DataBees at CheckThat! 2024: Check-Worthiness Estimation

Notebook for the CheckThat! Lab at CLEF 2024

Tanisha Sriram[1,*], Yadushree Venkatesh[1], Sowmya Anand[1] and Bharathi B[1]

[1]SSN College Of Engineering

Abstract

In today's world, it is very essential for us to identify claims in social media posts or transcriptions in order to combat misinformation. The task 1[1] from CheckThat! lab[2] has enabled us to provide a solution to this problem. Through this project, we propose a machine learning-based approach to evaluate if the given claims are worth fact-checking. By employing various pre-trained models like BERT, RoBERTa, and language-specific models such as AraBERT for Arabic, and traditional classifiers like MultinomialNB and Logistic Regression, the system is designed to work across three languages i.e. English, Dutch, and Arabic. Through training and experimentation using the given datasets, the models with the highest F1 scores were identified for each language. We observed that the best F1 scores were given by DistilBERT and BERT for English, AraBERT for Arabic, and BERT for Dutch. These models were then used to predict the check-worthiness of claims in unseen test data, which demonstrated the effectiveness of the proposed solution.

Keywords

fact-checking, pre-trained models, traditional classifiers

1. Introduction

Where information flows rapidly, in today's digital age, across various platforms such as social media, email, and messaging apps, distinguishing between real and fake information has become extremely challenging. Misinformation can spread quickly, leading to confusion, misunderstandings, and even harmful consequences. Therefore,through this project, we aim to develop a worthiness checking solution to enhance a user's ability to check the credibility of information. The issue of spreading of misinformation is greatly magnified due to the sheer volume and speed at which information is being spread. Social media, for instance, allows anyone to post and share content without checking the credibility of the content. This democratization of information has its benefits, but it also has the chances that there may be sharing of misinformation Most people lack the tools or skills to evaluate the information that they come across. They may unknowingly spread false information because it aligns with their beliefs or because they assume that it is accurate without verification. The psychological phenomenon of "confirmation bias" aggravates this issue, as people tend to agree with the information that confirms their preexisting beliefs and disregard information that contradicts them. The consequences of misinformation can be severe. In the realm of health , if there is false information regarding a certain vaccine, people may tend to avoid the process which may lead to spread of the disease which otherwise would have been controlled by the vaccine . Similarly, in politics, misinformation can influence public's opinion and voting , which may eventually lead to bad results. Misinformation about disasters can create unnecessary panic and inappropriate responses from the public.

The aim of this task is to determine whether a claim in a tweet transcription is worth fact-checking. Generally, this decision is made by professional fact-checkers or by human themselves who have

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

*Corresponding author.

†These authors contributed equally.

tanisha2310538@ssn.edu.in (T. Sriram); yadushree2310494@ssn.edu.in (Y. Venkatesh); sowmya2310543@ssn.edu.in (S. Anand); bharathib@ssn.edu.in (B. B)

to answer several auxiliary questions such as “does it contain a verifiable factual claim?”, and “is it harmful?”, before deciding on the credibility of the content. We used the machine learning approach by training the model on the train data sets and finding the best F1 score over the positive class of the classification.

Three languages have been used (English, Dutch and Arabic) and various pre-trained models along with some traditional methods were used depending on the language. The following were the models that were used. BERT, RoBERTa, XLM-RoBERTa, DistilBERT, AlBERT, Electra, AraBERT (Arabic) and BERTje (Ducth). Traditional classifiers such as MultinomialNB, SVC-linear and Logistic Regression were also used. In the realm of natural language processing (NLP) and information authenticity verification, we faced challenges while dealing with Dutch and Arabic, which have very different linguistic structures. These challenges become more and more pronounced as the existing resources and tools are mostly designed for English. Pre-trained models for languages such as Dutch and Arabic are less available compared to English. While English has many pre-trained models like BERT, GPT, and their variants, non-English languages often lack these many resources. Arabic uses a completely different script from the Latin based languages like English and Dutch. Arabic script is cursive and includes letters that change its shape based on their position in a word. This increases the complexity to text processing tasks, such as tokenization and character recognition. The Arabic script is known for its rich morphology and a single root can be used to generate several other related words. This is quite different from English and Dutch since these languages have less structural variations. Achieving high F1 scores was quite challenging in Arabic and Dutch.

