Article Genuineness Classification

Vrutanjali Rakesh Patel David Weissteiner Third Co-author

June 29, 2023, Graz

Abstract

Our project aims to develops a system for classifying whether a given news are real or generated. The challenges we face were extracting useful information and collecting training data for model training and evaluation. The methodology includes collecting document characteristics, assembling the learned distributions, and scoring the deviation of the articles.

1 Introduction

1.1 Problem Statement

The overall goal of this project is to develop a system which classifies whether a given news article is real or generated. More specifically, the system receives a set of text files —each containing a news article— and outputs which of them is 'real' and which of them is 'generated'. Every news article has between 50 and 100 sentences. Moreover, the set of documents contains 60% genuine articles and 40% "fake" articles. The genuine articles originate from an existing news source (e.g., newspaper, online news, ...). The generation method of the generated articles (40%), however, is limited to one of the following three generation techniques: (1) 20% N-gram based, (2) 10% Students' own implementation, and (3) 10% GPT based. Anyhow, the differentiation between these generation methods is not required. The task solely consists of classifying whether the article is generated or genuine.

1.2 Challenges

From the Problem Statement in Section 1.1, we identify two major challenges: (1) Extracting useful information from the document and (2) collecting training data for model training and evaluation. Once we have useful information about the documents (together with their interpretation), the construction of a respective classifier does not create a big challenge.

Information Extraction

One main challenge of this project is to extract useful information from a given document, which can then be used to solve the classification task. Hence, the extracted information should give an indication whether the document was taken from a real source (i.e., written by a human) or has been generated.

Training Data

Since we do not get task-specific training data beforehand —neither labelled nor unlabelled—, we face the challenge of obtaining training data. Without any kind of training data, it is impossible to evaluate whether the extracted information is useful in advance.

1.3 Idea

Information extraction from text documents can be addressed in many different ways due to the high complexity of the natural language. Depending on the task, different types of information have to be extracted in order to solve the problem. Since we are unaware of which type of information characterizes a potential difference between genuine articles and computer-generated articles, we stick to multiple simple statistics (Section 3.1) of text corpora with the belief that the deviation of generated articles from real articles is captured by at least one characteristic.

Unfortunately, collecting a whole training dataset with real and generated articles is beyond our project extent, since we need a specific article generator to match our requirements. We overcome this problem by focusing on authentic articles (i.e., not considering fake articles during training), which we can easily collect from the internet. For this

purpose, we then try to learn the distribution of selected document features from real news articles. Thereby, we are able to identify deviations from this learned distribution which likely indicate a different writing process (e.g., computer-generated writing).

2 Related Work

While fake news is not a new concept, it has become more prevalent in the digital age due to the availability of digital tools for mass media and the ease of spreading information through social media platforms and bots. A wide range of digital tools have been developed to detect and counter the spread of fake news, which can be broadly classified into five categories: Language-based approach, Topic-agnostic approach, Machine learning approach, Knowledge-based approach and Hybrid approach.

Language Approach

The language-based approach focuses on the study of linguistic structures and lexical choices. There are three main methods that contribute to this approach: (1) Bag of Words (BOW), (2) Semantic Analysis, and (3) Deep Syntax.

Topic-Agnostic Approach

The topic-agnostic approach uses linguistic and web markup features to identify fake news. For example, a recent state of the art study in this field describes that topic-agnostic features are the presence of a large number of advertisements or the presence of an author's name [Horne and Adali(2017)].

Machine Learning Approach

Machine learning (ML) algorithms can be used to identify fake news using training data sets. For example, a state-of-the-art approach in this area is the use of the Twitter crawler, which is a machine learning approach that helps to identify and further prevent the spread of false information through fake likes, comments and accounts [Atodiresei et al.(2018)].

Knowledge Based Approach

The knowledge-based approach integrates machine learning and knowledge engineering to detect fake news. In short, this approach tries to use external sources to verify whether the news is real or fake and further to identify the news before it spreads faster [de Beer and Matthee(2020)].

