A Multi-Aspect Classification Ensemble Approach for Profiling Fake News Spreaders on Twitter

Notebook for PAN at CLEF 2021

Clemens Hörtenhuemer and Eva Zangerle

Department of Computer Science
University of Innsbruck, Austria
clemens.hoertenhuemer@student.uibk.ac.at
eva.zangerle@uibk.ac.at

Abstract In this work, we attempt to differentiate authors of fake news and real news as part of the Profiling Fake News Spreaders on Twitter task at PAN. We propose a set of eight different language features to represent tweets. These representations are subsequently used in an ensemble classification model to identify fake news spreaders on Twitter. The approach is confined to the English language.

Keywords: Stacking ensemble, natural language processing, ensemble pruning, fake news detection

1 Introduction

Threats like public deceit or deep fakes, i.e., the artificially created, realistic imitation of an individual reasonably concern politicians, journalists, and sociologists [14]. In online social networks, fake messages and rumors are usually spread with the intention of deceiving users and manifesting certain opinions. Fake news is not new, but social media platforms have enabled the phenomenon to grow exponentially in recent years [17].

Therefore, technologies to detect intentionally spread fake messages are sought after. At the CLEF 2020 conference, the Profiling Fake News Spreaders on Twitter task at PAN addresses this matter [17]. The objective of the task is to study whether it is possible to distinguish authors who have disseminated fake news from those who, in their good faith, have never done so. For this task, a collection of sample messages from known fake news spreaders and truth-tellers was gathered from the Twitter microblogging platform and provided to participants. By using the same data and publishing the different approaches, the various teams can mutually inspire each other. Consequently, this should foster mutual improvements to teams' models and jointly advance approaches for detecting fake news spreaders.

According to [15], there are three different approaches to automatically determine the credibility of a certain post, tweet, or article:

– Truth finding refers to the extraction of structured claims from a certain post, tweet, or article and the comparison of those claims with trustworthy sources.
– Analysis of community behavior in social media aims to determine the credibility of a text based on probabilistic graph models and social media analysis.
– Natural language claims try to determine the credibility of a text by recognizing characterizing patterns in the writing style of fake news spreaders.

The goal of this work is to contribute to the systematic detection of fake news in social media networks. By applying the concepts of natural language claims, the approach offers an executable decision model for computing the probability that the author is a fake news spreader. The choice of which text properties are used to determine whether a message is fake or not plays a crucial role. Hence, a central part of this work is investigating which text features are suitable as indicators for fake news. In this context, Ghanem et al. [5] showed that the decomposition of a tweet, article, or post into manifold emotional features can help to detect fake news. Similarly, [23] found that positively associated words are relevant to identify sarcasm and negative words to identify irony. Hence, besides conventional text features such as TF-IDF or POS-tags, we also incorporate mood-related aspects for the detection of fake news. Given a set of eight text features, we propose to utilize an ensemble classification approach for the task of differentiating fake news spreaders and truth-tellers. The models of this approach are constrained to the English language.

The remainder of this paper is structured as follows. In Chapter 2, we describe the used dataset, our approach for feature extraction, and the employed classification model. In Chapter 3, we present the results obtained by applying the developed classification model to the dataset and we conclude our work in Chapter 4.

2 Method

In this section, we present the proposed features as well as the supervised learning model employed to detect fake news spreaders. The objective of the machine learning model is to assign tweets either to the class of tweets written by fake news spreaders or to the class of tweets written by truth-tellers. Based on this classification of tweets, we assign authors of these tweets either to the class of fake news spreaders or the class of truth-tellers.

In the following, we firstly introduce the dataset underlying our experiments, before we describe the employed data preprocessing and the features used to characterize tweets, before we elaborate on the ensemble classification approach utilized.

2.1 Dataset

The dataset used was provided by the PAN task committee of the “Profiling Fake News Spreaders on Twitter” task [17]. This dataset contains tweets of 300 Twitter users, whereby users are labeled as either fake news spreaders or truth-tellers. For each user, a rich collection of tweets was provided. Table 1 depicts an overview of the dataset.

