Qword at CheckThat! 2021: An Extreme Gradient Boosting Approach for Multiclass Fake News Detection

Rudra Sarker Utsha[1], Mumenunnessa Keya [1], Md. Arid Hasan[1] and Md. Sanzidul Islam [1]

[1] *Daffodil International University, Dhaka, Bangladesh*

Abstract

Fake news basically means fabricating a story without verifiable information, source or quote of the story. CheckThat! 2021 Task-3 has two subtasks, subtask 3a and 3b. We participated in Subtask 3a, which is a multi-class fake news classification problem. The goal was to determine whether the main claim of a news article is true, partially true, false or other. We were provided with a dataset of news articles by the organizers which consists of news articles, their titles and the rating of the article. We took advantage of TF-IDF vectorization and proposed an Extreme Gradient Boosting algorithm for our best classification model. The approaches were very interpretative with a highest classification accuracy of 0.57 and highest f1-macro score of 0.54 on the given dataset. We also tried other classification models and got varying results which are simple Logistic Regression Classifiers, Passive Aggressive Classifiers and Random Forest Classifiers.

Keywords

Fake news, XGBoost, Logistic Regression, Passive Aggressive, Random Forest, Classification, Machine Learning.

1. Introduction

The number of social media users has increased so much in a decade that the spread and dissemination of information online is being noticed very fast. Due to this, the number of online abusers increases and it becomes very dangerous, furthermore the result is the dissemination, distribution and reproduction of fake news [12]**.** The term fake news has been given different definitions, one of them being, Fake news is fabricated and misinforming news that is often created with the intention to manipulate people's perceptions of reality [1]. Fake news identification is the ability to verify the accuracy and veracity of information by analyzing various information and the features associated with it. The spread of Fake news poses real-life consequences with serious negative impacts on individuals and society. We have seen in recent years, especially in the 2016 US election how misinformation can impact even the presidential election [4]. This is causing deep concern and the spread of "fake news" is also reflected in the political arena, strengthening its backbone. On the other hand, the main problem is that people's confidence in government institutions is declining and democracy is constantly being undermined by fake news. Identifying fake news is a significant advance to keep fake news from proliferating through social media. Although fabricated news is not a new phenomenon, the detection of fake news has never been more important than today.

As a part of CLEF 2021 CheckThat! Task 3 [21, 22], fake news detection was a subtask, where given the text of a news article, it is required to determine whether the main claim made in the article is true, partially true, false, or other.

[1]CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania
EMAIL: rudra15-10422@diu.edu.bd (A. 1); mumenunnessa15-10100@diu.edu.bd (A. 2); arid.cse.c@diu.edu.bd (A. 3); sanzid.cse@diu.edu.bd (A. 4)
ORCID: 0000-0001-7594-2754 (A. 1); 0000-0001-6399-3669 (A. 2); 0000-0001-7916-614X (A. 3); 0000-0002-9563-2419 (A. 4)

In this paper, we portray the work we performed for this common assignment. For this task a dataset of news articles written in English was prepared manually and shared with the participants by the organizers. We tried different types of machine learning algorithms (i.e XGBoost, Logistic Regression, Passive Aggressive & Random Forest) for this problem and found better results with Gradient boosting algorithms [2] & RF. We have found the best result with both the XGBoost [3] classifier model & RF to classify the fake news in four categories. This multi class classification problem and how we tackled the problem with a ML based classification algorithm is structured in the rest of the paper as follows.

We dedicated section 2 to related work in fake news detection. In section 3 there is an overview of the proposed methodology of our work, section 4 is in the Result & Discussion section and finally the Conclusion is in section 5. The abbreviations we use in our research is given in Table 1.

Table 1
Machine Learning Acronym

Acronym	Definition
ML	Machine Learning
XGBC	Extreme Gradient Boosting Classifier
LRC	Logistic Regression Classifier
PAC	Passive Aggressive Classifier
RFC	Random Forest Classifier
BERT	Bidirectional Encoder Representations from Transformers
RoBERTa	A Robustly Optimized BERT Pretraining Approach
ALBERT	A Lite BERT

2. Related Work

The task of detecting fake news is currently receiving a huge response in research, and researchers are focusing on detecting fake news [21], bots [13] and clickbaits [14], among many others [15]. Fake news is misinformed and fabricated news often created to deliberately mislead or deceive readers [1]. In recent years, we have seen how fake news can even have an impact on the election process [4]. The detection of fake news by linguistic approaches focusing on text properties, such as the writing style and content [9] depend on misleading language and leakage cue to foresee misdirection. Considering the dangerous impact fake news can have in our current society, a lot of research has been conducted on this topic in recent years.

