A New Language Independent Strategy for Clickbait Detection

Claudia Ioana Coste
Department of Computer Science
Babeş-Bolyai University
Cluj-Napoca, Romania
c.i.coste@cs.ubbcluj.ro

Darius Bufnea
Department of Computer Science
Babeş-Bolyai University
Cluj-Napoca, Romania
bufny@cs.ubbcluj.ro

Virginia Niculescu
Department of Computer Science
Babeş-Bolyai University
Cluj-Napoca, Romania
vniculescu@cs.ubbcluj.ro

***Abstract*—Clickbait is a bad habit of today's web publishers, which resort to such a technique in order to deceive web visitors and increase publishers' page views and advertising revenue. Clickbait incidence is also an indicator for fake news and so, clickbait detection represents a mean in the fight against spreading false information. Recently, both the research community and the big actors on the WWW scene such as social networks and search engines, turn their attention towards this negative phenomenon that is more and more present in our everyday browsing experience. The detection techniques are usually based on intelligent classifiers, features selection being also of great importance. This paper aims to bring its own contributions in clickbait analysis and detection by presenting a new language independent strategy for clickbait detection that considers only general features that are non language specific. This approach is justified by the need for a higher level of abstractization in the clickbait detection, allowing its usability for articles written in different languages. A complex experiment on a real sample dataset was conducted and the obtained results are compared with the most relevant previous work results.**

***Index Terms*—clickbait detection, features, intelligent classifier, natural language, accuracy**

I. Introduction

In the last two decades, some classical content providers went through a major change, part of the old media industry moving to online. In order to survive, a lot of newspapers made this step, giving up to their printed edition and publishing all their content in online editions or through different social media/networks where it can be easily accessed and shared by everyone. In this new business model, most news websites do not charge a subscription for their provided content, relying only on advertisements displayed on their webpage or on some other monetization techniques such as affiliate links to secure their income. In order to increase their page views and advertising revenue, the online content providers often use some deceptive methods so that the online content consumer (i.e. the web visitor) stays trapped as much as possible, clickbait being one of these content providers' bad habits. Both the research community and the news consumer are becoming lately more and more aware of the serious implications that low quality articles and thin content have in the online world.

The term clickbait is used for identifying links to articles having confusing, misleading, or meant to shock titles that exaggerate the content on the landing page. The term itself was born in the 20th century, when the TV audience was advised not to change the channel during the commercial breaks. More recently, according to the Oxford Advanced Learner's Dictionary, this word refers to "material put on the Internet in order to attract attention and encourage visitors to click on a link to a particular web page". Although, clickbait is in contradiction with journalism ethics, web editors use it abusively in order to increase their page views and their income as well. Following such a bait link usually has a negative impact on user experience. Mainly due to some confusing, incomplete or exaggerated links, most web consumers are "trapped" in the publishers' bait, leaving them disappointed or frustrated because the clicked article does not meet their expectations (as induced by the title). Moreover, this time consuming and unhappy user experience can also propagate outside a publisher's network to third party websites. An incomplete or a catchy misleading title can be used by a social network in its news feed or as a text link by a search engine in SERP - Search Engine Results Page [1]. In such situations, the source of the bait can be considered the social network or the search engine, even though they are not directly involved nor they are affiliated in any way with the bait's creator (i.e. the content publisher). Through "bait" messages, other low quality articles are easily disseminated and shared through online ecosystems. Such articles include, but are not limited to: fake news, gossip or unfounded rumours. In [3] is stated that clickbait is a strong indicator for fake news and by using it, false information is spread widely. Clickbait is considered to be the commercial or teaser while the fake news is considered to be its content. Furthermore, psychology studies presented in [2], revealed that a correction in perception is highly unlikely to happen in the human brain once the false one has been formed originally. Baruch Spinoza's principles states that only on the second step the human brain starts questioning if the received information is valid, the first basic step is registering as true any information received from the senses. It often happens that due to external or internal factors, such as: tiredness, stress, noise, the source's credibility etc., this second step is missed. The online content providers rely on this vulnerability and exploit it. Considering all the above arguments, it is of high importance to take action against these publishers' bad habits.

