faKy: A Feature Extraction Library to Detect the Truthfulness of a Text

Sandro Barres Hamers[1] [0009−0002−1016−5610] and
Davide Ceolin[2] [0000−0002−3357−9130]

[1] Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
sbarreshamers@gmail.com
[2] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
davide.ceolin@cwi.nl

Abstract. The transparency and explainability of fake news detection is a crucial feature to enhance the trustability of the assessments and, consequently, their effectiveness. Textual features have shown their potential to help identify fake news in a transparent manner. In this paper, we survey a list of textual features, evaluate their usefulness in predicting fake news by testing them on a real-world dataset, and collect them in a Python library called "faKy".

Keywords: faKy · Natural Language Processing · Fake News detection · Feature extraction.

1 Introduction

Fake news has always been a phenomenon known to humankind [11]. Nevertheless, the Web and Social Network Systems (SNS) in particular, exacerbated the societal threats posed by misinformation for two reasons. First, people find it hard to distinguish information from misinformation [36]; this is becoming harder with the rise of state-of-the-art Artificial Intelligence (AI). Secondly, fake news can spread extremely fast on SNS [8]. Therefore fake news can reach a high volume of consumers quickly.

In this paper, we look at the interpretability of fake news assessments. While there are numerous Neural Networks (NN) and Large Language Models (LLM) that can accurately classify fake news with very high accuracy, in some cases up to 99% [34], these models, most of the time, lack interpretability: their reasoning is hardly interpretable by humans. Recent work shows that very accurate models tend to be less interpretable [7]; this phenomenon is called the interpretable accuracy trade-off.

We study whether linguistic features, obtained using Natural Language Processing (NLP), can provide a basis for assessing fake news. Among such features, we consider Readability, which measures the ease with which the text is read. Additionally, we investigate the Information Complexity (IC), which quantifies the amount of information contained in the object, and conduct sentiment analysis to assess the emotional tone of the text. Subsequently, we analyze Named

Entity Recognition (NER) to identify instances where the object represents specific individuals, places, or other proper nouns in the text. Lastly, we employ Part of Speech (POS) Tags to determine a sentence's grammatical category or syntactic function. These features are carefully selected and based on existing literature, which we elaborate on in the related work section.

The novelty of our contribution is in faKy[3], an extensive library that collects a comprehensive list of NLP features known to have shown a correlation with fake news assessment. These features (and, consequently, the faKy library) are here aggregated, tested, and evaluated on real-world datasets. In this manner, faKy provides a validated toolkit for extracting features from a text that are potentially correlated to fake news, thus contributing to the explainability of the assessment process.

The overarching research question that we address is:

RQ: Can the truthfulness of textual information be accurately predicted using specific linguistic features, and how do these linguistic features contribute to distinguishing between truthful and untrustworthy textual content?

We decompose this question through the following subquestions:

SRQ1: *Does the readability measure of a text provide a basis to predict its truthfulness?*
SRQ2: *Does the IC measure of a text provide a basis to predict its truthfulness?*
SRQ3: *Does the sentiment of a text provide a basis to predict its truthfulness?*
SRQ4: *Do the of Named Entities recognized in a text provide a basis to predict its truthfulness?*
SRQ5: *Do the POS tags in a text provide a basis to predict its truthfulness?*

The rest of the paper is structured as follows. Section 2 introduces related work. Section 3 introduces the research methodology and the experimental design. Results are discussed in Section 4, while Section 5 concludes the paper.

2 Related work

The fake news research body is extensive and much work has been done to understand and classify fake news. This is not surprising as this phenomenon threatens our society's foundations.

A recent study presented a comprehensive review of methods for detecting fake news on SNS. They include multiple techniques like ML, NLP, and information propagation analysis, which looks at how the different agents behave in the SNS ecosystem. They discuss content-based, network-based, and hybrid approaches, as well as machine and deep learning models. The paper also highlights the challenges and limitations of fake news detection. The authors found

[3] faKy repository

that readability features impact fake news detection and recommend considering them in developing detection systems [5].

Subsequently, a benchmarked study of 19 ML models on three different datasets, including the Liar dataset, demonstrated the superior performance of BERT-based models with the best f1-score of 62% for the Liar dataset. Additionally, they show an F1 score of 57% for a Naive Bayes and an F1 score of 51% for tree-based models for the Liar dataset. The researchers evaluated the models on accuracy, recall, precision, and F1-score. However, the researchers did not use k-fold-cross validation, which could have improved their evaluation [28].