Hence, the challenges of developing natural language processing models for languages like Dutch and Arabic were significant. By focusing on the above stated areas, we can improve the performance of models in many languages, enabling more accurate and reliable information verification across diverse languages.

The second section contains information about the datasets used for each language. Third section is the related work section where the reference of relevant documents for similar tasks will be discussed. Fourth section contains the methodology with the complete process of the task starting from the pre-processing to various models that were used. The ranks obtained will be presented in the results section.

2. Dataset

We had been given the dataset which consisted of the training set, the development (dev) set, and the test set. The training set was the largest part of the dataset and was used to train our ML models. It had labeled text with their respective class labels . These samples were then used for learning by the model, and the patterns and relationships in the data were taken in by the algorithms of our model. The development set, or dev set was the intermediate step in the modeling process. This was the validation process and helped to fine-tune our model. Finally, the test set, which is the evaluation stage, containing unlabeled data were passed to the model and the predictions were generated. This was compared to the ground truth labels. These are then used to measure overall performance and generalization capabilities of our solution. Each instance is composed of only text, which could come from a tweet, the transcription of a debate or the transcription of speech. The English dataset has the sentence id, the text as well as the class label. Similarly for Dutch and Arabic, we were provided with datasets that contained the tweet id, tweet url, tweet text and the class label. The class labels were provided for the train and dev sets while the labels had to be predicted for the test set using our project. The model with the best f1 score over the positive class was evaluated with the dev set.

3. Related Work

Check-worthiness estimation has been implemented in various studies. Copenhagen [3] was a team who obtained a MAP of 0.155 in CheckThat! 2018 lab. The sentence was represented using word

embedding alone with POS tags. It was used as input to an RNN with GRU memory units, from which the output from each word was aggregated using attention, and a fully connected layer, from which the output was predicted using a sigmoid function.

In ClaimBuster[4], the authors used the transcripts of all of the US presidential debates that were manually annotated. The authors proposed a SVM-based model with sentence-level features such as sentiment, length, TF-IDF, POS-tags, and Entity Types.

Gencheva et al. integrated several context-aware and sentence-level features totrain both SVM and Feed-forward Neural Networks[5]. This approach outperforms the ClaimBuster system in terms of MAP and precision.

Patwari al [6] predicted whether a sentence would be selected by a fact-checking organization using a boosting-like model. Similarly, Vasileva al.[7] used a multi-task learning neural network that predicts whether a sentence would be selected for fact-checking by each individual fact-checking organization (from a set of nine such organizations).

NUS-IDS team was one of the top teams in the subtask related to detecting check-worthiness of tweets in 2022[8]. They explored the feasibility of adapting sequence-to-sequence models for detecting check-worthy social media content in a multilingual environment (Arabic, Bulgarian, Dutch, English, Spanish and Turkish) provided in the competition. They ranked first in 4 out of 6 languages at Checkthat! 2022 Task 1A.

AI Rational team[9] in the same subtask employed three different pre-trained transformer models: BERT, DistilBERT (a distilled version of BERT) and RoBERTa. All models used have been taken from huggingface. Fine-tuning of parameters was done on DistilBERT model since it was the fastest one to train.

PoliMi-FlatEarthers team [10] used a generative pre-trained GPT-3 model in English. They showed how much larger GPT-3 models, despite being developed primarily for text-generation, outperformed previous language models on the task of automated claim detection on the 2022 CheckThat! Challenge dataset. Not only that, they also showed that GPT-3, while designed for handling mainly English tasks, can maintain competitive performances on other languages as well.