Hybrid Approach

Hybrid models combine several approaches for a comprehensive fake news detection mechanism. A current state-of-the-art example of a hybrid model is CSI (capture, score, integrate). It operates on three main elements, namely the steps of extracting representations of articles using RNN, creating a vector of scores and representations, integrating the results of capture and score, resulting in a vector that is used for classification [Ruchansky et al.(2017)].

Our work aligns most closely with the language approach and the machine learning approach.

3 Methodology

With the belief that real articles differ from generated articles in simple document characteristics, we collect such characteristics (also called 'statistics') from real news articles to learn their distributions. We provide a description and a brief overview of the collected statistics in Section 3.1. Furthermore, we explain how we approached the task of 'measuring' the deviation of an article by density estimation, averaging, and scoring in Section 3.2.

3.1 Document Statistics

In order to combine multiple characteristics of the documents, we need a common shape of the resulting statistics. Therefore, we limit the information extraction to methods which map the document to a single, real-valued number (i.e., $f : \mathcal{D} \to R$). Anyhow, we can split a multi-dimensional output (i.e., R^n) into separate real-valued numbers if the dimensions (and their interpretation) stay the same for different documents.

The collected statistics range from simple counts (e.g., the count proportion of a specific word or n-gram, the number of words per paragraph, etc.) up to more advanced statistics (e.g., deviation from the Zipf's law, etc.). Additionally,

we can easily extend our pool of statistics by simply adding a new method and considering its result in the statistics ensemble.

In the following paragraphs, we provide a brief overview of all collected statistics with the respective amounts of individual statistics and the amount of collected instances. Note that the terms 'title' and 'content' in this list refer the article's heading and the article's text corpus respectively.

Named Entity Recognition

(37 statistics × 688 instances). We perform a named entity recognition on the content with 9 different entity tags and obtain the proportions of nameable entities and individual entity types with respect to the total word count. From these proportions, we collect the raw proportions over the whole content as well as aggregated differences between the first half and the second half of the content's paragraphs.

Paragraph Length Word Count

(3 statistics × 1002 instances). We count the number of words of the individual paragraphs and obtain the minimum, maximum, and average word per paragraph.

Part of Speech Tags

(24 statistics × 634 instances). We perform a Part of Speech (PoS) tagging with 12 different PoS tags and obtain the proportions of tag counts scaled by the number of words for the overall content as well as the differences between the first half and the second half of the content's paragraphs.

Sentence Word Count

(5 statistics × 940 instances). We count the number of words per sentence and obtain the quartiles, the mean, and the difference between mean and median.

Sentiment Analysis Score

(11 statistics × 858 instances). We perform a Sentiment Analysis on the content and obtain aggregated statistics among the sentiment score of the individual paragraphs like the min/mean/max, the variance, and the mean squared difference of the sentiment scores.

Stop Words Count Proportion

(9 statistics × 952 instances). We collect the proportion of the frequencies from the 8 most frequent stop words in the content scaled by the total number of words.

Title Content Commonality

(2 statistics × 646 instances). We count the common occurrences between words in the title and words in the content and scale them by the number of sentences in the content. In addition, we compare these proportions between the first and the second half of paragraphs by the means of a difference.

Word Frequencies Proportion

(7 statistics × 988 instances). We collect the frequencies of the 7 most frequently occurring words with respect to the total number of words.

Zipf's Law Deviation

(7 statistics × 952 instances). We obtain the deviation from the theoretical word counts from Zipf's Law by considering the difference in occurrences between the 7 (second-)most frequent words and their theoretic count from Zipf's Law. Note that the most frequent word from the content forms the reference for the theoretic values from Zipf's Law.

No Paragraph [*]

(48 statistics × 575 instances). We collect the statistics from Named Entity Recognition, Part of Speech Tags, Sentiment Analysis Score, and Title Content Commonality which incorporate the paragraphs in the same way but instead of using the paragraphs, we relate them to the sentences.

3.2 Deviation Scoring

With the set of collected statistics as described in Section 3.1, we can obtain their distributions by applying a kernel density estimation to all statistics individually. We call the collection of all individual distributions 'learned distribution', as it should represent the distribution of these statistics in genuine articles.