In order to provide a more pleasant and less deceiving online browsing experience, the current paper aims to bring its own contributions in clickbait detection. The paper presents an overview of the most important clickbait detection techniques developed so far by the research community in this field and introduces a new language independent strategy for clickbait detection. The results obtained by experimenting with the proposed strategy are compared with the most relevant previous work results. The rest of this paper is organized as follows. The next section presents the main results of other related work in the field. Section III describes our new proposed strategy, and the subsequent results obtained through the experiments; the comparative analysis of the results being done in section IV. Section V concludes our paper, summarizing our results and presenting possible future work.

II. Related Work

Lately, the academic research community but also big actors on the WWW scene such as social networks and search engines, turned their attention towards this negative phenomenon that is more and more present in online media. Some extensive studies have been performed so far in order to identify clickbait links or articles, the vast majority of them using machine learning based approaches.

Being a relative new topic of study, the clickbait detection algorithms developed so far do not have a reliable accuracy (over 95%). Moreover, the datasets used for testing and training these algorithms contain only English entries. Because clickbait has begun to grow into a global phenomenon, the detection algorithms should be trained on multiple languages.

Authors of [4] classify clickbaits into eight categories: details omitting, exaggeration, usage of vulgar words, reporting of vulgar and unbelievable stories, bait and switch (users have to click once more to get the rest of the content), confusing titles, articles containing erroneous information, and punctuation formatting (excessive use of exclamation marks, question marks and uppercases). They developed a classification algorithm based on decision trees with 74.9% accuracy. The classification was performed on the following characteristics: title, informality, similarity between title and content, URL and use of references (demonstratives, third person pronouns, definite articles, starting the title with adverbs). The most relevant features turned out to be informality ($fmeasure$ [5], $Coleman - LiauScore$ [6], LIX and RIX indexes [7]) and the use of references features.

Paper [8] describes a dataset containing almost three thousand tweets manually annotated by volunteers as being clickbait or not. The considered tweets were published by well known online publishers such as: BBC, ABC News, New York Times, The Guardian, CNN, Fox News etc. Authors selected over two hundred features for the classification model that can be split into three categories: the "bait" message (textual statistics, dictionary and language patterns), the web link and the meta data of the website. The best accuracy of around 79% was obtained for the RF (Random Forest) Classifier [12].

The authors of [9] organized the Clickbait Challenge event in order to find new ways to fight clickbait, a global mass disinformation phenomena, but also to draw attention about its serious political and economic implications. They constructed a database by using the Twitter API formed from clickbait and non clickbait samples annotated by five different volunteers in a regression manner, such as: 0 means non clickbait, 0.33 - slightly clickbait, 0.66 - quite clickbait and 1 - clickbait. The database is publicly available for everyone, and researchers can still submit their solutions for validation. The best algorithm is called "albacore", which uses a recursive bidirectional neural network of type biGRU [13] and has an accuracy of 85.5%.

Paper [11] developed a configurable browser plugin and a clickbait classifier that is using only the article's title. This plugin tries to automatically detect clickbaits, also offering an option to block them. The classifier was built using a natural language processor (Stanford NLPCore [14]), support vector machine and 10-fold validation, achieving an accuracy of 93%. The used dataset consists in over 18000 non clickbait Wikinews articles, while the clickbait samples were extracted from publications well known for writing low quality articles. Authors of [11] draw some important conclusions: high quality articles contain a higher ratio of proper nouns, while clickbait headlines use causal complements and adverbs, possessives and personal pronouns. Another conclusion was that verbs are used differently: while non clickbait articles use participles and third person singular, clickbait articles use past tense verbs at the first and second person.