Glasgow Terrier at CLEF CheckThat! 2019[11] proposed to represent each sentence using their mentioned entities using a TF-IDF representation. They used a SVM classifier to predict the check-worthiness of each sentence. Their approach ranked 4th out of 12 submissions. Their experiments showed that the pronouns and coreference resolution pre-processing procedure they used as part of their approach does improve the effectiveness of sentence checkworthiness prediction. Furthermore, their results show that entity analysis features provide valuable evidence for this task.

Zindex[12] used the oversampling technique to balance the dataset and applied SVM and Random Forest (RF) with TF-IDF representations. They also used BERT multilingual (BERT-m) and XLM-RoBERTa-base pre-trained models for the experiments. They used BERT-m for the official submissions and our systems ranked as 3rd, 5th, and 12th in Spanish, Dutch, and English, respectively. In further experiments, their evaluation showed that transformer models (BERT-m and XLM-RoBERTa-base) outperformed the SVM and RF in Dutch and English languages where a different scenario is observed for Spanish.

Hence various approaches have been used such as SVM classifier, feed-forward neural networks, POS-tags, RNN, pre-trained models like BERT, DistilBERT and RoBERTA, GPT-3 model, random forest and graident boosting and they were proven successful.

4. Methodology

4.1. Pre-Processing

The following are some very important steps that were taken in the pre-processing phase to increase the quality of the textual data before submitting them to machine learning models. First, all the text was converted into lowercase for further uniformity and getting rid of word duplication based on case differences. Punctuation marks were eliminated to reduce the noise and make the meaningful content

Table 1
F1 scores of various pre-trained models.

Models	English	Dutch	Arabic
BERT	0.78	0.64	0.64
RoBERTa	0.62	0.47	-
XLM-RoBERTa	0.70	0.48	-
DistilBERT	0.78	-	-
ALBERT	0.72	-	-
Electra	0.74	-	-
AraBERT(Arabic)	-	-	0.67
BERTje(Dutch)	-	0.52	-

Table 2
F1 scores of traditional classifiers.

Models	English	Dutch	Arabic
MultinomialNB	0.63	0.38	0.58
SVC-linear	0.63	0.44	0.52
Logistic regression	0.57	0.43	0.42

more distinct. Stemming and Lemmatization were applied in order to normalize the words by bringing them back to the root, thereby reducing the vocabulary size to increase the efficiency of the model.

Further, common words with small semantic value, like articles and prepositions, which are the stop-words, were removed to bring into sharp focus the content-bearing words. While emojis add an expressive element to the text, they have been taken out because semantically they add nothing to factual content and might introduce noise into the dataset.

Second, binary encoding was carried out; standardized responses converted yes and no into 1s and 0s, respectively. These binary features allow uniformity in representation across the dataset.

All these pre-processing steps play an important role in refining textual data, hence ensuring machine learning models can effectively catch the semantic content and make accurate predictions regarding the check-worthiness of claims.

4.2. Models

BERT(Bidirectional Encoder Representations from Transformers) developed by Google AI Language, has greatly improved the Natural Language Processing (NLP). BERT employs an encoder-only architecture. Its bidirectional Transformer architecture process text sequentially, either from left to right or right to left improving its understanding of language. After its breakthrough, BERT led to the creation of successive models, like RoBERTa, ALBERT, and DistilBERT, building on the architecture and adopting strategies to improve the model performance on various NLP tasks. All the models have elevated the performance, efficiency, and scalability of the NLP models. It gave us an f1 score of 0.78 in English, 0.64 in Dutch and 0.64 in Arabic as shown in table 1.

RoBERTa, an evolution of BERT, working towards enhancing its language understanding tasks. It takes off the next sentence prediction task and uses a dynamic mask during training. RoBERTa demonstrates better performance on most benchmarks of NLP. The bidirectional context encoding scheme is aptly suited for subtle understandings of textual data that a task might require, such as sentiment analysis, text classification, and question answering. RoBERTa's success in capturing contextual information from large-scale corpora enables it to generate very accurate representations of language semantics. It gave us an f1 score of 0.62 in English and 0.47 in Arabic as shown in table 1.