Based on the learned distribution, we construct a scoring function which gives a higher score to a sample within the distribution and a lower score to a sample outside from the distribution. In order to penalize differing sample much more than slightly deviating samples during scoring, we transform the learned densities according to the following transformation function:

$$f(x) = \log_2 \left(1 + \left(\frac{x}{\max(density)} \right)^{\frac{1}{4}} \right). \tag{1}$$

Furthermore, we apply moving average smoothing with a window size of 48 to our 1024-points-discrete score function to remove some inconsistencies that arise from the data and to broaden its range (i.e., accounting for unlearned small deviations). Figure 1 shows an example of such a scoring function. Note that the scoring function ranges from 0 to 1 but is scaled in this figure to properly match the data in the visualization.

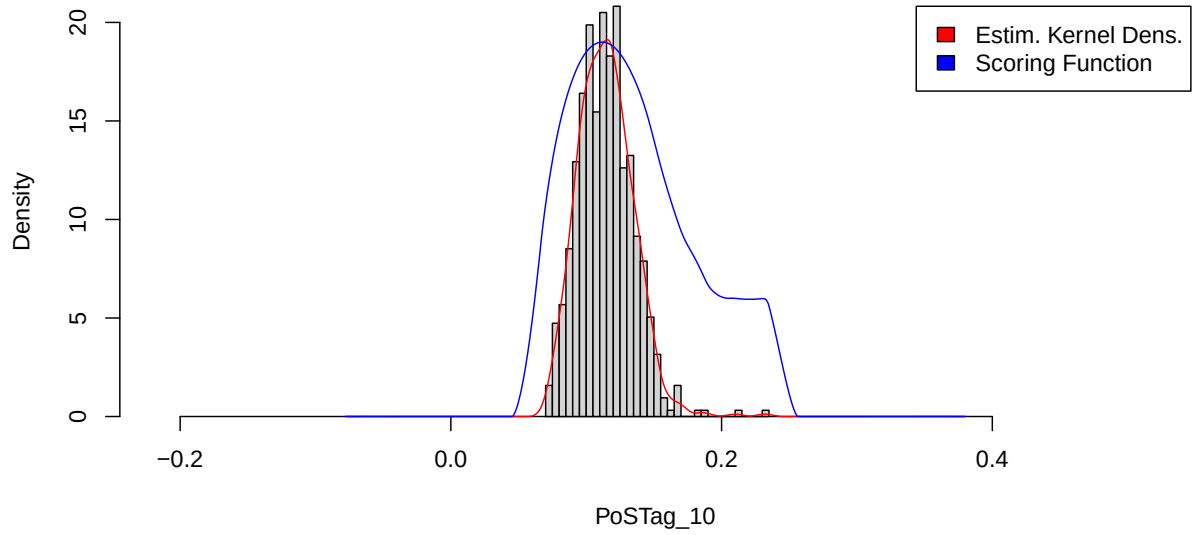

Figure 1: Histogram of the collected statistics 'PoSTag_10' with the estimated density and the respective scoring function.

For scoring an actual article, we obtain the very same statistics from the article under test and average the respective deviation scores from the scoring functions over all statistics. Thereby, we can compute the test scores for a set of 100 articles which results in a single value per article that indicates how likely the article is genuine. In our specific example with a given amount of 40 generated articles out of 100 articles, we classify the 40 article with the lowest score as 'generated'.

4 Experiments

4.1 Dataset Description

In order to obtain a proper dataset for training (i.e., learning the distributions of statistics), we crawl news articles with 50 to 100 sentences from various news websites. From these news articles, we then collect the statistics (like described in Section 3.1) in a collection with up to 1002 instances per statistic. The news websites considered for training contain FoxNews, Reuters, CNN, BBC, NewYorkTimes, NPR, and TheGuardian.