	Fake News Spreaders	Truth-tellers	Total
Number of authors	150	150	300
Number of tweets per author	100	100	100
Total number of tweets	15,000	15,000	30,000
Average word length in characters	5.51	5.40	5.4589
Average tweet length in words	14.20	14.50	14.3479

Table 1. Overview of the dataset.

2.2 Preprocessing

Each tweet belongs to either an author who belongs to the class of fake news spreaders or the class of truth-tellers. In a first step, we group tweets of fake news spreaders and tweets of truth-tellers as we aim to generalize patterns that allow to distinguish these two classes. Each tweet is then labeled with the respective class.

Combining multiple tweets into a combined message taking into account the labels provides a more comprehensive information base for pattern recognition. In preliminary experiments, we observed that this concatenation has a positive effect on the accuracy of the classification system. Therefore, for further processing, groups of four tweets of the same author and annotated with the same label are joined together into one message, which is then used as input for all further steps.

2.3 Text Features

Based on the input tweets (or rather, the concatenation of four tweets), we aim to extract meaningful features for the classification of tweets and hence, authors. Our choice of appropriate text features was motivated by multiple prior works regarding both general text classification, as well as specifically existing work regarding the detection of fake messages.

The Bag of Words model (BOW) serves as a first initial basis for the representation of tweets [24]. Furthermore, we add the features proposed by the winner of the 2018 PAN-Task for Style Change Detection [26]: N-Grams, Term Frequency–Inverse Frequency, POS-Tags, Readability using Textstat and Named Entities (NER) using SpaCy[5]. Ghanem et al. [5] showed that incorporating emotions can be crucial for the recognition of fake news. Therefore, we also leverage the NRC emotional dictionary [13] to incorporate emotional features in our approach. Furthermore, we used Vader (Valence Aware Dictionary and sEntiment Reasoner) [8] to reflect the mood of a text (positive or negative). The average word length of each tweet was also added as a further feature. Moreover, we also utilized sentence embedding vectors for for each tweet to incorporate semantic properties of the tweets. Figure 1 illustrates the features extracted from a tweet in multiple strands. We detail the individual features in the following.