XLM-RoBERTa furthers the capabilities of RoBERTa towards other languages. The cross-

lingual pretraining techniques in a transformer-based architecture mean that XLM-RoBERTa could do well in tasks calling for cross-lingual understanding, such as machine translation, cross-lingual document classification, and multilingual sentiment analysis. Its competence to deal with diverse languages makes it an asset for applications with global communication requirements that understand and process multilingual content. It gave us an f1 score of 0.70 in English and 0.48 in Ducth as shown in table 1.

DistilBERT, a distilled version of BERT, addresses the computational and memory challenges associated with large transformer models. By reducing the model size and employing knowledge distillation techniques, DistilBERT retains the essence of BERT's contextual understanding while significantly reducing computational resources. This makes it useful for use in those environments where resources are constrained. It gave us an f1 score of 0.78 in English as shown in table 1.

ALBERT also known as "A Lite BERT", further enhances the scalability and efficiency of transformer models. By implementing factorized embedding parameterization and cross-layer parameter sharing, ALBERT achieves comparable performance to BERT while significantly reducing memory and computational requirements. This makes it suitable for large-scale NLP tasks such as document classification, text generation, and language modeling. ALBERT's efficiency and scalability make it an attractive option for applications requiring high-performance NLP models deployed at scale. It gave us an f1 score of 0.72 for English as shown in table 1.

ELECTRA stands for Efficiently Learning an Encoder that Classifies Token Replacements Accurately. It's a transformer model coming from Google. It differs from traditional masked language models—MLMs, like BERT, predict the actual masked tokens in the pre-training phase; ELECTRA uses the replaced token detection task. The input text gets corrupted by replacing some of the tokens with plausible alternatives generated by a small generator model. Then, the main model—the discriminator—gets trained to detect these replacements. Due to this approach, ELECTRA is more sample-efficient and can even reach competitive or superior performance to BERT with much less computation. It gave us an f1 score of 0.74 in English as shown in table 1.

AraBERT is a language model developed explicitly for Arabic text by the Applied Artificial Intelligence Institute at the University of Sharjah. It is based on the BERT architecture, but it is modified and fine-tuned explicitly for the particularities of the Arabic language. AraBERT has been trained on a large corpus of Arabic text and has shown impressive performance over many NLP tasks, such as text classification, named entity recognition, and sentiment analysis. That's what makes it a valuable resource in Arabic NLP research and applications. It gave us an f1 score of 0.67 for Arabic as shown in table 1.

BERTje is the Dutch version of BERT developed by the University of Amsterdam and Tilburg University. Based on the BERT architecture, like AraBERT, it is fine-tuned to the Dutch language. It is pre-trained on a large corpus of Dutch text and shows good performance for Dutch NLP tasks like text classification, question answering, and language understanding. BERTje has become for NLP research in Dutch and other applications because of its ability to capture the intricacies of Dutch language. It gave us an f1 score of 0.52 for Dutch as shown in table 1.

Multinomial Naive Bayes (MultinomialNB) is a probabilistic classifier based on Bayes' theorem with the assumption that every feature is independent of all others. It creates class probabilities according to the frequency of features. This model is, therefore, very apt for tasks in NLP such as sentiment analysis and document classification, since it deals efficiently with large and very sparse feature spaces. However, the major benefits of Multinomial NB are the ease in its implementation and the fact that it works well with very modest computational overhead, which makes this classifier a great baseline in many text classification research projects. It gave us an f1 score of 0.63 in English, 0.38