Since we do not have generated articles for classical evaluation at hand, we need to come up with an alternative approach. Basically, our classifier learns to discriminate real news articles from any other kind of documents based on various textual characteristics. With our belief that the generated articles differ from the real news articles in terms of the collected statistics, we can evaluate the performance of discriminating real news articles from other documents by testing against documents which are not news articles. For that purpose, we consider articles from ScienceDaily (SD), articles from PCMag (PCM), and documents from the Kaggle datasets MediumArticles (Md), NipsPapers (NP), WikiMoviePlots (WMP), and DataScientistJobDescriptions (DSJD). We then test the documents from these sources against folds from our training data when performing cross validation, and against articles with 30 to 200 sentences from the new source NBCNews (NBC) for the basic evaluation.

4.2 Setup

Using the datasets described in Section 4.1, we would like to evaluate our classification approach. Therefore, we set up two evaluation settings as described in the following paragraphs.

Cross-Validation against other Articles

We perform k-fold cross-validation with 17 folds on the training dataset. The left-out fold with 59 instances constitutes the real article portion of our test set which we test against other articles from NBC, SD, PCM, and Md. These other articles are assumed to be the 'generated' article portion. Note that we also test against NBC which are unseen news articles and, hence, theoretically belong to the 'real' test set. Finally, we compute and average the precision from the 17 cross-validation runs to obtain our performance metric.

Testing against other Documents

In addition to the cross-validation setting, we perform simple tests with 60 of the NBC articles as real test set against 40 randomly selected documents from Md, NP, WMP, and DSJD. In this scenario, we train our classifier using the entire training dataset. Just like before, we obtain our performance metric by computing the classification precision.

4.3 Results

In this section, we present the results from our two evaluation settings. In general, we perform an ablation study where we consider our complete ensemble as full model and compare it against the partial model considering only the statistics of 'Word Frequencies Proportion' (see Section 3.1) which seem to play an important rule for this type of classification. The results from the partial model are indicated in the plots by a preceding 'WF'. In addition to that comparison, we consider the baseline of random guessing to see whether our approach is useful at all. Note that most of the 'Named Entity Recognition' statistics are not considered in evaluation due to an implementation error (see Section 5).

Figure 2 shows the precision results from our cross-validation with other article sources. Obviously, we expect the results from NBC to be worse than random guessing since this dataset contains real news articles. With the thought that we learn the distribution of news articles, however, we expect the better results from non-news sources. As this is not the case in our evaluation and even the partial model stays around the random guessing baseline, news articles do not differ from other articles in our set of statistics. Anyhow, we still expect them to differ from other document types.

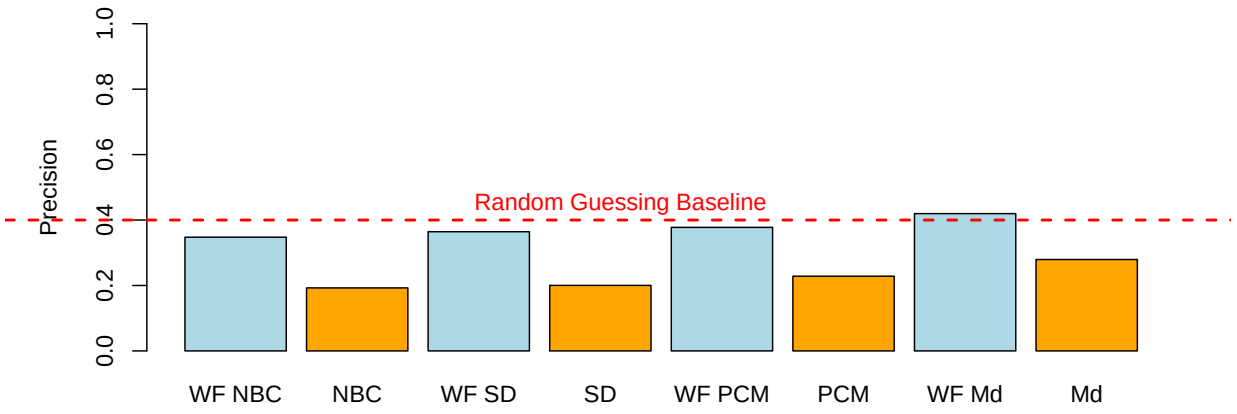

Figure 2: Precision Results from the Articles Cross-Validation Experiment

In Figure 3, we present the results from the evaluation with other document types. As we can observe, our classifier manages to detect the documents from NP, WMP, and DSJD almost perfectly. We included again the 'Md' articles from the previous experiment, which also show improvements in this case where we test against the new NBC articles. Comparing the results from the partial model to the full model, we can see that the full ensemble of statistics outperforms the partial model, even though there is not such a substantial difference as we would expect.