On social media, some features and instances are presented to identify false news that do not work in the same way that they are based. False news is deliberately fabricated and done in such a way that it is not easy to identify them from the subject matter of the text [8].

We can see the significant increase of published papers indexed in the Scopus database regarding fake news. The number of papers on this topic was less than 20 in 2006 to more than 200 in 2018 [10]. The task of detecting fake news has been approached from various perspectives, such as Natural Language Processing (NLP), Data Mining (DM), Social Media Analysis (SMA). In many cases, the classification was reasoned as a binary problem, either the news is fake or real. However, there are cases where the news can be partially false or others. For this reason, several systems capable of multiclass classification were proposed in [8]. The Natural Language Processing field has been tackling the problem of detection and classification with techniques such as Machine Learning and Deep Learning [7], focusing on content-based features that can be extracted from the text content, like linguistic features. In recent research on a relevant subtopic of fake news spreaders detection showed satisfactory results. Very often fake news spreaders are referred to as bots and can spread fake news in completely automated manners. In [5], the author proposed a behavior enhanced deep model (BeDM) that reports an F1-score of 87.32% on a Twitter-related dataset on distinguishing between bots and humans. The proposed model of that research regards user content as temporal text data instead of plain text, fuses content information and behavior information using a deep learning method. Recent research on fake

news detection with Bidirectional Encoder Representations from Transformers model (BERT) has shown to perform very well in [6].

3. Overview of the Proposed Method

Our proposed model is pre-processed with given data (e.g., stopword removal, porter steamer and TF-IDF) by importing the NLTK and then using the model when the data is ready to predict fake news. The model is designed using each group of special features. Our proposed model is illustrated in Figure 1.

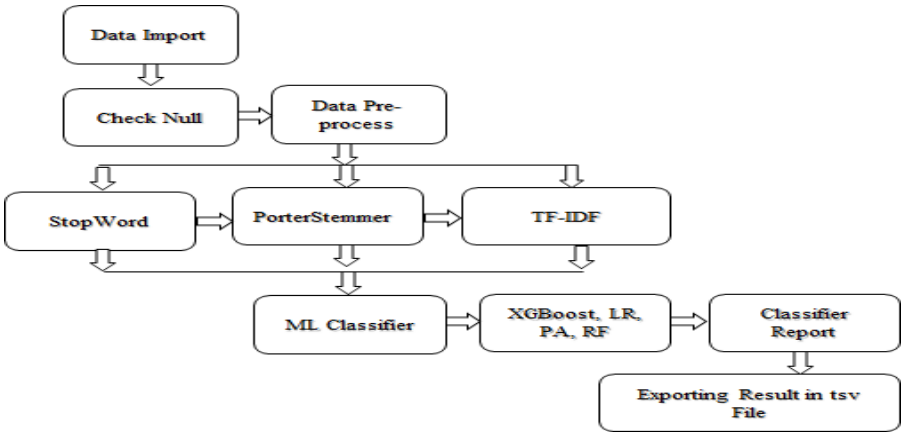

Figure 1: The Proposed Model of the Work

3.1. Dataset Description

For the task, all the participants were given training and a testing dataset by the organizers. The datasets were formatted as csv files containing news articles that were identified manually and the fake news texts were added to the collection with the decision label from the fact checking sites. All of these news articles are written in English language.
Training dataset has 4 columns containing 900 total rows of news articles with their public id and title and the end column is the rating column. The test dataset has a total of 364 rows of articles.
The training dataset overview is in Table 2:

Table 2
Training dataset overview

Class	No. of articles
False	465
True	142
Other	76
Partially false	217

3.2. Data Preprocessing

Text data are unstructured and can't be used to train any ML model without some kind of preprocessing. To prepare the dataset for our proposed models, we used several widely used approaches for text data in the industry.