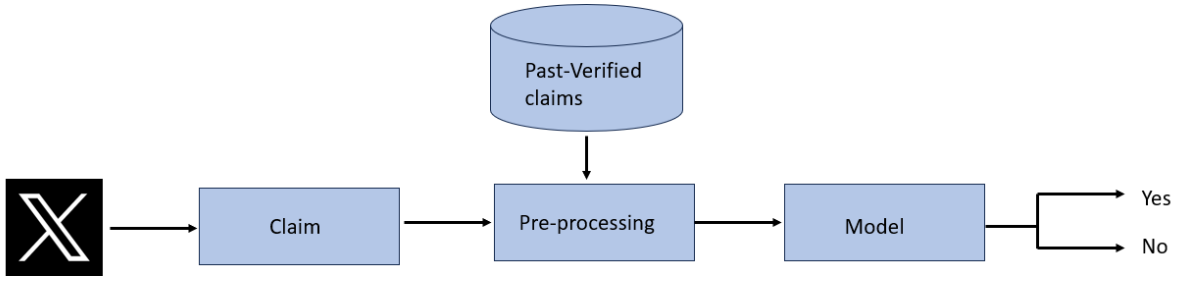

Figure 1: Check-Worthiness Estimation

in Dutch and 0.58 in Arabic as shown in table 2.

Support Vector Classification with a linear kernel (SVC-linear) is yet another traditional machine learning algorithm commonly used for text classification tasks. The work done by SVC-linear is to find the best hyper-plane that separates the classes in a high-dimensional space. It uses a linear kernel to compute the dot products between feature vectors, and it has proved very effective in text classification when dealing with high-dimensional feature spaces. It is useful for not being prone to overfitting, more so in combination with apt regularization techniques. It can treat nonlinear relationships through kernel functions, making it flexible with the capacity to model complicated data distributions. It gave us an f1 score of 0.63 in English, 0.44 in Dutch and 0.52 in Arabic as shown in table 2.

Logistic Regression provides an estimate of the likelihood of a binary outcome as a function of input features by applying the logistic function to a linear combination of these features. Thus, it has been immensely employed for all those tasks in NLP where interpretability and efficiency are prime requirements, such as in sentiment analysis and spam detection. Advantages of Logistic Regression include the simplicity of returning probabilities that can easily be understood as class predictions, making the interpretation of results easy. It works decently where the relationship between features and the target variable is linear or approximately linear, and it's often a stable choice used as a baseline model in NLP research. It gave us an f1 score of 0.57 in English, 0.43 in Dutch and 0.42 in Arabic as shown in table 2.

5. Results and Analysis

5.1. Performance Metrics

An overview of a machine learning model's performance on a set of test data is provided via a confusion matrix. It is a way to show how many instances, depending on the model's predictions, are accurate and inaccurate. It is frequently used to assess how well classification model, which seek to assign a categorical label to each instance of input, perform. The number of instances that the model generated on the test data is shown in the matrix.
True positives (TP): occur when the model accurately predicts a positive data point.
True negatives (TN): occur when the model accurately predicts a negative data point.
False positives (FP): occur when the model predicts a positive data point incorrectly.
False negatives (FN): occur when the model predicts a negative data point incorrectly.

The precision, recall, and F1-score macro averages are used to evaluate this task. The metrics are computed for each class separately, and the averages are then used to provide equal significance to each

Table 3
Rank list for Arabic.

Team	F1
visty	0.569
teamopenfact	0.557
DSHacker	0.538
TurQUaz	0.533
SemanticCUETSync	0.532
mjmanas54	0.531
FiredfromNLP	0.530
Madussree	0.530
pandas	0.520
hybrinfox	0.519
Mirela	0.478
DataBees	0.460
Baseline	0.418
JUNLP	0.212

class. Precision in classification relates to the likelihood that the classification was done correctly. It is the proportion of points that are accurately classified to all points that have been projected to belong to that class.

$$Precision = TP/(TP + FP) \tag{1}$$

Conversely, recall provides an estimate of the number of correctly performed classifications of a type. It is the ratio of a class's correctly classified points to the total of that class's correctly and wrongly categorized points.

$$Recall = TP/(TP + FN) \tag{2}$$

The F1-score is a weighted average of recall and precision that is typically employed when there is a significant class imbalance or when both metrics need to be balanced.

$$F1score = 2((Precision)(Recall))/(Precision + Recall)) \tag{3}$$

5.2. Results

To objectively assess how well the models that were fitted to the training dataset performed, the test dataset was utilized. As with this assignment, the macro averages, which comprise precision, recall, F1-score and confusion matrix, were employed in addition to accuracy as performance indicators for analysis.