Another popular method to extract information from a text object is NER and POS tagging, which looks at an object's structure, style, and content. A recent study proposes a linguistic method to detect fake news that can be applied to any language. They compare news articles using POS Tags and NER and train four ML models. They evaluate the model's performance with the F1-score and show that a Gradient Boosting model has the highest score, with an average of 70.83%. The paper presents a novel approach that delivers high-level performance using POSTag+NER features [39].

Lastly, using morphological tags and n-grams in decision tree-based ML algorithms demonstrates superior accuracy, precision, and F1 score performance in the scope of fake news [26]. The authors extract n-grams from the tags and use them for training decision trees in machine-learning algorithms where an n-gram is the probability distribution for the following word, given the corpus size. They use several n-grams (1-gram, 2-gram) for words and POS tags. Where a 2-gram considers two words, and a 3-gram three words, their approach outperforms other models in accuracy, precision, and F1 score. They argue that future work should explore more sophisticated models and linguistic features. However, these linguistic features have also shown promising results in predicting a broad set of information quality aspects (beyond the mere veracity prediction considered in fake news detection) by supporting argument computation [13].

3 Research Methodology and Experimental Design

As is common in many NLP problems, the classification of fake news can be formulated mathematically as an optimization task, represented by equation 1. This equation captures the essence of the language processing algorithm's two main modules. The search module's objective is to discover the optimal output y^* that maximizes the scoring function Ψ given an input x and the possible outputs y from the set $Y(x)$. The learning module is responsible for iteratively adjusting the parameters θ to minimize a loss function, $L(y, y*)$ which quantifies the disparity between the true outcomes y and the generated ones y^* during the learning process [17].

$$y^* = \arg\max_{y \in 0,1} \Psi(x, y; \theta) \tag{1}$$

We will employ an incremental experimental design, allowing for a step-by-step breakdown of the various variables. The experimental design is based on the methodology presented in [2] and has been slightly modified to align with the requirements of this specific experiment. The experiment comprises four main parts, visually depicted in Figure 3: data collection, experimental pipeline, classification, and validation. The subsequent subsections will provide a detailed examination of these components.

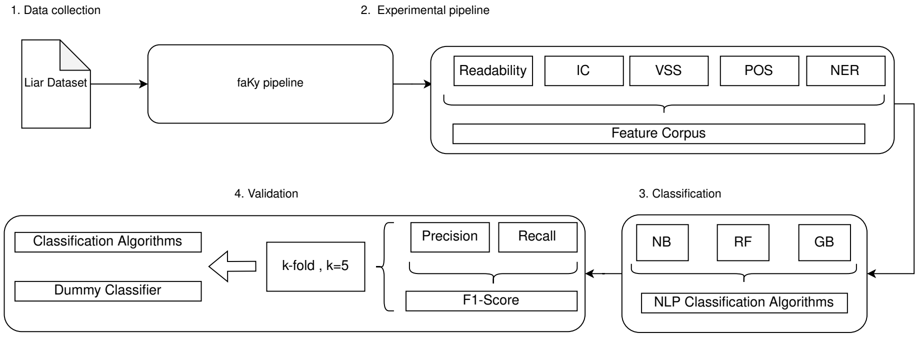

Fig. 1. Experimental Design

3.1 Data collection

We use the Liar dataset, obtained from Hugging Face[4]. The Liar dataset is a publicly available resource for fake news detection and consists of 12.8 thousand manually labeled statements collected from various contexts. These statements were originally sourced from PolitiFact, a non-profit organization known for fact-checking the accuracy of claims. PolitiFact provides detailed analysis reports and links to source documents for each case. Furthermore, each statement in the dataset has been evaluated by PolitiFact editors for its truthfulness. By utilizing the Liar dataset from Hugging Face, we aim to reduce label bias in our research [21]. We categorized the statements into three categories: **True**, **False**, and **In Between (IB)** claims. We classify True claims as the 'negative class' (0), False claims as the 'positive class' (1), and IB claims as (2). It should be noted that we assigned these labels on a qualitative rationale. The Liar train dataset consists of 10239 rows; Table 1 provides an overview of the distribution of claims in the dataset. The Table includes four columns: "Label," "Claim," "Number of statements," and "Percentage of total."