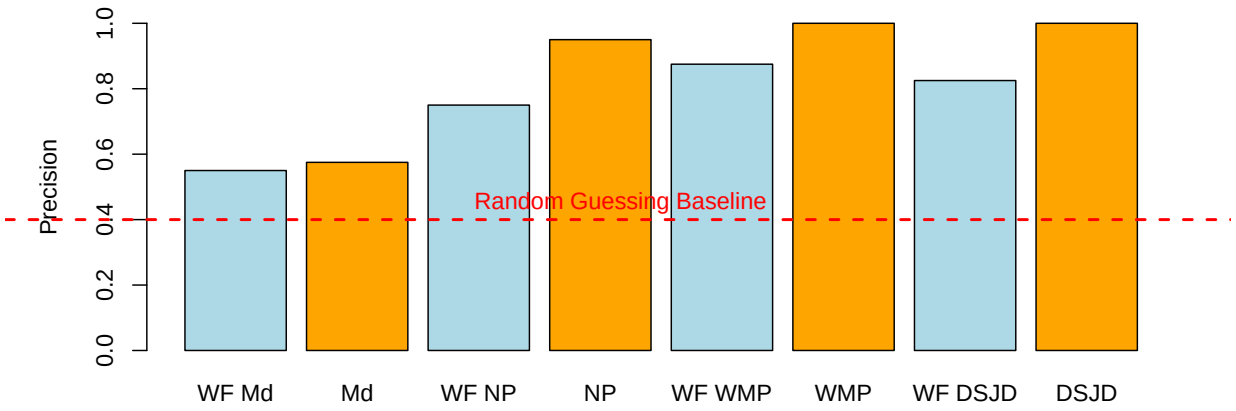

Figure 3: Precision Results from the Experiment with other Document Sources

5 Discussion and Conclusion

In our classification approach, we rely on some assumptions about the input documents. We generally assume that the documents are (news) articles written in English. Furthermore, we assume the documents to contain paragraphs and headline as it is usual for news articles. Unfortunately, the received test documents do not follow these assumptions. Especially the presence of paragraphs is of great importance for some of our statistics, as they are based on paragraphs. Because of these obstacles, we quickly re-implemented the statistics which rely on paragraphs and based them on sentences ('No Paragraph [*]' statistics in Section 3.1). With these fixes, we introduced an error which makes most of the 'Named Entity Recognition' statistics useless. Therefore, we had to exclude these particular statistics in our evaluations.

Other than these problems, we conclude from the evaluations that our system is capable of learning by which features articles are characterized. However, it cannot distinguish news articles from other articles. Therefore, we generally expect our classifier to detect the difference between genuine and generated articles if they differ in at least one of our 157 statistics that we consider.

References

[Atodiresei et al.(2018)] Costel-Sergiu Atodiresei, Alexandru Tănăselea, and Adrian Iftene. 2018. Identifying Fake News and Fake Users on Twitter. *Procedia Computer Science* 126 (2018), 451–461. `https://doi.org/10.1016/j.procs.2018.07.279` Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia.

[de Beer and Matthee(2020)] Dylan de Beer and Machdel C. Matthee. 2020. Approaches to Identify Fake News: A Systematic Literature Review. *Integrated Science in Digital Age 2020* 136 (2020), 13 – 22.

[Horne and Adali(2017)] Benjamin D. Horne and Sibel Adali. 2017. This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News. *ArXiv* abs/1703.09398 (2017).

[Ruchansky et al.(2017)] Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A Hybrid Deep Model for Fake News Detection. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (2017).