Taking into consideration the dataset containing tweets in English, it was evident that DistilBERT and BERT were the best classifiers having an F1 score of 0.78 on the positive class. This was followed by Electra and Albert having an F1 score of 0.74 and 0.72 on the positive class respectively.

This run secured the 18th rank in Task 1 English which used the dataset containing tweets in English. The models performed on the test set with a macro F1 score of 0.78.

The reason BERT performs so highly in English tweets' sentiment analysis tasks lies in being bidirectional. In a large-scale English text pretraining, specific deep context is gained, which enables the extraction of subtle expressions of the language, slang, and other contextual clues in the tweets, making it very effective in classifying sentiments.

While slightly smaller and faster than BERT by a factor of distillation techniques, DistilBERT retains much of the performance by distilling the essential knowledge from BERT pre-training. Much of the ability that BERT derives to interpret complex language structures and sentiments in tweets is still

Table 4
Rank list for Dutch.

Team	F1
TurQUaz	0.732
DSHacker	0.730
visty	0.718
Mirela	0.650
Zamoranesis	0.601
FCRUG	0.594
teamopenfact	0.590
hybrinfox	0.589
mjmanas54	0.577
DataBees	0.563
JUNLP	0.550
FiredfromNLP	0.543
Madussree	0.482
Baseline	0.438
pandas	0.308
SemanticCUETSync	0.218

retained with this model, still making this very practical when computational efficiency is of value without a compromise on performance.

When the dataset of tweets in the language Dutch was trained on a variety of models, we observed that BERT had the best performance with an F1 score of 0.64 on the positive class. DistilBERT came second in terms of accuracy with an F1 score of 0.52 on the positive class.

This run secured the 10th rank in Task 1 Dutch which used the dataset containing tweets in Dutch as shown in table. The model performed on the test set with a macro F1 score of 0.64.

The success of BERT in conducting sentiment analysis for Dutch tweets can be attributed to its pre-training on enormous Dutch texts that enable the model to get a feel for the specific nuisances of the use of the Dutch language. On morphology, syntax, and contextual changes typical of Dutch tweets, BERT does well; hence, it generalizes with high accuracy on tasks like sentiment classification.

Considering the results of the analysis on the Arabic tweet dataset, it is possible to conclude that AraBERT was the best classifier with an F1 score of 0.67 on the positive class. This was followed by MultinomialNB which had an F1 score of 0.58 on the positive class.

This run secured the 12th rank in Task 1 Arabic which used the dataset containing tweets in Arabic as shown in table. The model performed on the test set with a macro F1 score of 0.67.

Specifically, AraBERT is tailored to Arabic text and excels in sentiment analysis on Arabic tweets due to fine-tuning on a large corpus of Arabic text. The success with this model lies in the intricate morphology and syntax of the Arabic language, as well as sentiment expressions that are peculiar in the Arabic language tweets. This special training will ensure that subtlety and context-specific nuances, very important to be picked up for sentiment classification in Arabic, are so captured.

6. Conclusion

Although this solution promises to automate the checking of claims in several languages, there are a few areas where it can be improved. Firstly, more research can be done to combine the advantages of different models may result in greater F1 scores. Secondly, adding bigger and more varied training datasets can improve the model's generalization, especially for languages with limited resources. Moreover, cross-validation method and hyperparameter adjustments can be used to improve the overall performance.

Table 5
Rank list for English.

Team	F1
FactFinders	0.802
teamopenfact	0.796
innavogel	0.780
mjmanas54	0.778
ZHAWStudents	0.771
SemanticCUETSync	0.763
SINAI	0.761
DSHacker	0.760
visty	0.753
FiredfromNLP	0.745
TurQUaz	0.718
hybrinfox	0.711
SSNNLP	0.706
sz06571	0.696
NapierNLP	0.675
Mirela	0.658
KushalChandani	0.658
DataBees	0.619
TrioTitans	0.600
Madussree	0.583
pandas	0.579
JUNLP	0.541
mariuxi	0.517
grig95	0.497
CLaC2	0.494
AquaWave	0.339
Baseline	0.307

Finally, given that both language and misinformation are constantly changing, retraining of model and adaptation to new linguistic quirks and misinformation strategies can improve the system's ability to detect any misinformation

References

[1] M. Hasanain, R. Suwaileh, S. Weering, C. Li, T. Caselli, W. Zaghouani, A. Barrón-Cedeño, P. Nakov, F. Alam, Overview of the CLEF-2024 CheckThat! lab task 1 on check-worthiness estimation of multigenre content, in: G. Faggioli, N. Ferro, P. Galuščáková, A. García Seco de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CLEF 2024, Grenoble, France, 2024.