3.2 Experiment pipeline

This subsection discusses the NLP pipeline, which involves converting text objects from the Liar dataset into a machine-readable format using the spaCy

[4] https://huggingface.co/datasets/liar

Table 1. Number of claims

Label	Claim	Number of statements	percentage of total
0	True	1676	16%
1	False	2833	28%
2	IB	5730	56%

library[5]. By applying the spaCy nlp function, the text objects are transformed into doc objects, which undergo tokenization and processing through the spaCy pipeline. Features are computed and integrated into the pipeline, creating the feature extraction library faKy. FaKy utilizes spaCy extensively and is therefore named after it. The spaCy library offers comprehensive NLP and linguistic capabilities, ensuring efficient and accurate execution. The subsequent paragraphs illustrate the computation of different features.

Readability The readability of a text tells us how easy it is to read and understand a particular text. A text with low readability indicates that the text consists of complex and unique words; this can be, for example, an academic paper. We compute the readability via the Flesch-Kincaid Reading Ease (FKE) score; we chose FKE for three reasons. First, the FKE score can be computed with the spaCy readability object[6], enabling the spaCy pipeline's use. Secondly, the FKE score is developed for English text, the target language for this study. At last, FKE is ubiquitous and the standard test of readability for documents and forms for the US military [29], insurance policies [40], and word-processing programs [1].

$$\mathrm{FKE} = 206.835 - 1.015\frac{\mathrm{TW}}{\mathrm{TS}} - 84.6\frac{\mathrm{TSL}}{\mathrm{TW}} \tag{2}$$

The FKE score is computed based on the equation represented in equation 2. The FKE is established on the total number of words (TW), the total number of sentences (TS), and the total number of syllables (TSL). The FKE estimates a text's readability by estimating the ratio of words to sentences and syllables to words. A higher FKE score thus means that a text consists of shorter sentences and fewer syllables per word and is thus easier to read, enhancing readability.

Complexity In fake news research, linguistic features have been the primary focus for distinguishing fake and True objects. However, we propose a novel approach by incorporating the concept of IC derived from algorithmic information theory. We posit that the IC encapsulates crucial information and can be computed relatively easily, making it an interesting feature for investigation.

We use the Kolmogorov complexity to compute the IC of an object; the IC tells us how much information an object conveys. We define the Kolmogorov

[5] `https://spacy.io/`

[6] `https://spacy.io/universe/project/spacy_readability`

complexity $C(x)$ of a string x as the length of the shortest program p that processes x running on a universal Turing Machine U. The formula conveys the following idea: A string with low information complexity is highly compressible as the information it contains can be encoded in a program much shorter than the length of the string itself [47].

$$C(x) = min_p\{\text{length}(p) : U(p) = x\} \tag{3}$$

We mathematically represent the Kolmogorov complexity in equation 3. Nonetheless, $C(x)$ is proven to be uncomputable because finding the shortest program is equivalent to solving the halting problem, which is undecidable [43]. This means that no general algorithm or formal system can accurately determine the halting behavior of all programs and inputs, making the computation of $C(x)$ impossible [24]. We can thus only approximate $C(x)$ due to its uncomputable nature. We do this by using a compressing algorithm since the $C(x)$ may be roughly described as the compressed size [38]. The IC is thus computed by applying a compressing algorithm to the object and returning the compressed size of that object.

Sentiment We conduct the sentiment analysis using the Vader Sentiment Scores (VSS). VSS measures the polarity (positive, negative, neutral) and the intensity of the emotion for a given text. A very negative sentiment will thus have a different score than a moderately negative object. VSS relies on a lexicon of words and phrases with known sentiment values, as well as rules to capture sentiment in context. VSS has demonstrated superior performance in detecting sentiment intensity compared to other sentiment analysis models [22], and it has also proven effective in fake news detection systems [6]. Computation of VSS was implemented using the NLTK library [7].

Sequence labeling We adopt two sequence labeling tasks, NER and POS tagging. As discussed earlier, NER and POS tagging have proven valuable in detecting fake news. Furthermore, NER tags provide semantic representations, such as identifying events and relationships [25]. On the other hand, POS tagging is the first step toward syntactic analysis (which, in turn, is often helpful for semantic analysis).

We compute both the total number of NER tags as well as the count per NER category. We represent this by means of a numerical vector.

$$\text{NERvector} = \left[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0\right] \tag{4}$$

The initialized vector represents the frequency of each unique NER tag, where the first position corresponds to the 'CARDINAL' tag, the second corresponds to the 'DATE' tag, and so on.

[7] `https://www.nltk.org/howto/sentiment.html`

Syntactic labeling Unlike NER tagging, POS tagging is a dependent feature that relies heavily on the context and placement of words within a sentence. To illustrate this point, let us examine the same object used in NER tagging but apply dependency parsing, which results in specific POS tags. Within dependency parsing, we describe the syntactic structure of a sentence in terms of directed binary grammatical relations between the words [25].