Authors of [1] relied on a community driven clickbait database based on user reports. They developed a Chrome plugin that will assist users in reporting a link they consider clickbait. Continuing the work started in [1], paper [15] proposes completing the previously built community driven clickbait database with non clickbait samples from different sources such as Squid logs. These logs are analysed and then filtered in order to determinate user navigation patterns. The purpose of browser extension presented in [1] is not to block low quality articles, but to develop a reliable dataset with relevant information written in multiple languages that will be later used as input for machine learning algorithms. [16] gives a deep analysis of clickbait and fake news in the context generated by the 2016's major media events: Trump's election in the USA and Brexit UK referendum. Authors talk from an interdisciplinary point of view about fake news, clickbait, their characteristics, their financial and ideological implications.

III. Clickbait Detection Strategy

We have developed and tested a clickbait detection strategy based on an intelligent classifier that assures language independence considering a proper features selection. Such an approach has the advantage that it can be used to classify articles written in multiple languages. A similar goal has been followed in [10], where the authors used a Convolutional Neural Networks based approach in order to detect multilingual clickbait articles, using distributed word and character embeddings as features to their neural network models. The

main stages and decisions of our proposed strategy will be described together with a comparative analysis of the results obtained through the subsequent conducted experiments.

The strategy consists in: a set of decisions related to the classification model, the application of the intelligent classifier and the considered metrics of accuracy. It is driven by the decision to obtain a language independent clickbait detection but also by a deep analysis of the related work. When designing the classification model, we have taken into consideration two main steps: the features extraction process, using just natural language processors and not linguistic dictionaries or other language dependent resources, and the selection step, where characteristics are chosen based on their importance achieved in clickbait detection.

Considering only features that are non-dependent on a specific language it is possible to use a sample dataset that contains articles and links written in multiple languages, and even articles that are written in more than just one language.

Features' types and their metrics. Analyzing the clickbait literature, we have observed that most of the detection methods are extracting features from the article's title and the bait message, which form the teaser that urges users to click on it. There are also features extracted from the metainformation of the web page [8] or the web link [4].

The most relevant features that lead to a high accuracy in classification are: N-grams - meaning language specific phrases or frequent groups of words, clickbait words often used for teasing the reader. Moreover, characteristics like the title or the bait message are measured formally with metrics as $fmeasure$ [5], LIX and RIX indexes [7] and $CLScore$ [6]. The number of proper or common nouns, usage of uppercases, punctuation patterns, morphological or syntactic patterns are features that have been proven to be quite important as well.

Datasets. The dataset used in our experiments is taken from The Clickbait Challenge [23]. There are three available datasets, two of them labeled and one unlabeled. The samples were collected using Twitter API from several accounts that are well known for sharing news and articles. The annotation process took place through some questionnaires on the Amazon Mechanical Turk (AMT - https://www.mturk.com) and each tweet was labeled by five volunteers. As it is well argued in [9], clickbait is a subjective phenomenon, being strongly influenced by the cultural, social and economic background of the reader. Thus, the dataset was labeled with a 4-point scale: 0 (not clickbaiting), 0.33 (slightly clickbaiting), 0.66 (considerably clickbaiting), 1.0 (heavily clickbaiting). Moreover, each annotated result was reviewed and if there were found any irregularities, the label was discarded, resubmitting the tweet in the AMT platform. The used dataset contains two files (in JSONL format) and a pictures folder containing the images added in the tweets. The two JSONL files have the following structure presented in Table I.

Preprocessing. Before using the dataset in the classification application, we have processed the samples by verifying them for any infiltrated any errors. Moreover, in order to balance our database, we needed to even out the number of clickbait samples with the number of non clickbait samples as it can be seen in Table II. When using the data into the machine learning algorithm, it was randomly split into 80% training and 20% testing, each set containing equal number of samples for clickbait, respectively for non clickbait.