[2] A. Barrón-Cedeño, F. Alam, J. M. Struß, P. Nakov, T. Chakraborty, T. Elsayed, P. Przybyła, T. Caselli, G. Da San Martino, F. Haouari, C. Li, J. Piskorski, F. Ruggeri, X. Song, R. Suwaileh, Overview of the CLEF-2024 CheckThat! Lab: Check-worthiness, subjectivity, persuasion, roles, authorities and adversarial robustness, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. Di Nunzio, P. Galuščáková, A. García Seco de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), 2024.

[3] C. Hansen, C. Hansen, J. G. Simonsen, C. Lioma, The copenhagen team participation in the check-worthiness task of the competition of automatic identification and verification of claims in political debates of the clef-2018 checkthat! lab, in: Conference and Labs of the Evaluation Forum, 2018. URL: https://api.semanticscholar.org/CorpusID:215822646.

[4] N. Hassan, A. Nayak, V. Sable, C. Li, M. Tremayne, G. Zhang, F. Arslan, J. Caraballo, D. Jimenez, S. Gawsane, S. Hasan, M. Joseph, A. Kulkarni, Claimbuster: the first-ever end-to-end fact-checking system, Proceedings of the VLDB Endowment 10 (2017) 1945–1948. doi:10.14778/3137765.3137815.

[5] P. Gencheva, P. Nakov, L. Màrquez, A. Barrón-Cedeño, I. Koychev, A context-aware approach for detecting worth-checking claims in political debates, in: R. Mitkov, G. Angelova (Eds.), Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, INCOMA Ltd., Varna, Bulgaria, 2017, pp. 267–276. URL: https://doi.org/10.26615/978-954-452-049-6_037. doi:10.26615/978-954-452-049-6_037.

[6] A. Patwari, D. Goldwasser, S. Bagchi, Tathya: A multi-classifier system for detecting check-worthy statements in political debates, 2017, pp. 2259–2262. doi:10.1145/3132847.3133150.

[7] S. Vasileva, P. Atanasova, L. Màrquez, A. Barrón-Cedeño, P. Nakov, It takes nine to smell a rat: Neural multi-task learning for check-worthiness prediction, 2019.

[8] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, J. M. Struß, T. Mandl, R. Míguez, T. Caselli, M. Kutlu, W. Zaghouani, C. Li, S. Shaar, G. K. Shahi, H. Mubarak, A. Nikolov, N. Babulkov, Y. S. Kartal, J. Beltrán, The clef-2022 checkthat! lab on fighting the covid-19 infodemic and fake news detection, in: M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg, V. Setty (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham, 2022, pp. 416–428.

[9] A. Savchev, Ai rational at checkthat!-2022: Using transformer models for tweet classification., in: CLEF (Working Notes), 2022, pp. 656–659.

[10] S. Agresti, S. A. Hashemian, M. J. Carman, Polimi-flatearthers at checkthat!-2022: Gpt-3 applied to claim detection, in: Conference and Labs of the Evaluation Forum, 2022. URL: https://api.semanticscholar.org/CorpusID:251471103.

[11] T. Su, C. Macdonald, I. Ounis, Entity detection for check-worthiness prediction: Glasgow terrier at clef checkthat! 2019 (2019).

[12] P. Tarannum, F. Alam, M. A. Hasan, S. Noori, Z-index at checkthat! lab 2022: Check-worthiness identification on tweet text, 2022. doi:10.48550/arXiv.2207.07308.