We compute the POS tags similar to the NER tags, except we are only interested in the sum of individual POS tags per object since computing the total count of the pos tags will count approximately every tokenized word, giving us no relevant information in the structure or style of the object.

3.3 Classification

The classification is a 3-way multi-class text classification problem classifying, True, False, and 'IB' claims. We employ three machine learning models: Naive Bayes (NB), Random Forest (RF), and Gradient Booster (GB). We choose these models since an NB performs highly due to the computation of conditional probabilities [4], an RF has shown superior performance in classifying fake news [2], and at last, a GB since it is advantageous in cases of unbalanced datasets and outperforms an RF with its robustness [31]. We use the Python-based ML algorithms from sci-kit-learn [8].

Since the dataset is unbalanced, we use an oversampler to avoid poor performance of the minority class, after oversampling the minority class, the train and test data are equally distributed; the distributions of these claims are presented in Table 2.

Table 2. Train instance counts before and after oversampling

Label	Class	Counts (initial)	Counts (after oversampling)
0	True	1676	4565
1	False	2833	4565
2	IB	5730	4565

Since the RF and GB are tree-based models, there is no need to scale the features; also, the NB does not need scaled features since it is a probabilistic model and thus only looks at the frequency of the features.

3.4 Evaluation

We validate if the features θ significantly differ between the three independent groups corresponding to the labels (True; 0, False; 1, IB; 2). We first plot the distributions of the features and test if they are parametric through the Kolmogorov-Smirnoff Test (KST). We conduct the KST for the continuous features: Readability, IC, and VSS; discrete features, represented by POS and NER

[8] `https://scikit-learn.org/stable/`

tags, are inappropriate for the KST since the discrete features are not used to measure some properties but solely as a count of tags [32]. The results are shown in this repository[9].

Next, we perform the Kruskal-Wallis Test (KWT) to validate the features. The KWT is chosen due to its suitability for analyzing data involving more than two independent groups, unbalanced datasets, and non-parametric distributions [46]. To determine significant differences between pairs of independent groups, we employ a post-hoc method. The Dunn test is selected as the most suitable post-hoc method for conducting pairwise comparisons [16]. We set the significance threshold at $p < 0.05$ since it is the industry standard [30]

Finally, a preliminary examination is undertaken, encompassing the computation of the mean, and standard deviations across the three labels for both continuous and discrete features, we compute the highest value for the continuous features. This analysis facilitates a fundamental comprehension of the data distributions, giving us a basic main difference between the different claims. We validate the performance of Ψ in combination with θ, and we conducted three validation methods. First, we assessed the performance of Ψ without any feature selection, thus incorporating all the computed features. Next, we evaluated the performance when selecting two-way significant interaction features. These features demonstrate significance between at least two of the three groups, for example, (True; False or IB; False, and so on). Finally, we computed the evaluation of three-way significant features, which are significant across all three groups, denoted as (True; False, True; IB, False; IB). We evaluated the performance of Ψ using the F1 score and k-fold cross-validation. The F1 score is selected due to its industry standard as the harmonic mean of precision and recall [20]. K-fold cross-validation divides the dataset into five subsets (k=5), which is considered a good compromise [19]. With k-fold cross-validation, we mitigate overfitting by applying Ψ to the k subsets. We compute the Coefficient Variance (CV), depicted in equation 5; the CV considers the spread of the distribution for the variation. High variance, indicated by $CV > 1.0$ [27], suggests inconsistent performance across subsets and is, thus, sensitive to overfitting.

$$CV = \frac{Standard\ deviation}{Mean} \tag{5}$$

At last, we compute a baseline evaluation to put the model's performance into perspective; we employ the 'uniform' dummy classifier since it generates uniform random predictions based on the three classes. We compare the F1 score performance of the ψ to the Dummy Classifier (DC) using Relative Improvement (RI). The RI returns a positive result if the ψ F1-score is high in comparison to the DC F1-score, and a negative result if it is low, the RI is depicted in equation 6 [14].

$$RI = \frac{F1_{\psi} - F1_{DC}}{F1_{DC}} \tag{6}$$

[9] Feature distributions repository

4 Results and Discussion

We present the findings of the conducted research in two distinct parts, aligning with the overarching objective of the experiment: validate if the features θ can maximize the output y^* through the natural language algorithm Ψ. The evaluation results for the features and ML algorithms are presented in the following repository [10]. Throughout this section, we will address the sub-research question of Section 1.

