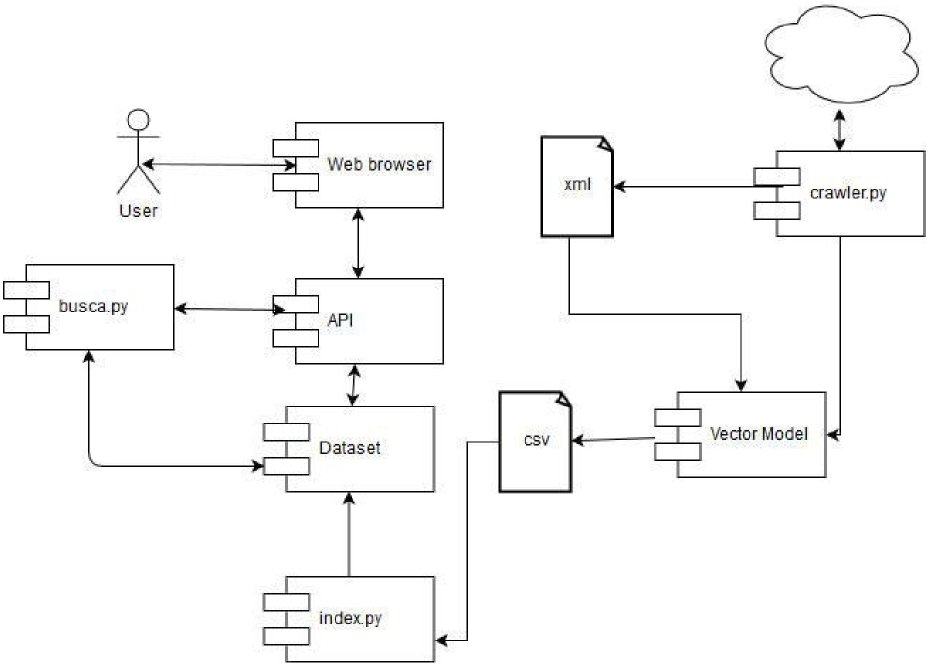
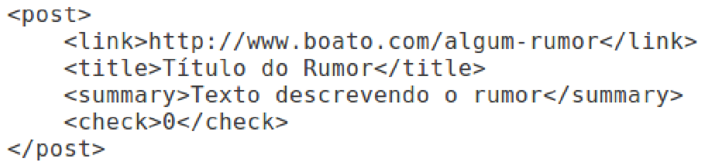


Fig. 1. Architecture of Fakepedia

4 Dataset

We created a dataset with the main fake news. For this, we developed (in Python) crawlers capable of extracting information about rumors already known in different websites and blogs specialized in the subject, in Brazil: *e-farsas*[17], *boatos.org*[18] and "*is or is not*"[19]. The crawlers extract data from sites using the beautiful soup library and write them to an XML file whose structure is shown in Figure 2:



```
<post>
    <link>http://www.boato.com/algum-rumor</link>
    <title>Título do Rumor</title>
    <summary>Texto descrevendo o rumor</summary>
    <check>0</check>
</post>
```

Fig. 2. Structure of the .xml file generated from the collection of rumors

After this step, we run another script for the creation of a structured CSV file with well-defined fields such as "short description of the rumor," and "it is rumor" (*boolean field*). The content saved in these files is then analyzed, so irrelevant terms such as articles and pronouns can be removed. The duplicated news is removed, too.

5 API

To facilitate the use of the data set and to encourage the development of applications that need to identify the rumors, we have developed an API (Application Programming Interface), whose accessibility functions allow interaction with data set records, through routines and patterns that allow the exchange of HTTP request and response messages in JSON format.

The API returns a list of all the records stored by the crawlers. In addition, it is possible to use filters by subject or by the Boolean field to identify whether an item has been classified as rumor or not. The developed API calls are intuitive and indicate what can be returned, as in Table 1.

Table 1. Using the API

URL	Return
/api/news	**List all news**
/apinews/fake	**List all fake news**
/api/news/true	**List all true news**
/api/news/find/[term]	**Search with the term sent**

One of API’s functionality is the search for a term or expression using the Vector Search and Information Retrieval Model. In this process, the Vector Model uses the search expression sent to the API to retrieve relevant documents in the fakenews dataset, resulting in a list of references sorted by the degree of similarity (relevance). The similarity is calculated by weighting the terms of the query and the terms of the dataset documents.