Punctuation Removing- News articles contain a lot of punctuations, links, digits and special characters that in most cases do not matter in terms of news being fake or true. Moreover, punctuations appear very frequently and lead to high metrics for them while making no impact on the text classification.

Capitalization- Mix of upper- and lower-case words can be useful for a human being when reading something, but for a computational model it is better to use the same type of register level. It does not matter what type of register level to use since it will all be transformed into digits. In this paper we have used lowercase registers.

Lemmatization- Lemmatization is used to reduce the number of words carrying similar kinds of meanings. Stemming can also be used to do the task. While lemmatization reduces word to its morphological root, stemming simply remove the affixes from the word to obtain a root. We used stemming, in this paper for our task.

Removing Stop-words- We can further reduce the number of words from our data while making no impact on how good our model predicts. Stop-words [11] are the words that appear in text extremely frequently while making no impact on the meaning of the text. This is the last step of cleaning our data before creating our bag of words.

After cleaning the texts our goal is to make the data trainable as ML models can't really work with text or strings. So, we convert all our text into numbers for it to be trainable. We do so by vectorizing our text data with the TF-IDF vectorization method and making our final trainable data.

3.3. ML Algorithms

The more the machine learning algorithm comes in contact with the data, the better it will perform. The word "learning" in machine learning means that processes change over time as data is processed and at the same time the way people process data changes.

XGBoost This is highly credited by the machine learning practitioners and popular among ML competitors. XGBoost is an implementation of gradient boosted decision trees that were designed for speed and performance. This framework can be found for all popular data analysis languages and performs considerably well for multiclass classification problems. XGBoost is used for both regression and classification problems thus expected to perform well for our classification task. XGBoost decision tree being a gradient boosted tree, usually outperforms random forest. More about this method is in the original paper, written by Tianqi Chen and Carlos Guestrin [3].

Logistic regression This is a well-known statistical model that in its default form is limited to predict binary classification problems. In our case it can be used with some kind of extension like one-vs-rest to allow it to classify multiclass classification problems. However, the classification problem first transformed into multiple binary classification problems. Basically, the idea behind the method is to calculate the probability of a news article to be true vs anything else. And similarly calculating the probability of a news article to be False vs anything else and so on. For the reason that the probability function (1) has a logistic form, the model also got the name Logistic Regression:

$$f(z) = \frac{1}{1 + e^{-x}} \qquad (1)$$

Here, z is a set of model input factors, in our case these are vectors of count vectorized matrix. More about this regression can be found in [17] written by Cramer.

Passive Aggressive Another classifier we implemented was a Passive Aggressive classifier. It is an incremental learning algorithm and the concept is that the classifier adjusts its weight vector for every misclassified training sample it receives trying to correct it. The passive part is to leave the model as it is if the prediction is correct and the aggressive part is to make changes to the model when the prediction is incorrect. That is, some changes to the model may correct it. Passive Aggressive classifiers can be used in binary classification, multiclass class classification, and uniclass classification problems. More on this topic can be found in [18].

Random Forest A very well-known and extremely frequently used by machine learning competitors, this is an ensemble learning method used for both classification and regression tasks.

Random forest constructs an ensemble of decision trees (Forest) to get a stable and more accurate prediction. The idea behind this combination of tree predictors is that as the trees in the forest become large the generalization error for the forest converges to a limit. Detailed information about this method can be found in the paper [20] by Breiman.

4. Result and Discussion

This section will discuss all the accuracy and f1-macro scores and some parts of classification reports of fake news detection. The accuracy and macro-average F1 score we have achieved through the application of XGBoost algorithm and other algorithms are shown in Table 3 and the results of classifiers in terms of precision, recall and F-1 standard is presented in Table 4. And with the ID (public_id) that we represented our dataset through preprocessing, we will see the predicted ratings (predicted_rating) of XGBoost, LR, PA & RF of the first 5 texts from the test dataset in Table 5 to see if the model can accurately detect fake news.