4.1 Distinguishing True and False Claims: Insights from IC, Readability, VSS, NER, and POS tags.

The statistical results of the significant continuous features: Readability and IC are presented in Table 3 where we show the features corresponding mean, max, mode, and standard deviation associated with the three labels.

The IC is a significant feature between True, False, and IB claims since the p-value is lower than 0.05 between the three labels. Therefore, the truthfulness of textual information can be distinguished based on the IC. True claims convey more information compared to False claims as depicted in Table 3. Moreover, False claims demonstrate a greater distribution, suggesting that the information they convey is more scattered than True claims. This becomes more evident since the maximum value for the IC is higher for False claims while the mean is lower, indicating the scattered nature of the IC for False claims.

Similar to the IC, the Readability of an object is also a significant feature since the p-values are lower than 0.05; we thus conclude that Readability plays a crucial role in determining the truthfulness of a text. Our analysis further reveals that False claims tend to have increased difficulty regarding Readability, reflecting more complex word choices and sentence structures, see Table 3. This observation aligns with the existing literature [41]. However, comprehending the underlying factors contributing to this distinction needs interdisciplinary research. Additionally, we find a similar pattern observed for the IC, where False claims exhibit a greater degree of distribution, indicating a dispersed nature in their Readability compared to True claims. This discovery corresponds with the findings of [12], who demonstrated that deceptive content encompasses a wide range of readability levels.

The VSS are not significant between True and False claims since the p-value is greater than 0.05; thus, the VSS of a text is not suitable for determining the truthfulness of a text. While we show that the VSS of text conveys no relevant information to determine the truthfulness of a text, it is essential to note that the VSS is tailored explicitly for analyzing SNS text objects. In contrast, this study's investigation subject is focused on political claims. Consequently, the utilization of VSS in this context lacks its specific purpose. This limitation deserves attention, and future research should consider exploring the significance

[10] Fake News Classification repository

Table 3. Significant continuous features.

Feature	Label	Average	Maximum	Mode	Standard Deviation
IC	True	8555	149837	5592	4779
IC	False	8130	197589	5964	4929
IC	IB	8775	294076	5951	5112
Readability	True	60	127	74	21
Readability	False	56	124	60	22
Readability	IB	59	151	56	22

Table 4. ML models performance.

Model	Feature Selection	F1 Score Test (%)	RI (%)
Naive Bayes	**Unselected**	**34.11 (+/- 6.44)**	**7.00**
Naive Bayes	two-way significant	33.06 (+/- 11.07)	3.71
Naive Bayes	three-way significant	27.35 (+/- 2.48)	-14.19
Random Forrest	Unselected	27.59 (+/- 7.77)	-13.44
Random Forrest	two-way significant	29.40 (+/- 6.09)	-7.76
Random Forrest	**three-way significant**	**31.72 (+/- 6.49)**	**-0.47**
Gradient Booster	Unselected	30.48 (+/- 9.72)	-4.37
Gradient Booster	**two-way significant**	**30.94 (+/- 12.86)**	**-2.93**
Gradient Booster	three-way significant	28.92 (+/- 6.98)	-9.27
Dummy Classifier Uniform	Nonapplicable	31.88 (+/- 2.33)	0.00

of VSS on SNS objects or employing alternative sentiment analysis approaches to assess the truthfulness of the claims.

In the appendix we present an overview of the statistical results for the discrete features, we show a significant difference between False and True claims regarding NER tags. On average, True claims exhibit more NER tags than False claims, which aligns with the existing literature [37]. An explanation for this would be that False claims reference fewer existing entities due to their fictitious nature. However, we acknowledge that such a conclusion is overly simplistic, necessitating further investigation into the underlying factors driving this behavior. The only NER tags that are more present in False than True claims are 'PERSON' and 'ORG'; this may be related to the fact that fake news specifically targets political figures; think of the pizza gate incident or conspiracy groups like Qanon [42]. Moreover, the emphasis on Organizations is reasonable, given that fake news commonly targets large pharmaceutical entities like Pfizer, as well as other major corporations and government institutions [10, 15, 23]. This phenomenon warrants comprehensive investigation to better comprehend its underlying causes. Lastly, we conclude that the specific style and syntax indicated by POS tags can distinguish the truthfulness of textual content. We do, however, not see that the specific POS tag groups: prepositions, adjectives, and nouns, appear more in False than in True claims; this contradicts the existing literature [26]. This may be due to the use of a different dataset or methodology than that adopted in the literature. Furthermore, our study demonstrates that True claims exhibit a greater prominence of POS tags, indicating greater linguistic diversity. Notably, False claims display a higher occurrence of 'Verb' and 'Part' POS tags (e.g., 'to go') when compared to True claims.