6 Information Search and Retrieval

In the Vector Model [16], a document is represented by a set of indexed terms and associated with a normalized value that indicates its degree of relevance to the document. Similarly, the search expression is represented by a numeric vector whose elements represent the degree of relevance of the term to the expression. For the calculation of these weights and creation of the vector model of the rumor dataset, we have implemented a script that creates an inverted list from the .xml files generated by the crawlers, containing each term and the set of news related to it. The lists help in the calculation of the *tf-idf* for each term, where *tf* ("term frequency") is the number of times a given term appears in the text of a document and *idf* ("inverse document frequency") the frequency that a term occurs in any set of documents.

The calculation of *idf* characterizes the term considering the entire corpus, decreasing the weight of terms that occur in more documents and increasing the weight of those that occur rarely. In this way, the *tf-idf* measure is used to assign weight to each element of the vectors representing the corpus documents. The most heavily indexed terms are those that occur very frequently in very few documents. The calculation of *tf-idf*is given by equation 1:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \tag{1}$$

The Figure 3 illustrates the representation of the *eBUSCA*1 search expression (0.2, 0.35, 0.1), along with the documents *DOC*1 (0.3, 0.0, 0.5) and *DOC*2 (0.5, 0.4, 0.3) in a three-dimensional vector space formed by the terms *t*1, *t*2 and *t*3:

	t_1	t_2	t_3
$eBUSCA_1$	**0.2**	**0.35**	**0.1**
DOC_1	**0.3**	**0.0**	**0.5**
DOC_2	**0.5**	**0.4**	**0.3**

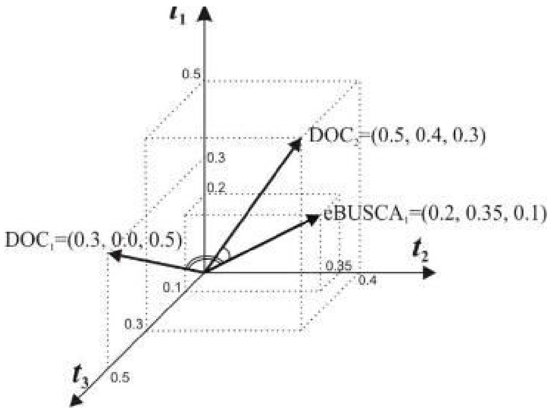


Fig. 3. Search representation.

The partial comparison between document representation and the search expression allows the model to calculate the degree of similarity and create a ranking of documents closer to the query performed. The similarity calculation is given by the equation 2:

$$sim(x, y) = \frac{\sum_{i=1}^{t}(w_{i,x} \times w_{i,y})}{\sqrt{\sum_{i=1}^{t}(w_{i,x})^2} \times \sqrt{\sum_{i=1}^{t}(w_{i,y})^2}} \tag{2}$$

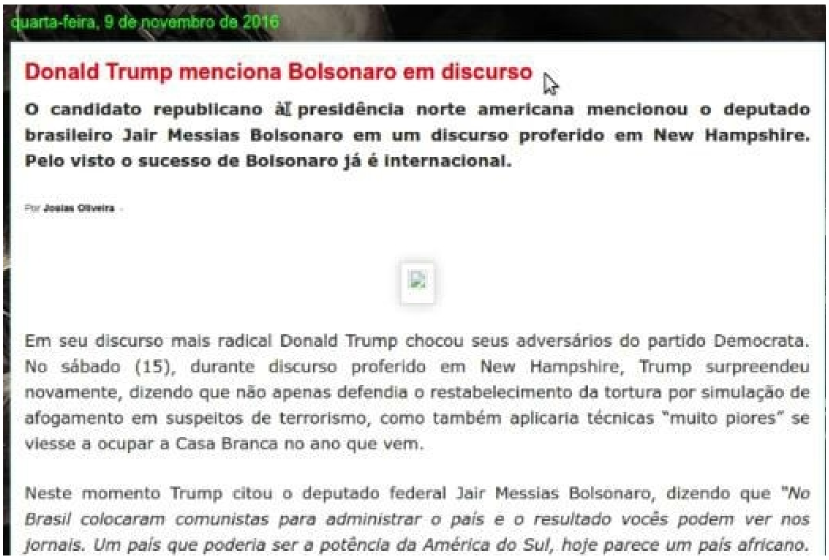
where $w_{i,x}$ is the weight of the i_{th} term of document x and $w_{i,y}$ is the weight of the i_{th} term of the search expression y. Using the representation of the search presented in Figure 2 and applying the calculation of the degree of similarity, we have for $sim(DOC_1, eBUSCA_1) = 0.45$ and $sim\ (DOC_2, eBUSCA_1) = 0.92$, concluding that DOC_2 has greater similarity with the search expression and therefore is displayed first in the list.