Table 3
Accuracy Table

Model	Accuracy	F1-macro Avg
XGboost	0.571	0.543
Logistic Regression	0.412	0.272
Passive Aggressive	0.542	0.489
Random Forest	0.534	0.502

Table 3 shows the classification accuracy score and f1-macro average score of four models. Table 4 represents the precision, recall and f1-score. In the first column of the table, we've taken the algorithms and in the second one there we place the state of the dataset which provides for fake news as if the data is False or True or Other or Partially False. We have found out the precision, recall and f1-score. For different classifiers, all of the values sometimes perform well and some are given quite small values.

Table 4
Performance overview

Algorithms	**State**	**Precision**	**Recall**	**F1 Score**
XGBoost	False	0.49	0.91	0.64
	True	0.42	0.26	0.32
	Other	0.76	0.55	0.64
	Partially False	0.79	0.45	0.57
Logistic Regression	False	0.36	0.92	0.52
	True	0.35	0.12	0.18
	Other	0.00	0.00	0.00
	Partially False	0.75	0.26	0.39
Passive Aggressive	False	0.56	0.85	0.68
	True	0.35	0.26	0.30
	Other	0.38	0.53	0.44
	Partially False	0.72	0.43	0.54
Random Forest	False	0.42	0.98	0.59
	True	0.71	0.15	0.25
	Other	0.91	0.53	0.67
	Partially False	0.83	0.36	0.50

Table 5
Testing with the Test Data

	public_id	XGBC_predicted_rating	LRC_predicted_rating	PAC_predicted_rating	RFC_predicted_rating	True_rating
0	81a67c96	partially false	partially false	partially false	partially false	partially false
1	6e5ec6fb	false	false	false	false	false
2	d9cd4895	false	false	false	false	false
3	4a1a9b9f	false	false	false	false	false
4	6d16fa40	false	false	false	false	false

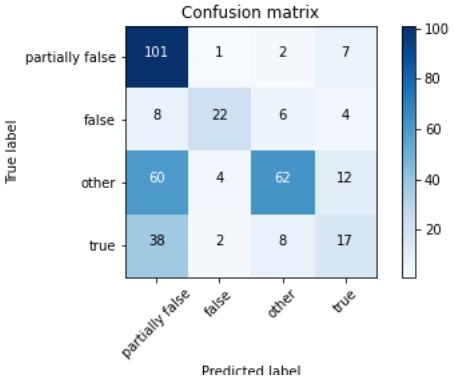

Figure 2: XGBoost

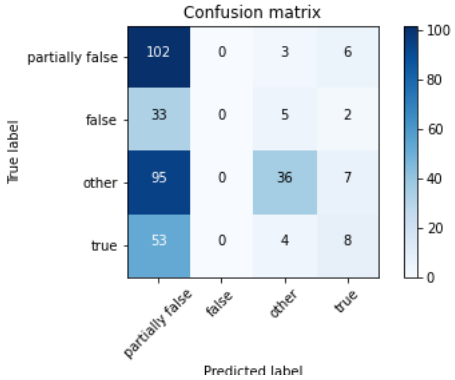

Figure 3: Logistic Regression

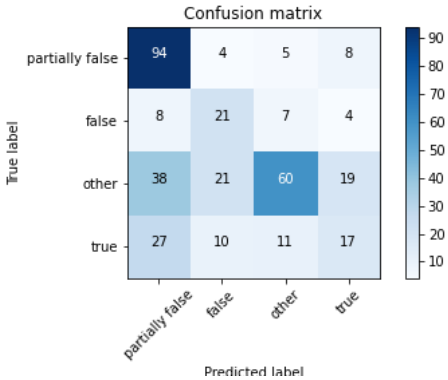

Figure 4: Passive Aggressive

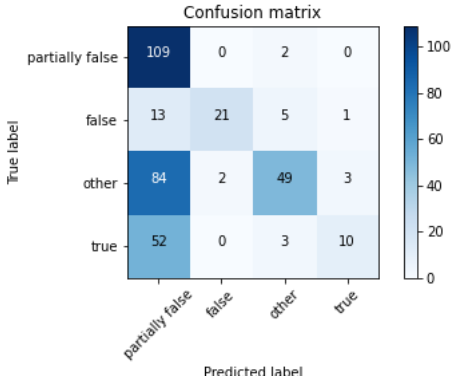

Figure 5: Random Forest

Figures 2,3,4,5 are for all the confusion matrix graphical representations of the algorithms, where in the X-axis has the predicted label and the Y-axis has the True label. Also, both X & Y axis there are True, False, Others and Partially False labels present.