In summary, this study reveals notable differences between True and False claims. Consequently, False claims exhibit substantial distinctions in their linguistic structures. Additionally, empirical evidence demonstrates that False claims, on average, embody a higher level of distribution compared to True claims. This variation may be attributed to the fictitious characteristics inherent in False claims, in contrast to True claims that adhere to prescribed formats dictated by established standards.

4.2 Classifying fake claims: Insights from NB, RF, and GB.

The outcomes of the three classification ML models with the corresponding feature selections are illustrated in Table 4. We show the mean F1-scores for the test data, acquired through k-fold cross-validation with k=5. Additionally, the Table includes the CV, as well as the model's RI in terms of the DC.

NB combined with no feature selection exhibited the highest performance, achieving an F1 score of 34.11%, with a CV of 6.44%. The high CV suggests potential overfitting, a finding further supported by the overperformance of the NB model on the training data. Compared to the DC, the NB model showed an RI of 7.00%. When applying an NB with the three-way significant features the model performance drops drastically, this could be because of the decreased number of features, Subsequently, we also notice that applying an NB with two-way significant features decreases the model's performance.

Subsequently, the RF model demonstrated the best F1 score when using three-way significant features, with a score of 31.72% for the test data, accompanied by a CV of 6.49%, indicating high variance. Notably, the training data significantly outperformed the test data in this instance, with a difference of approximately 45%, indicating extreme overfitting by the RF model. Next to the overfitting, the RF using three-way significant features showed a negative RI of -0.47% compared to the DI. We notice that the RF's performance declines when using two-way feature selection and reaches the lowest score when using no feature selection This can be attributed to the increased complexity introduced by the additional splits or the inclusion of potentially noisy or redundant features. Future research should analyze the importance of the individual features within the RF model and the way they interact with each other [9].

Lastly, the GB model achieved the poorest performance with the best F1 score of 30.94%, and a CV of 12.86%, when applying two-way significant features. Subsequently, the GB underperformed when compared to the DC, with an RI score of -2.93%. Based on our findings we thus conclude that the GB performs the best with two-way significant features however, this is a preliminary finding. Future research should focus on feature selection optimization [3, 45].

Previous studies achieved F1 scores of 50-60% for similar algorithms, for binary classification tasks [28], where random performance is typically around 50% and tends to yield higher F1 scores [33]. In contrast, our study addressed a more challenging three-class classification task: True, False, and In Between. This distinction is vital for interpreting our results. While our models outperformed the baseline established by [44], it's important to note that our task was less demanding than the 6-way multiclass text classification problem in the same paper. Despite the inherent complexity of our task, our models achieved an accuracy of approximately 30%, showcasing promising results. Future research should explore the performance of different ML algorithms using our introduced features across diverse classification problems and different datasets.

All three models exhibited better performance on the majority class (TN) than on the minority class (TP). Moreover, they demonstrated better performance on the training data, indicating overfitting, and displayed high variances. Future research should prioritize improving the models by addressing overfitting through increased data volume and enhanced model robustness.

The model's performances align with the overall behavior of the models since NB classifiers can quickly learn to use high-dimensional features with limited training data compared to more sophisticated methods like the RF [18], therefore the NB will perform better with a higher number of features compared to the RF which achieves high results with the most relevant features. We thus conclude that the model's performance is dependent on the applied feature selection.

To conclude, our study presented the performance of three ML algorithms with F1 scores ranging from 27.35% to 34.11%. We showed relative improvement for the NB when applying no feature selection and when applying two-way significant features. However, it is essential to note that the observed relative improvement of the NB may not be statistically significant. The statistical difference between the two may not be large enough to discriminate the items effectively, as the two distributions may overlap.

4.3 Limitations & Future Research

The POS and NER tags are the first step in understanding text objects' lexical and syntactic information. However, more sophisticated methods can be employed to extract and leverage this information effectively. For instance, TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical representation used in NLP that measures the importance of a term in a document by considering its frequency in the document and its rarity across the entire document collection. Furthermore, exploring advanced NLP techniques such as dependency parsers can enable the analysis of object styles and facilitate more advanced tasks like semantic parsing. Subsequently, the style of the objects could be further analyzed using dependency parsing, which, as a result, can be generalized to even more advanced NLP tasks such as semantic parsing, taking the text's actual meaning into account. Lastly, we propose using n-grams to analyze and model language patterns. Another area for future research is the optimization of the faKy library's runtime. While the current implementation utilizes the efficient spaCy pipeline, further improvements can be made to enhance its runtime capabilities. Investigating optimal data structures and pipeline components can significantly optimize the runtime performance, making the library even more efficient. Finally, it is crucial to recognize the ethical considerations surrounding this research, despite its aim to combat the spread of false information. The study introduces the concept of dual-use, where the same technology can be used for beneficial and potentially harmful purposes [35]. While FaKy has the potential to detect fake news, it could also be utilized by oppressive regimes to categorize dissenting texts, resulting in Orwellian practices. Therefore, the high risk of dual-use needs further examination and consideration.