When searching for a term through the API, the search script checks the database for the corresponding results, as well as the similarity of all results with the desired term. The output is the five most similar results.

7 Google Chrome Extension

Envisioning a better interface with the internet user, we developed an extension for the Google Chrome browser. So, the user can select a piece of text directly on the page that (s)he is visiting/reading and to search by the veracity of it. When selecting the text, the extension queries the dataset through the API searching and returning the relevant results.

The use of the proposed solution based on the extension developed for Google Chrome can be represented by four distinct moments. In the first moment, as shown in Figure 4, the user is confronted with the news that has an uncertain origin and seems suspicions. The translation of the news' title is: Donald Trump mentions Bolsonaro in his speech. PS: Bolsonaro is a Brazilian deputy, running for the presidential elections in 2018.



quarta-feira, 9 de novembro de 2016

Donald Trump menciona Bolsonaro em discurso

O candidato republicano à presidência norte americana mencionou o deputado brasileiro Jair Messias Bolsonaro em um discurso proferido em New Hampshire. Pelo visto o sucesso de Bolsonaro já é internacional.

Por **Josias Oliveira** -

Em seu discurso mais radical Donald Trump chocou seus adversários do partido Democrata. No sábado (15), durante discurso proferido em New Hampshire, Trump surpreendeu novamente, dizendo que não apenas defendia o restabelecimento da tortura por simulação de afogamento em suspeitos de terrorismo, como também aplicaria técnicas "muito piores" se viesse a ocupar a Casa Branca no ano que vem.

Neste momento Trump citou o deputado federal Jair Messias Bolsonaro, dizendo que "*No Brasil colocaram comunistas para administrar o país e o resultado vocês podem ver nos jornais. Um país que poderia ser a potência da América do Sul, hoje parece um país africano.*

Fig. 4. Page with fakenews (text in Portuguese)

With the extension installed in the browser, the user will have the option to select the suspect content (or part of it) and send it to the analysis tool, which will search for the text in the fakenews dataset. As we can see in Figure 5, this process is facilitated by having the user activate it by clicking the right mouse button.

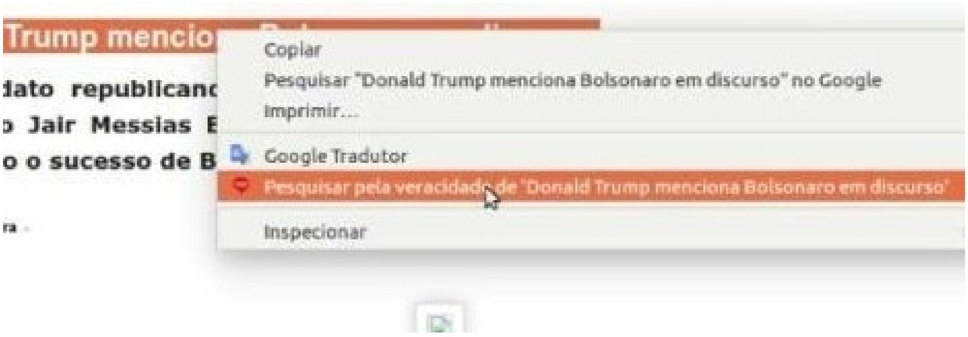

Trump mencio
dato republicanc
Jair Messias E
o o sucesso de B
Copiar
Pesquisar "Donald Trump menciona Bolsonaro em discurso" no Google
Imprimir...
Google Tradutor
Pesquisar pela veracidade de "Donald Trump menciona Bolsonaro em discurso"
Inspecionar

Fig. 5. Searching for rumors

The result of this verification is a list of links where the user may find relevant information about the subject (Figure 5). This list is ordered by the degree of similarity among the documents in the dataset and the text selected and searched.