Previously, working with these same models we tried a different approach to vectorize our data with the Count-Vectorization method and set max_feature to 5000 and got really bad results.The XGBoost classifier were showing a f1-macro score of 0.15 on test data even though it was showing good result on training data. Finally, we vectorized our dataset with TF-IDF vectorization and set max_feature to 18256.We also changed ngram_range from (1,3) to (1,1). These small changes and a different approach in vectorizing the dataset bumped our accuracy from 0.28 to 0.54 and f1-macro measure from 0.15 to a 0.52 on the test dataset.

Our best performing XGBClassifier has these hyperparameters in Table 6.

Table 6
Hyperparameters

learning Rate	0.3
Max_depth	15
min_child_weight	7
gamma	0.1
colsample_bytree	0.7

With the hyperparameters in Table 6, we got a slight improvement of performance, our accuracy went to 0.57 and F1-macro score to 0.54 and these are the final results from our best performing XGB model, in fact any ML model that we have tried. The next best model is Random Forest classifier with an accuracy of 0.53 and F1-macro score of .50. In third place is Passive Aggressive classifier with an accuracy of 0.54 and F1-macro score of 0.49. And at last, as expected the simple Logistic Regression classifier did a poor job of classifying news articles. It had an accuracy of 0.41 and F1-macro score of 0.27.

5. Conclusion

This paper showcases our work on fake news detection with various machine learning algorithms and provides the best results which we found. While our models were good at classifying false and partially false and other classes, all of the models were performing really bad at predicting true classes. We suspect it was due to the nature of our dataset, a high number of samples present in false and partially false classes made the training of the model very biased towards these classes. However, the overall results were not unsatisfactory as the model was really good at classifying the false news and partially false news, and that is a step closer towards preventing the spread of false news. We think it is likely that our models could perform better once we find the optimal hyperparameters, but this needs more time and computing power. As per our expectation, Random Forest classifier did well but the XGBoost model did overall better than Random Forest by a thin margin. But as we stated earlier, by tweaking the hyperparameters of both of the models we could get better results. We also observed that the Passive Aggressive Classifier was also very good at text classification tasks. Besides that, in future we might also try making a bigger and more balanced dataset where every class contains a similar number of samples and then perform classification with Deep Learning models such as Transformers based models like BERT, RoBERTa, ALBERT models. We might also try more traditional LSTM and CNN models for comparison. And our belief is that with a bigger dataset, these Deep Learning models would show a better classification performance.

6. Acknowledgements

This is unbelievable support we've got from some of the faculty members and seniors of DIU NLP and ML Research Lab to continue our whole research flow from the beginning. We acknowledge Dr. Firoj Alam for guiding us and informing the secretes of research workshops. Also, we're thankful to Daffodil International University for the workplace support and the academic collaboration in some cases. Dr. Touhid Bhuiyan and Dr. Sheak Rashed Haider Noori also supported us with guidance, motivation, and advocating in institutional supports. Lastly, for sure we're thankful to our God always for every fruitful work with our given knowledge.

7. References

[1] Lazer, D.M., Baum, M.A., Benkler, Y., Berinsky, A.J., Greenhill, K.M., Menczer, F., Metzger, M.J., Nyhan, B., Pennycook, G., Rothschild, D., et al.: The science of fake news. Science 359(6380), 1094–1096 (2018).

[2] Alexey Natekin, Alois Knoll: Gradient Boosting Machines, A Tutorial. Front. Neurorobot., 04 December 2013; doi: https://doi.org/10.3389/fnbot.2013.00021

[3] Tianqi Chen, Carlos Guestrin: XGBoost: A Scalable Tree Boosting System. 2016 ACM. ISBN 978-1-4503-4232-2/16/08; doi:http://dx.doi.org/10.1145/2939672.2939785

[4] Bovet, A., Makse, H.A.: Influence of fake news in twitter during the 2016 US presidential election. Nature communications 10(1), 1–14 (2019).