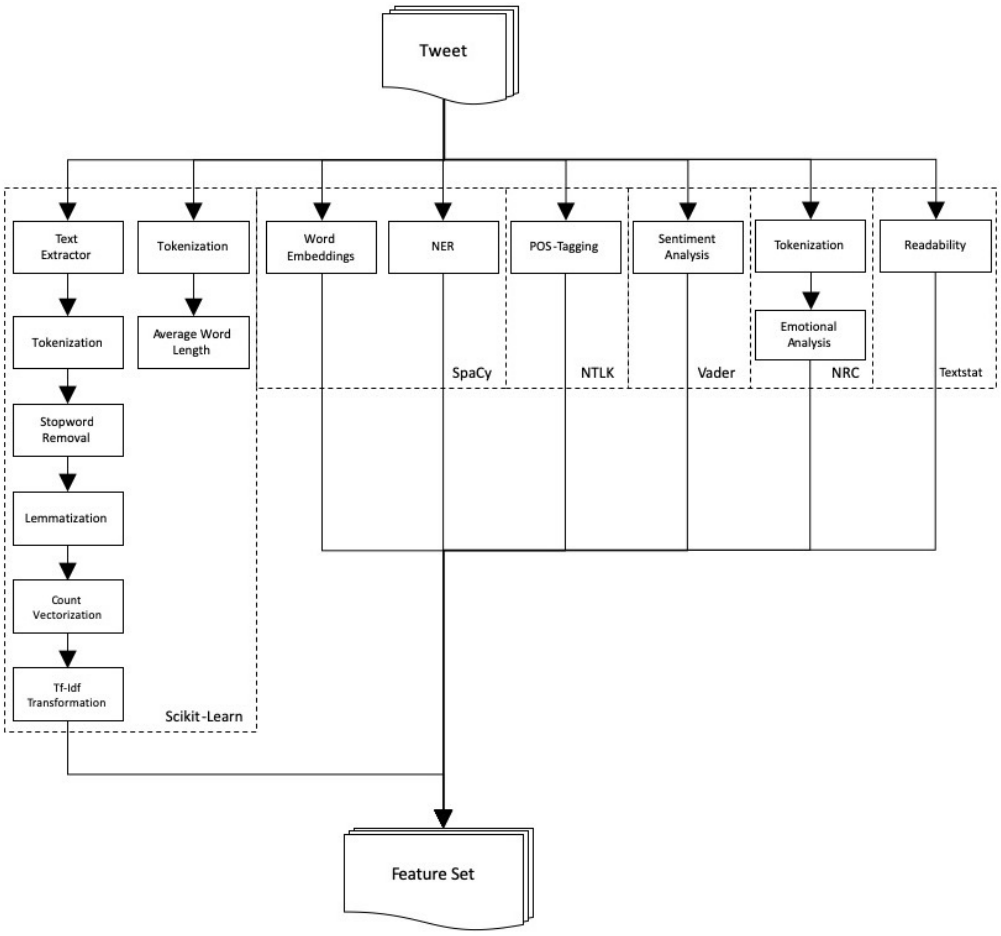

Figure 1. Feature Extraction Pipeline.

Term Frequency-Inverse Document Frequency (TF-IDF): In this strand, we extract basic text features: TF-IDF features and trigrams. Therefore, the texts are separated into tokens (words) using spaces and punctuation marks. We then remove stop words and transform words to their word stem. Based on this preprocessing, we extract word trigrams. We compute TF-IDF scores to reflect their relevance in relation to the entire text corpus for the individual word or trigram, respectively.

Average Word Length (AWL): In this strand, the average word length of a text is determined. The texts are separated into words using spaces. To determine the average word length of a text, the total number of characters in the text excluding spaces is divided by the total number of words in the text (also including stop words).

Word/Sentence Embeddings (WE): Here, we compute a sentence embedding for each text. The resulting numeric vectors allow to semantically compare texts. The NLP-library SpaCy[1] is used for the conversion into sentence vectors.

POS-Tags (POS): Part of Speech Tagging (POS-Tagging) is the classification of words into their part of speech. The words get classified with one of the following word types: Pronouns, prepositions, coordinating conjunctions, adjectives, adverbs, determinants, interjections, modals, nouns, personal pronouns, or verbs. For our approach, the number of occurrences of the different word types per text is added to the tweet representation. The NLP-libraries SpaCy[5] and NLTK[2] were tested for POS-tagging. Our preliminary experiments showed that NLTK contributes better to the accuracy of the overall system and is therefore used for this approach.

Named Entity Recognition (NER): Here, each proper name in the text is assigned to a specific category, such as person, company name or currency. We add the number of occurrences of each category as features to the tweet representation. The NLP-library SpaCy[5] is used for the extraction of the named entities.

Sentiment Analysis (SA): Using sentiment analysis, we aim to determine the sentiment of the text, whereby sentiment is measured by three dimensions:

- Positive (between 0 and 1)
- Negative (between 0 and 1)
- Neutral (between 0 and 1)

The positive, negative, and neutral scores represent the proportion of the text that falls into these three sentiment categories. Therefore, all these scores together should add up to 1. Additionally, there is the variable compound which expresses the three values in one dimension. We use the scores of the three dimensions and the compound value as a feature to describe the text. The sentiment analysis library Vader[3], which combines a sentiment-lexicon-approach and rule-based context consideration [8], is used for the extraction of the sentiments.

Emotional Analysis (EA): While sentiment analysis resolves the mood rather objectively between a positive or negative score, emotional analysis attempts to assess the text in terms of a multifaceted human perception of feelings. Since Ghanem et al. [5] state how important the consideration of human emotions is for the recognition of fake messages, this strand attempts to extract emotions from the given text. To achieve this, an analysis at token level is performed to check a text for the involvement of ten different types of emotions and their degree of expression.