5 Conclusion

Drawing upon the findings obtained from the addressed sub-research questions, we can respond to the main research question, RQ1: Can the truthfulness of textual information be accurately predicted using specific linguistic features? Our study concludes that linguistic features can accurately predict the truthfulness of a text. We show that fake objects have a greater distribution across all the features, are more complex in terms of readability, convey more information, hold more Named Entities, and significantly differ between style and syntax regarding Part-of-Speech tags. Consequently, we reject the null hypothesis and accept the alternative hypothesis. Additionally, we introduce faKy, a comprehensive feature extraction library that computes relevant linguistic features for fake news detection. Our study highlights the significance of these features and shows that by combining them with machine learning classification algorithms the truthfulness of text objects can be predicted. While faKy is still in its early stages of development, our findings indicate its potential in combating fake news and advancing explainable AI.

Acknowledgements

This research has been partly supported by the Netherlands eScience Center project "The Eye of the Beholder" (project nr. 027.020.G15) and by the AI, Media & Democracy Lab (https://www.aim4dem.nl/).

References

1. How to Use Readability Scores in Your Writing — Grammarly Spotlight, 4 2020.
2. Hugo Queiroz Abonizio, Janaina Ignacio de Morais, Gabriel Marques Tavares, and Sylvio Barbon Junior. Language-independent fake news detection: English, portuguese, and spanish mutual features. *Future Internet*, 12:87, 5 2020.
3. Afek Adler and Amichai Painsky. Feature importance in gradient boosting trees with cross-validation feature selection, 09 2021.
4. Abdulaziz Albahr and Marwan Albahar. An empirical comparison of fake news detection using different machine learning algorithms. *International Journal of Advanced Computer Science and Applications*, 11, 2020.
5. Ihsan Ali, Mohamad Nizam Bin Ayub, Palaiahnakote Shivakumara, and Nurul Fazmidar Binti Mohd Noor. Fake news detection techniques on social media: A survey. *Wireless Communications and Mobile Computing*, 2022:1–17, 8 2022.

6. Miguel A. Alonso, David Vilares, Carlos Gómez-Rodríguez, and Jesús Vilares. Sentiment analysis for fake news detection. *Electronics*, 10:1348, 6 2021.
7. Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.
8. Nicolas Belloir, Wassila Ouerdane, and Oscar Pastor. Characterizing Fake News: A Conceptual Modeling-based Approach. In *ER 2022 - 41st International Conference on Conceptual Modeling*, Hyderabad, India, October 2022.
9. Gèrard Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13:1063–1095, 2012.
10. Robert Blaskiewicz. The big pharma conspiracy theory. *Medical Writing*, 22:259–261, 12 2013.
11. Joanna M Burkhardt. Combating fake news in the digital age. *Library Technology Reports*, 53, 2017.
12. Carlos Carrasco-Farré. The fingerprints of misinformation: how deceptive content differs from reliable sources in terms of cognitive effort and appeal to emotions. *Humanities and Social Sciences Communications*, 9:162, 5 2022.
13. Davide Ceolin, Giuseppe Primiero, Michael Soprano, and Jan Wielemaker. Transparent assessment of information quality of online reviews using formal argumentation theory. *Information Systems*, 110:102107, 2022.
14. college of san mateo. 250 i-2.
15. Malaysian Communications and Multimedia Commission, Aug 2022. [Accessed 14-Jun-2023].
16. Alexis Dinno. Nonparametric pairwise multiple comparisons in independent groups using dunn's test. *The Stata Journal: Promoting communications on statistics and Stata*, 15:292–300, 4 2015.
17. Jacob Eisenstein. Natural language processing, 2018.
18. Shuzhan Fan. Understanding the mathematics behind Naive Bayes — shuzhanfan.github.io. `https://shuzhanfan.github.io/2018/06/understanding-mathematics-behind-naive-bayes/`. [Accessed 15-Jun-2023].
19. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York, 2009.
20. Haibo He and Yunqian Ma. *Imbalanced Learning*. Wiley, 6 2013.
21. Dirk Hovy and Shrimai Prabhumoye. Five sources of bias in natural language processing. *Language and Linguistics Compass*, 15, 8 2021.
22. C. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8:216–225, 5 2014.
23. Reuters Institute. Types, sources, and claims of COVID-19 misinformation — reutersinstitute.politics.ox.ac.uk. `https://reutersinstitute.politics.ox.ac.uk/types-sources-and-claims-covid-19-misinformation`. [Accessed 14-Jun-2023].
24. Mike James. Programmer's guide to theory - kolmogorov complexity, 6 2020.
25. Daniel Jurafsky and James H. Martin. Dependency parsing, 1 2023.
26. Jozef Kapusta, Martin Drlik, and Michal Munk. Using of n-grams from morphological tags for fake news classification. *PeerJ Computer Science*, 7:e624, 7 2021.
27. Joshka Kaufmann. What do you consider a good standard deviation?, 09 2014.