Extracting language independent features. The used characteristics are extracted only from the article's title ("targetTitle" attribute) and from the teaser message ("postText" attribute). The features are extracted using the natural language processor Stanford NLPCore [20]. This software offers the possibility to parse the text into sentences, words and label part of speech tags and syntactic tags. This tool was developed for many programming languages including Python and it has modules for 53 languages [18]. The natural language processor is based on a pretrained neuronal model. The modules for each language must be separately downloaded and then trained with the model. First, we need to detect the language and then use the corresponding language model.

When labeling the semantic and syntactic structures, we consider the universal POS tags as described in [19]. The characteristics from Tabel III were chosen as being relevant.

$Fmeasure$ [5], RIX, LIX indexes [7] and $CLScore$ [6] were computed as presented in Equation 1, Equation 2 and, respectively, Equation 3.

$$fmeasure = \frac{nounFreq + adjectiveFreq + prepositionFreq + articlesFreq}{2} + \frac{-(pronounsFreq + verbsFreq + adversFreq + injectionsFreq) + 100}{2} \tag{1}$$

$$RIX = \frac{LW}{S}; LIX = \frac{W}{S} + \frac{(100 * LW)}{W}, \tag{2}$$

where W is the total number of words, S is the number of sentences and LW is the number of long words (long words are considered words with more than 7 characters).

$$CLScore = 5.88 \cdot L - 29.6 \cdot S - 15.8, \tag{3}$$

where L is the average number of letters per 100 words and S is the average number of sentences per 100 words.

Normalisation. Most of the input data are float numbers, between 0 and 1. The data that represents the number of particular parts of speech tags have limited values, that don't differ much between the samples. The boolean values are mapped as 0.0 for false and 1.0 for true. The LIX index values are relabeled in order to have natural numbers between 0 and 4, representing a level of formality: very easy (0-24), easy (25-34), standard (35-44), difficult (45-54), very difficult (>55). In the same way, the RIX real values were also relabeled according to the corresponding interval based on the intervals generated by the following numbers: 0.2, 0.5, 0.8, 1.3, 1.8, 2.4, 3.0, 3.7, 4.5, 5.3, 6.2, 7.2 as in [4]. Thus, the values for RIX features are natural values between 0 and 13.

Intelligent Classifier. The related work analysis reflects that the most frequently applied intelligent classification algorithms are: Gradient Boosted Decision Trees [4], Random Forest, Logistic Regression, Naive Bayes [8]. From this analysis is

TABLE I
THE CLICKBAIT CHALLENGE DATABASE STRUCTURE [23]

File	Attributes	Data type	Observations
instances.jsonl contains the tweet samples and other information collected later prefixed with "target"	id	string	
	postTimestamp	datetime	the date on which the post was published
	postText	string array	the text post without any links, meaning the bait message
	postMedia	string array	relative path to the attached photos from the "media" folder
	targetTitle	string	the title of the shared article
	targetDescription	string	article's description tags
	targetKeywords	string	keywords separated by comma
	targetParagraphs	string array	all paragraphs of the web news
	targetCaptions	string array	all the descriptions of the article's attached pictures
truth.jsonl contains the annotations of the 5 volunteers, the mean and the output class	id	string	
	truthJudgments	float number array	the annotated scores labeled by the 5 volunteers
	truthMean	float number	the arithmetic mean of the five scores from "truthJudgments"
	truthMedian	float number	it represents the middle value of the 5-element array sorted ascending or descending
	truthMode	0.0 or 1.0	codified output classes
	truthClass	string	"clickbait" or "non clickbait", the name of the output class

TABLE II
DATASET DISTRIBUTION

Database	No. of clickbait samples	No. of non clickbait samples	No. of deleted samples (containing "noise")
The Clickbait Challenge	5523	16474	997
Total number of correct samples	4988	16012	-
Number of samples used for detection	4988	4988	-