We use the NRC emotion dictionary [13] to determine emotions. A word in the dictionary may have markers for the emotion types anger, anticipation, disgust, fear,

[1] https://spacy.io/
[2] https://www.nltk.org/
[3] https://pypi.org/project/vader-sentiment/

joy, negative, positive, sadness, surprise, and trust. A word can also have more than one marker if it is associated with more than one emotion. Each word in the text is looked up in the emotion dictionary, matching emotional markers are grouped within their type and counted across the entire text. The count of each type is normalized by dividing it by the total number of words with any emotional marker in the text. The normalized values for each emotion type is used as emotional features.

Readability (READ): How easy it is to read a text can also be a crucial feature describing a text. Zlatkova et al. [26] have promoted the consideration of readability in their work on style change detection. There are various static analysis methods for this purpose, for example, the Flesch Reading Ease-Test (FRE) [18], which calculates a score from the total number of sentences, words, and syllables. The score indicates how easy it is for the reader to understand the text. It is calculated as follows:

$$\text{FRE} = 206.835 - 1.015\Big(\frac{total\ words}{total\ sentences}\Big) - 84.6\Big(\frac{total\ syllables}{total\ words}\Big) \quad (1)$$

Along the lines of Zlatkova et al. [26], we also incorporate the following readability scores:

- Smog Grade[12]
- Flesch Kincaid Grade [18]
- Coleman Liau Index [22]
- Automated Readability Index [10]
- Dale Chall Readability Score [1]
- Difficult Words [7]
- Linsear Write Formula [3]
- Gunning Fog Index [6]

Each score is considered separately as a feature for the text. The library Textstat[4] was used for the calculation of the scores.

2.4 Classification

Based on the set of extracted features, the classifier aims to predict authors as fake or as real. When applied to a collection of tweets of an author, the probability of the author being a fake news spreader can be estimated.

In our approach, we evaluated multiple classification algorithms with the eight different feature types proposed in the previous section to obtain suitable combinations of classification algorithms and representations. In particular, we evaluated Support Vector Machines (SVM) [20], Random Forests (RF) [11], Artificial Neural Networks (ANN) [9], Adaptive Boosting (AdaBoost) [19], and Extreme Gradient Boosting (XGBoost) [2] approaches. For each of these classification approaches, we performed cross-validation with five folds [4] on the provided training data and with hyper parameters set according to Table 2.

Classifier	Hyper Parameter	Value
Support Vector Machine	penalty C	1
	kernel	linear
	maximum iterations	20000
Random Forest	maximum depth	none
	number of estimators	300
	random state	4
Artificial Neural Network	hidden layers	1
	hidden layer size	100
	output layer size	1
	alpha	0.001
	tolerance	0.001
	activation	Hyperbolic Tan
	maximum iterations	10000
	learning rate	0.001
	fitting algorithm	Backpropagation
	optimization	Adam
	regularization	L2
XGBoost	booster	Gbtree
	number of estimators	300
	maximum depth	8
AdaBoost	base estimator	Decision Tree
	base estimator maximum depth	7
	number of estimators	300

Table 2. Classifier Hyper-Parameters.

Classifier	TF-IDF	AWL	WE	POS	NER	SA	EA	READ	Average
Support Vector Machine	84.360	**53.920**	66.333	63.520	58.853	57.533	56.68	60.240	62.679
Random Forest	79.520	52.279	72.480	**70.330**	57.973	55.106	55.040	**60.907**	62.954
Artificial Neural Network	**84.800**	53.373	72.187	64.547	**60.080**	**57.653**	**57.440**	60.267	63.793
XGBoost	73.600	52.134	**73.747**	69.667	56.720	54.013	53.787	58.667	61.544
AdaBoost	73.000	51.920	70.813	67.760	55.400	52.960	53.653	58.080	60.448
Average	79.056	52.725	71.112	67.164	57.805	55.453	55.320	59.632	

Table 3. Accuracy of individual features and classifiers, computed on the training dataset.

Table 3 shows the accuracy values for each feature in combination with each proposed classification algorithm for the provided training data set. The best accuracy scores for each representation are highlighted in bold.