Finally, by clicking on one link, the user accesses the site that verified the news. Then, the user can get more information about a topic, and acknowledge about a rumor. At this point, the verification cycle ends.

8 Conclusion

Social web shatters the barrier to communicate in anytime and anywhere for people from all walks of life. With a fast and many- to-many model of information dissemination, a new challenge is the recognition of an unreliable content. Additionally, false information can be propagated through social media, resulting in embarrassment, loss or irreversible damages. Consequently, we need methods and techniques to identify rumors, helping in the clarification of doubts and confirmation of facts.

In this article, we present an approach for the identification of fake news. This proposal is composed by: a) a collaborative dataset about fake news, b) an extension (plug-in) for Chrome web browser, which allows the fast verification of content selected by the user, c) an API that can be reused by other applications. Also, we illustrated its use. Through this approach, it was possible to provide a functional solution that allows users to search by rumors in verified sources, dynamically, directly in the browser.

This approach used as referential the main Brazilian sites about fake news, but the process and the code source can be applied in any language and domain. We decided to instantiate this approach for the Brazilian scenario due to the existence offew public solutions. FakePedia, at this moment, is applied to the veracity verification of news related to Zika epidemics and will be used to follow the news during the 2018 election year.

As future works, we will improve the search process through the adoption of paraphrase techniques capable of identifying variations of a rumor, besides documenting the use of the API and implementing mechanisms capable of encouraging society to collaborate with the creation of the public dataset.

9 References

1. Silverman, C. This analysis shows how viral fake election news stories outperformed real new on Facebook. BuzzFeed (Nov. 16, 2016);
2. Gu, L., Kropotov, V., Yarochkin, F.: The fake news machine, how propagandists abuse the internet and manipulate the public. In 1st ed. Trend Micro, p. 81 (2017).
3. Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news.
   Proceedings of the Association for Information Science and Technology, 52, 1-4.
4. Vosoughi, S., & Roy, D. (2017). Rumor Gauge: Predicting the Veracity of Rumors on Twitter. ACM Transactions on Knowledge Discovery from Data (TKDD). 11, 1–38.
5. Rubin, V. L., Chen, Y., & Conroy, N. J. (2015). Deception Detection for News: Three Types of Fake News.
   Proceedings of the Association for Information Science and Technology, 52 (1), 1-
6. Sherman, A. 'Florida Democrats just voted to impose Sharia law on women,' bloggers say.Politifact (May. 08, 2014)
7. Kang, C. In Washington Pizzeria Attack, Fake News Brought Real Guns. New York Times (May. 12, 2016) https://www.nytimes.com/2016/12/05/business/media/comet-ping-pong-pizza-shooting-fake-news-consequences.html
8. Castillo, C., Mendoza, M., & Poblete, B. (2011, March).
   Information credibility on twitter. In Proceedings of the 20th international conference on World Wide Web (pp.675-684). ACM.
9. Shao, C., Ciampaglia, G. L., Flammini, A., & Menczer, F. (2016). Hoaxy: A Platform for Tracking Online Misinformation.
10. Ratkiewicz, J., Conover, M., Meiss, M., Gonçalves, B., Patil, S., Flammini, A., & Menczer, F. (2011). Truthy: Mapping the spread of astroturf in microblog streams. Proceedings of the 20th International Conference Companion on World Wide Web (WWW ’11), 249–252.
11. Resnick, P., Carton, S., Park, S., Shen, Y., & Zeffer, N. (2014). Rumorlens: A system for analyzing the impact of rumors and corrections in social media. In Proc.
    Computational Journalism Conference

12. Metaxas, P. T., Finn, S., & Mustafaraj, E. (2015). Using TwitterTrails. com to Investigate Rumor Propagation. Proceedings of the 18th ACM Conference. Companion on Computer Supported Cooperative Work & Social Computing, 69–72.
13. E-Farsas.http://www.e-farsas.com
14. Boatos.org.http://www.boatos.org
15. É ou não é. https://g1.globo.com/e-ou-nao-e
16. G. Salton, A. Wong, and C.S. Yang. A vector space model for information retrieval. Journal of the American Society for Information Science, 18(11):613-620, Nov. 1975