TABLE III
CHARACTERISTICS

1	No. of words in the title
2	Average length of words in the title
3	Punctuation patterns found in title, such as: "!?", "...", "***", "!!!", "???", "(", ")", "$"
4	No. of common nouns found in title and the "bait" message
5	No. of proper nouns found in title and the "bait" message
6	No. of common nouns with the syntactical tag of subject found in title and the post message;
7	No. of proper nouns with the syntactical tag of subject found in title and the post message
8	No. and presence (boolean value) of question marks found in title and message
9	No. and presence (boolean value) of exclamation marks found in title and message
10	No. of uppercases in title and message;
11	*Fmeasure* for the title
12	*LIX* and *RIX* indexes for the title and the message
13	*CLScore* for the message;
14	Presence (boolean value) of demonstratives in the title
15	Presence (boolean value) of personal pronouns found in the title
16	Presence (boolean value) of possessives found in the title
17	If the title starts with an adverb (boolean value);
19	No. of acronyms from the title [1]
19	Avg. of the words' length per sentence computed for the message
20	No. of numerals in the title

deduced that RandomForestClassifier could be an appropriate choice ([4], [8], [24], [11], [25]). RandomForestClassifier is an ensemble learning formed from multiple decision trees. They have a low bias, a high variance and a voting system when choosing the output class through which the overfitting problem is solved [26]. Each tree is built based on just a set of features used in classification, the process being called feature bagging and tries to avoid a high correlation between the trees [12]. (We have used an already implemented variant written in Python, from "scikit-learn" package, version 0.21 [28]).

As concerning the parameters calibration of the RandomForestClassifier, we choose them as it can be seen below as a result of several tests that we have done:

```
RandomForestClassifier(bootstrap=True, class_weight='balanced',
   criterion='entropy', max_depth=None, max_features='auto',
   max_leaf_nodes=None, min_impurity_decrease=0.0,
   min_impurity_split=None, min_samples_leaf=1,
   min_samples_split=2, min_weight_fraction_leaf=0.0,
   n_estimators=150, n_jobs=-1, oob_score=True, random_state=0,
   verbose=0, warm_start=False)
```

Table IV contains all the parameters used for adjusting the RF Classifier, their values, a brief description and motivation for the value chosen. The final values were set according to the official documentation [28], [27] and our empirical observations on the test results.

IV. RESULTS AND COMPARATIVE ANALYSIS

Metrics. In order to evaluate the performance of the RF Classifier we consider the following metrics (the same metrics as those used in The Clickbait Challenge evaluation): accuracy, precision, recall, F1 score, mean squared error (MSE) [21] and normalized mean squared error (NMSE) [22]. The results obtained with our algorithm are presented in Table V. All results represent the average of 200 executions of the program.

Analysis. Comparing our results with The Clickbait Challenge results [23], we will be classified on the 1st rank, considering F1 score or precision, 3rd rank for recall, 14th rank for accuracy and 15th rank for mean squared error (MSE). But we should take into consideration that the results of The Clickbait Challenge's submissions were computed after submitting the application on TIRA platform, and the organizers keep secret their testing dataset. We compute our results on the free dataset, which had around 1995 samples and we know that the output classes are balanced (approximately 997 clickbaits and 997 non clickbaits).

Even if the dataset used for training and testing are different, we could compare our results with the one presented in [4], which scores a 74.9% accuracy (75% precision, 76% recall and 74.9% F1 score) using a Decision Tree. Their dataset was