Given that we found that different features work differently well when combined with different classification algorithms, we propose to use an ensemble of classifiers for our approach. The primary assumption of ensemble methods is that if weak models are combined appropriately, more accurate and robust models can be achieved [25]. More precisely, we have chosen a stacking ensemble approach that deliberately combines various weak models of different types. Accuracy values were determined for all

[4] https://pypi.org/project/textstat/

combinations of algorithms and representations (see Table 3). However, only the best combinations of each representation and a classification method (marked in bold in the table) were used in the ensemble. This reduction to the essence is known as ensemble pruning [21]. It is a method to increase efficiency and prediction performance by reducing the ensemble of model components. The results of the eight classifiers are aggregated using logistic regression as meta classifier. Thereby the classifiers are weighted according to the accuracy scores they achieved on the training dataset (cf Table 3). The hyper parameters are set as specified in Table 2 and Table 4.

Parameter	**Value**
Classifier	Logistic Regression
Penalty C	1
Optimization	lbfgs
Maximum iterations	50,000
Regularization	L2
Tolerance	0.0001

Table 4. Meta Classifier Hyper Parameters

Figure 2 shows the model architecture of the pruned stacking classifier. Once the

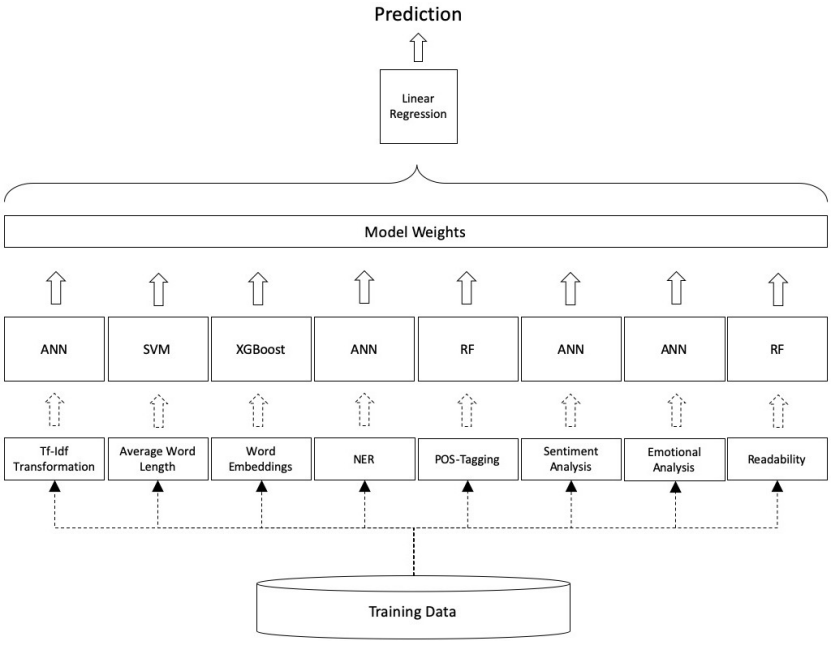

Figure 2. Pruned Stacking Classifier.

tweets of an author have been classified, the probability of being fake news spreader is used to classify the author itself. It is equal to the ratio of tweets classified as fake news to tweets classified as not fake news of an author, as represented by Equation 2. T_a denotes the set of tweets of an author a, F is the class of tweets containing fake news, and A_f is the class of fake news spreaders (i.e., authors of fake news).

$$P(a \in A_f) = \frac{1}{T} \sum_{t \in T_a} |t \in F| \tag{2}$$

An author is considered to be a fake news spreader if the calculated probability is above 0.5. If the probability is lower, the author is assigned to the class of truth-tellers.

3 Results and Discussion

In Table 3 we depict the accuracy scores of the individual representations. Here, TF-IDF stands out as the superior representation with an average accuracy of 79.056%, which was determined by applying a variety of classification algorithms. However, utilizing the proposed ensemble approach, we were able to increase the accuracy score by 6.144%.

The pruned stacking classifier, which utilizes the best performing classifiers of each representation, was evaluated by a seven-fold cross-validation of all labeled tweets of the training data. Thereby the following result was obtained:

Accuracy: 85.2002
Precision: 85.2329

The model was used for the classification of the test set of the according PAN task [17] in the TIRA [16] evaluation platform. Thereby a classification of authors was conducted based on authors' tweets according to Equation 2 and a score of 0.72 was obtained.