28. Junaed Younus Khan, Md. Tawkat Islam Khondaker, Sadia Afroz, Gias Uddin, and Anindya Iqbal. A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, 4:100032, 2021.
29. J. D. Kniffin. The new readability requirements for military technical manuals. *Technical Communication*, 26(3):16–19, 1979.
30. Giovanni Di Leo and Francesco Sardanelli. Statistical significance: p value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach. *European Radiology Experimental*, 4:18, 12 2020.
31. Olga Lyashevska, Fiona Malone, Eugene MacCarthy, Jens Fiehler, Jan-Hendrik Buhk, and Liam Morris. Class imbalance in gradient boosting classification algorithms: Application to experimental stroke data. *Statistical Methods in Medical Research*, 30:916–925, 3 2021.
32. Salvatore S. Mangiafico. R Handbook: Introduction to Parametric Tests — rcompanion.org. `https://rcompanion.org/handbook/I_01.html#:~:text=It%20is%20sometimes%20permissible%20to,data%20or%20other%20discrete%20data`, 2016. [Accessed 15-Jun-2023].
33. Pablo Del Moral, Slawomir Nowaczyk, and Sepideh Pashami. Why is multiclass classification hard? *IEEE Access*, 10:80448–80462, 2022.
34. Jamal Abdul Nasir, Osama Subhani Khan, and Iraklis Varlamis. Fake news detection: A hybrid cnn-rnn based deep learning approach. *International Journal of Information Management Data Insights*, 1(1):100007, 2021.
35. National Institute of Health. Dual-use research, 2022.
36. Gordon Pennycook, Jonathon McPhetres, Yunhao Zhang, Jackson G. Lu, and David G. Rand. Fighting covid-19 misinformation on social media: Experimental evidence for a scalable accuracy-nudge intervention. *Psychological Science*, 31:770–780, 7 2020.
37. Mohammadreza Samadi and Saeedeh Momtazi. Fake news detection: deep semantic representation with enhanced feature engineering. *International Journal of Data Science and Analytics*, 3 2023.
38. Alexander Shen. Around kolmogorov complexity: basic notions and results. *CoRR*, abs/1504.04955, 2015.
39. Marcos A. Spalenza, Leopoldo Lusquino Filho, Felipe Maia Galvão França, Priscila Machado Vieira Lima, and Elias de Oliveira. Lcad - ufes at fakedes 2021: Fake news detection using named entity recognition and part-of-speech sequences. In *IberLEF@SEPLN*, 2021.
40. Florida Statutes. Florida statutes section 627.4145 - readable language in insurance policies. (fla. stat. § 627.4145), 9 2016.
41. Tavakoli Mohammadali T, Alani Harith, and Burel Grégoire. On the readability of misinformation in comparison to the truth. 4 2023.
42. Marc Tuters and Tom Willaert. Deep state phobia: Narrative convergence in coronavirus conspiracism on instagram. *Convergence: The International Journal of Research into New Media Technologies*, 28:1214–1238, 8 2022.
43. Paul M.B. Vitányi. How incomputable is kolmogorov complexity? *Entropy*, 22:408, 4 2020.
44. William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. pages 422–426. Association for Computational Linguistics, 2017.
45. Zhixiang Eddie Xu, Gao Huang, Kilian Q. Weinberger, and Alice X. Zheng. Gradient boosted feature selection. 1 2019.
46. Yury Zablotski. Kruskal–wallis test: compare more then two groups, 9 2019.
47. Hector Zenil. A numerical method for the evaluation of kolmogorov complexity, an alternative to lossless compression algorithms. 7 2011.