TABLE IV
PARAMETERS CALIBRATION

Parameter	Value	Observations
bootstrap	True (default)	Will split the input samples between trees in the constructing process
class_weight	'balanced'	The results proportion are inversely proportional with the class frequency
criterion	'entropy'	The tree construction will be done using information gain
max_depth	None (default)	Allows trees to not have a maximum depth, meaning the leaves will have a single output class
max_features	'auto' (default)	Allows the algorithm to automatically adjust the maximum numbers of features used in a split
max_leaf_nodes	None (default)	Lower impurity rate when the tree growth process takes place
min_impurity_decrease	0.0 (default)	Represents the minimum degree for impurity decrease when making a split in the tree
min_impurity_split	None	There will not be a threshold value to stop the trees' growth
min_samples_leaf	1 (default)	Represents the dimension of the terminal nodes of the tree
min_samples_split	2 (default)	Implies having at least 2 samples to make a split in the tree
min_weight_fraction_leaf	0.0 (default)	The same signification as "min_samples_leaf", but represents a percent
n_estimators	150	The value was chosen based on several test results
n_jobs	-1	The number of CPU cores. Value -1 means that all available CPU cores will be used
oob_score	True	The algorithm will use cross validation during training
random_state	0	Implies using a random instance or a given one in the bootstrap process
verbose	0 (default)	Refers to the printed output given when running the classifier
warm_start	False (default)	Set to False means creating new estimators at each run and not adding estimators to a previous instance We set it False such that the tests will give us proper results, not influencing one another

TABLE V
RESULTS OBTAINED

Metric	Value
Accuracy	75.80886773547095%
Recall	71.99860725632206%
F1 score	74.86521566230624%
Precision	78.00013377034282%
MSE	0.2419113226452905
NMSE	1.046850961569075

manually annotated, containing articles from: The Post, The New York Times, CBS, Forbes, The Huffington etc., having in total: 1349 clickbaits and 2724 non clickbaits.

The best solution to our knowledge is presented in [11], which scores a 93% accuracy (precision 95%, recall 90% and F1 score 93% obtained for all features) with a Support Vector Machine and 10-fold cross validation. Their dataset contained Wikinews articles for the non clickbait samples and articles manually annotated by volunteers from journalistic publications such as: ViralStories, BuzzFeed, Upworthy, ViralNova and Scoopwhoop. But it is important to remark that this solution also uses language dependent features, and uses a specialized dataset.

Authors of [8] obtained an accuracy of around 79% (precision 70%, recall 73%, F1 score 74% - results obtained for all features) with a RF Classifier. Their database consists of 2992 tweets manually annotated by three volunteers from famous journalistic providers such as: BBC, New York Times, ABC News, CNN, Fox News and so on.

Table VI presents the importance of the top 17 features, as it was determined by Random Forest classifier. Initially, there had been over 41 features but the ones that had an importance less than 1% were dismissed from the classification model. In Table VI, it can be observed that the most relevant feature was the number of proper nouns found in the tweet postText. As it was well observed before in the clickbait literature [11], clickbait articles use a few proper nouns and they usually don't give many details about the event, the place or the persons involved. Unlike clickbait articles where the message transmitted to the public is vague and confusing, high quality

TABLE VI
FEATURES' IMPORTANCE

No.	Feature	Importance percent (%)
1	No. proper nouns (postText)	13.7442157
2	*CLScore (postText)*	12.057613
3	*Fmeasure (targetTitle)*	10.3834903
4	Average word length (targetTitle)	10.1007825
5	Average no. words per sentence (postText)	8.0996589
6	No. common nouns (postText)	6.6355859
7	No. proper nouns (targetTitle)	5.9261148
8	*RIX (postText)*	5.3657109
9	No. common nouns (targetTitle)	5.1646995
10	*RIX (targetTitle)*	4.4815785
11	No. uppercases words (postText)	3.9665707
12	No. acronyms (postText)	3.7826569
13	*LIX (postText)*	3.092507
14	*LIX (targetTitle)*	3.0179575
15	If starts with a numeral (targetTitle)	1.5171796
16	Existence of personal pronouns (targetTitle)	1.481859
17	If starts with an adverb (targetTitle)	1.1818192

news use plenty of proper nouns in order to present accurately the events, giving all the details they know.

The second and the third features refer to article informality measures $CLScore$ and $Fmeasure$. Both achieved high importance as it was also the case of experiments done in [4], where together with RIX, were in the top 10 features. It is known that clickbait articles put visible effort in attracting a click, so they want to use informal language, slang or even vulgar words to draw attention and to increase readers curiosity. In contrast, non clickbait articles use a more formal rigorous language, maybe even specialized in order to objectively report the series of events.