4 Conclusion

In this work, we aimed to identify suitable text features for the detection of fake news. An increase in accuracy was not achieved by unification at the representation level, but by combining multiple classification results based on the different representations independently of each other using different classification algorithms. Based on these findings, a pruned stacking classifier was developed which incorporates Support Vector Machines, Random Forests, Artificial Neural Networks, and Extreme Gradient Boosting Machines and considers eight different text representations.

References

1. Chall, J.S., Dale, E.: Readability revisited: The new Dale-Chall readability formula. Brookline Books (1995)

2. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794 (2016)
3. Eltorai, A.E., Naqvi, S.S., Ghanian, S., Eberson, C.P., Weiss, A.P.C., Born, C.T., Daniels, A.H.: Readability of invasive procedure consent forms. Clinical and translational science 8(6), 830–833 (2015)
4. Fushiki, T.: Estimation of prediction error by using k-fold cross-validation. Statistics and Computing 21(2), 137–146 (2011)
5. Ghanem, B., Rosso, P., Rangel, F.: An emotional analysis of false information in social media and news articles. ACM Trans. Internet Technol. 20(2) (Apr 2020), https://doi.org/10.1145/3381750
6. Gunning, R.: The fog index after twenty years. Journal of Business Communication 6(2), 3–13 (1969)
7. Hüning, M., Hüning, M.: Textstat simple text analysis tool. Dutch Linguistics, Free University of Berlin, Berlin (2005)
8. Hutto, C.J., Gilbert, E.: Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Eighth International AAAI Conference on Weblogs and Social Media (2014)
9. Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial neural networks: A tutorial. Computer 29(3), 31–44 (1996)
10. Kincaid, J.P., Fishburne Jr, R.P., Rogers, R.L., Chissom, B.S.: Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Tech. rep., Naval Technical Training Command Millington TN Research Branch (1975)
11. Liaw, A., Wiener, M., et al.: Classification and regression by randomforest. R news 2(3), 18–22 (2002)
12. Mc Laughlin, G.H.: Smog grading-a new readability formula. Journal of reading 12(8), 639–646 (1969)
13. Mohammad, S.M., Turney, P.D.: NRC emotion lexicon. National Research Council, Canada (2013)
14. Nemitz, P.: Constitutional democracy and technology in the age of artificial intelligence. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 376(2133), 20180089 (2018)
15. Popat, K., Mukherjee, S., Yates, A., Weikum, G.: DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 22–32. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018), https://www.aclweb.org/anthology/D18-1003
16. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World. Springer (Sep 2019)
17. Rangel, F., Giachanou, A., Ghanem, B., Rosso, P.: Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2020)
18. Rudolf, F.: How to write plain english. University of Canterbury (2016)
19. Schapire, R.E.: Explaining adaboost. In: Empirical inference, pp. 37–52. Springer (2013)
20. Schölkopf, B., Smola, A.J., Bach, F., et al.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press (2002)
21. Tsoumakas, G., Partalas, I., Vlahavas, I.: An ensemble pruning primer. In: Applications of supervised and unsupervised ensemble methods, pp. 1–13. Springer (2009)

22. Vajjala, S., Meurers, D.: On improving the accuracy of readability classification using insights from second language acquisition. In: Proceedings of the seventh workshop on building educational applications using NLP. pp. 163–173. Association for Computational Linguistics (2012)
23. Wang, P.Y.A.: # irony or# sarcasm—a quantitative and qualitative study based on twitter. In: Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation (PACLIC 27). pp. 349–356 (2013)
24. Zhang, Y., Jin, R., Zhou, Z.H.: Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics 1(1-4), 43–52 (2010)
25. Zhou, Z.H.: Ensemble learning. Encyclopedia of biometrics 1, 270–273 (2009)
26. Zlatkova, D., Kopev, D., Mitov, K., Atanasov, A., Hardalov, M., Koychev, I., Nakov, P.: An ensemble-rich multi-aspect approach for robust style change detection. CLEF 2018 Working Nots of CLEF (2018)