[5] Cai, C., Li, L., Zengi, D.: Behavior enhanced deep bot detection in social media. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). pp. 128–130. IEEE (2017).

[6] Heejung Jwa 1 , Dongsuk Oh 2 , Kinam Park 3 , Jang Mook Kang 4 and Heuiseok Lim 1,*: exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT). *Appl. Sci.* **2019**, *9*(19), 4062;doi:https://doi.org/10.3390/app9194062 .

[7] Ruchansky, N., Seo, S., Liu, Y.: Csi: A hybrid deep model for fake news detection. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 797–806 (2017).

[8] Rashkin, H., Choi, E., Jang, J.Y., Volkova, S., Choi, Y.: Truth of varying shades: Analyzing language in fake news and political fact-checking. In: Proceedings of the 2017 conference on empirical methods in natural language processing. pp. 2931–2937 (2017).

[9] Pérez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R.: Automatic detection of fake news. In: Proceedings of the 27th International Conference on Computational Linguistics. pp. 3391–3401. Association for Computational Linguistics, Santa Fe, New Mexico, USA (Aug 2018), https://www.aclweb.org/anthology/C18-1287.

[10] Zhou, X., Zafarani, R.: Fake news: A survey of research, detection methods, and opportunities. arXiv preprint arXiv:1812.00315 (2018).

[11] Luhn, H. P.:Keyword-in-Context Index for Technical Literature (KWIC Index). American Documentation. 11 (4): 288–295. CiteSeerX 10.1.1.468.1425. https://doi.org/10.1002/asi.5090110403.

[12] Despoina Mouratidis, Maria Nefeli Nikiforos and Katia Lida Kermanidis, Deep Learning for Fake News Detection in a Pairwise Textual Input Schema, Computation 2021, 9, 20. https://doi.org/10.3390/computation9020020 https://www.mdpi.com/journal/computation.

[13] Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Detecting automation of twitter accounts: Are you a human, bot, or cyborg? IEEE Transactions on Dependable and Secure Computing 9(6), 811{824 (2012).

[14] Anand, A., Chakraborty, T., Park, N.: We used Neural Networks to Detect Clickbaits: You won't Believe what Happened Next! In: Proceedings of the 2017 European Conference on Information Retrieval. pp. 541{547. ECIR '17 (2017).

[15] Francisco Rangel1, Anastasia Giachanou2, Bilal Ghanem2;1 and Paolo Rosso2, Overview of the 8th Author Proling Task at PAN 2020: Proling Fake News Spreaders on Twitter, CLEF 2020, 22-25 September 2020, Thessaloniki, Greece.

[16] M. Rahimzadeh and A. Attar, “A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2,” Informatics in Medicine Unlocked, vol. 19, p. 100360, 2020.

[17] Cramer, J. S. (2002). The origins of logistic regression (PDF) (Technical report). 119. Tinbergen Institute. pp. 167–178. https://doi.org/10.2139/ssrn.360300.

[18] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online Passive-Aggressive Algorithms. Journal of Machine Learning Research. 7 (12/1/2006), 551–585.

[19] W. Lin, Z. Wu, L. Lin, A. Wen and J. Li, "An Ensemble Random Forest Algorithm for Insurance Big Data Analysis," in IEEE Access, vol. 5, pp. 16568-16575, 2017, doi: 10.1109/ACCESS.2017.2738069.

[20] Breiman, L. Random Forests. Machine Learning 45, 5-32 (2001), https://doi.org/10.1023/A:1010933404324.
[21] Nakov P. et al. (2021) The CLEF-2021 CheckThat! Lab on Detecting Check-Worthy Claims, Previously Fact-Checked Claims, and Fake News. In: Hiemstra D., Moens MF., Mothe J., Perego R., Potthast M., Sebastiani F. (eds) Advances in Information Retrieval. ECIR 2021. Lecture Notes in Computer Science, vol 12657. Springer, Cham. https://doi.org/10.1007/978-3-030-72240-1_75.
[22] Gautam Kishore Shahi, Julia Maria Struß, and Thomas Mandl. Task 3: Fake News Detection at CLEF-2021 CheckThat!, apr (2021), doi: https://doi.org/10.5281/zenodo.4714517.