The characteristic referring to the average word length is quite interesting considering the information disseminated through the literature. [11] states that in clickbait articles the average word length for the title is around 7, unlike non clickbait news where it is 10. This is due to the fact that bait articles make excessive use of acronyms, abbreviations, slang, which contain simplified words and phrases.

Also, we observed that the first two characteristics are extracted from the post text, the bait message which engages

the user to click on it. The next two are taken from the article title, which could be mentioned or not in the bait post. The table was split into three sections: one containing the features that have an importance of over 10%, the next one with values between 5% and 10% and the last one with less than 5%. What is worth mentioning is that the first features are general features which makes a clear difference between what is clickbait and what is not. Properties that obtained a relevance of less than 5% need to be also taken into consideration because they resolve the overfitting problem. Moreover, on the used dataset these could be quite rare, but on another dataset, they could be relevant in clickbait detection process.

V. Conclusions and Future Work

We proposed a strategy to test the viability of a clickbait detection algorithm that rely only on features that are language independent. The results are satisfactory enough, we achieved 75% accuracy, 78% precision, 71% recall and 74% F1 score using a RF Classifier. Moreover, the values computed for feature importance confirm previous results obtained in clickbait detection literature. The most important characteristic of our strategy is that it uses only non-language dependent features, and so allows training of the classification algorithm on a sample dataset with articles written in different languages. The feature extraction process took into consideration just universal characteristics, based on universal part of speech tags annotated by a natural language processor. In this context, the obtained accuracy and precision could be evaluated as being very good. The proposed technique could be considered as a general instrument in fighting clickbait. As future work, we intend to test the developed strategy to other several intelligent algorithms such as K-Nearest Neighbor, Decision Tree, Naive Bayes, Neural Networks in order to maximize the accuracy. Also, we plan to conduct our experiments on an extended dataset with clickbait samples collected using the plugin presented in [1] and non clickbait samples collected using the plugin presented in [17], [29] and [30]. Finally, we intend to implement a browser plugin that will signal clickbaits found in media content, helping users to have a better, undeceived online experience.

Acknowledgment

Ms. Claudia Ioana Coste was supported in this research by a Special Scholarships for Scientific Activity granted by the Babeş-Bolyai University via its STAR-UBB Institute.

References

[1] D. Bufnea and D. Sotropa, “A community driven approach for click bait reporting,” in Proceedings of the 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1-6, September 13-15, 2018, Split - Supetar (Island of Brac), Croatia.

[2] D.T. Gilbert et al., “Unbelieving the unbelievable: some problems in the rejection of false information,” in Journal of Personality and Social Psychology 59 (4), 601-613, 1990.

[3] K. Shu et al., “Fake news detection on social media: a data mining perspective,” in: ACM SIGKDD Explorations Newsletter, pp. 22-36, 2017.

[4] P. Biyani, K. Tsioutsiouliklis and J. Blackmer, “8 Amazing Secrets for Getting More Clicks: Detecting Clickbaits in News Streams Using Article Informality,” in Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016. pp. 94-100, Phoenix, Arizona, SUA.

[5] F. Heylighen and J.-M. Dewaele, “Formality of language: definition, measurement and behavioral determinants,” 1999.

[6] M. Coleman and T.L. Liau, “A computer readability formula designed for machine scoring, in Journal of Applied Psychology,” 1975, Vol. 60. pp. 283-284.

[7] J. Anderson, “Lix and Rix: Variations on a Little-known Readability Index,” in Journal of Reading 26 (6), pp. 490-496, 1983.

[8] M. Potthast et al., “Clickbait detection”, in: Advances in Information Retrieval. ECIR 2016. Lecture Notes in Computer Science, vol 9626. Springer, Cham. DOI: 10.1007/978-3-319-30671-1_72, 2016.

[9] M. Potthast et al., “The clickbait challenge 2017: towards a regression model for clickbait strength,” s.l. : CoRR, 2018, Vol. abs/1812.10847.

[10] A. Anand, T. Chakraborty and N. Park, “We used neural networks to detect clickbaits: you won’t believe what happened next!,” in Advances in Information Retrieval, 39th European Conference on IR Research (ECIR’17), Lecture Notes in Computer Science, Springer, 2017.

[11] A. Chakraborty et al., “Stop clickbait: detecting and preventing clickbaits in online news media,” San Francisco, CA, SUA: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2016. 978-1-5090-2846-7.

[12] D. Denisko and M.M. Hoffman, “Classification and interaction in random forests,” Proceedings of the National Academy of Sciences Feb 2018, 115 (8) 1690-1692; DOI: 10.1073/pnas.1800256115.

[13] A. Zhang et al., “Dive into deep learning”, Chapter “Bidirectional recurrent neural networks,” URL: https://d2l.ai/d2l-en.pdf (visited on: 10/06/2020).

[14] C. Manning et al., “The Stanford CoreNLP natural language processing toolkit,” in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Association for Computational Linguistics. Baltimore: Maryland, pp. 55-60, 2014.

[15] C.I. Coste, “Controlling the click bait,” Arad, Romania: Proceedings of the International Student Conference StudMath-IT, 2018. pp. 11–17.

[16] C.I. Coste, “Online bad habits: fake news and clickbait,” in Proceedings of the Student Interdisciplinary Conference The European Union and Global Order, April 5th, 2019, Cluj-Napoca, Romania.

[17] I. Badarinza, A. Sterca and D. Bufnea, “A dataset for evaluating query suggestion algorithms in information retrieval,” in Proceedings of the 27th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2019.

[18] “StanfordNLP - Python NLP Library for Many Human Languages,”, 2017, URL: https://stanfordnlp.github.io/stanfordnlp/ (visited on 05/12/2019).

[19] “Universal Dependencies,” 2014, Universal POS tags. URL: https://universaldependencies.org/u/pos/ (visited on 05/12/2019)

[20] P. Qi et al., “Universal Dependency Parsing from Scratch,” in Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, 2018. pp. 160-170.

[21] A. Mishra, “Metrics to Evaluate your Machine Learning Algorithm,” 2018, URL: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234, (visited on: 26/03/2020).

[22] NMSE, URL: https://rem.jrc.ec.europa.eu/RemWeb/atmes2/20b.htm (visited on: 14/05/2020).

[23] Bauhaus-Universität Weimar, The Clickbait Challenge, 2017, URL: https://www.clickbait-challenge.org/ (visited on 05/05/2020).

[24] X. Cao, et al., Machine Learning Based Detection of Clickbait Posts in Social Media, 2018, CoRR abs/1710.01977.

[25] A. Grigorev, Identifying Clickbait Posts on Social Media with Ensemble of Linear Models, 2017, CoRR abs/1710.00399.

[26] G. Louppe, Understanding Random Forests, from Theory to Practice, PhD dissertation, 2014, Cornell University.

[27] T. Plapinger, Tuning a Random Forest Classifier, 2017, URL: https://medium.com/@taplapinger/tuning-a-random-forest-classifier-1b252d1dde92 (visited on: 15/04/2020).

[28] scikit-learn Machine Learning in Python, 2017, URL: https://scikit-learn.org/stable/ (visited on 10/01/2020).

[29] I. Badarinza, A. Sterca, F. Boian, “Using the user’s recent browsing history for personalized query suggestions,” IEEE SoftCom 2018, Split, Croatia.

[30] I. Badarinza, A. Sterca, F. Boian, “The Role of the User’s Browsing and Query History for Improving MPC-generated Query Suggestions,” Journal of Communications Software and Systems, vol. 15, no. 1, pp. 26-33, 2019.