Comparing Improved TF-IDF Algorithms on Classification of Conspiratorial Content

Attila Balla
STUDENT NUMBER: 2064667

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

Supervisor: Dr. Michał Klincewicz
Second reader: Dr. Peter Hendrix

Tilburg University
School of Humanities & Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
June 2021

Acknowledgements

I would like to acknowledge the people who helped me during this research project. I would like to express my gratitude toward my supervisor, Michał Klincewicz for his patient support during the process. I also would like to thank Alfano and his colleagues and the previous students of the university who gathered data for this project. I am also thankful for the students from my cohort (Dimitris, Ivette, Neris & Thalia) for communication and support during the proposal writing and data collection, and to Noémi Farkas for proofreading and emotional support.

Table of contents

List of Figures

List of Tables

Comparing Improved TF-IDF Algorithms on Classification of Conspiratorial Content

Attila Balla

Abstract:

YouTube's recommendation system can make people become radicalized. Conspiracy theory videos are one of the problematic contents. The goal of this thesis is to find out to what extent improvements of the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm increase the efficiency of the vectorization to classify conspiratorial contents. Several previous studies solved this classification task with TF-IDF vectorization, but this is the first study to compare different alterations of TF-IDF. The original dataset of Alfano et al. (2020) *was expanded by the author and several other students. The used database contains 579 labelled transcripts of YouTube videos. It is unbalanced due to the low proportion of conspiratorial videos. The cleaned data were extracted with the textbook TF-IDF and with TF-IDF-CF and TF-IDF-ρ. The input data were trained with four classifiers with a proven record of solving text classification tasks and with low computational costs: Logistic Regression, Support Vector Machines, Naïve Bayes, Decision Tree. The main finding of this project that the modified TF-IDF vectorizations can improve the performance of the classification. TF-IDF-ρ outperformed the baseline and TF-IDF-CF and reached the overall best result with the SVM (macro average F1 score = 0.83). The highest rate of development from the baseline can be observed with the Naïve Bayes classifier (+41.07% and 39.29%). TF-IDF-ρ also produced significant improvement with the Logistic Regression classifier (+12.33%). The results are positive, but there are several limitations due to the charachteristics of the dataset and the method of the data collection.*

Keywords: TF-IDF, TF-IDF-ρ, TF-IDF-CF, Logistic Regression, Support Vector Machines, Naive Bayes, Decision Tree, YouTube, conspiracy theory

1. Introduction

The goal of this study is to compare the efficiency of different modifications of the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm for detecting conspiratorial content on transcripts of YouTube videos.

1.1 Context

YouTube is the most popular video-sharing website on the internet (Alexa., 2020). For many of their users, it is also an important source for getting news and knowledge about public topics (Pew Research Center, 2012). This means that the video-sharing website's recommendation system and their moderation practice have significant social relevance.

YouTube's recommendation system received several criticisms because of the claim of its radicalization pipeline (Tufekci, 2018). The critics state that the video-sharing site's recommendation system tends to suggest more and more extreme content to its viewers. Because more than 70% of the users' watch time is reached via the AI-driven recommendation, it is highly relevant to put this claim under scrutiny (Roose, 2019).

The video-sharing site's recommendation algorithm's main purpose is to maximise the users' view time on the website, which means that videos showing sensational or provocative content can be preferable for the algorithm (Faddoul, Chaslot, & Farid, 2020).

Riberio et al. (2019) gave the first quantitative evidence of the radicalization pipeline effect. They found that users of YouTube tend to migrate from milder to more extreme content. Their study also supports the allegation about the site's recommendation system's contribution to radicalization. Another recent article reviewing 23 studies and also came to the conclusion that the recommender is able to facilitate pathways to problematic videos (Yesilada & Lewandowsky, 2021).

Conspiracy theories help individuals to understand the world in a psychologically pleasant manner. It means that people are open to accepting information which can be fitted easily to their pre-existing beliefs and values (Krekó, 2015). This makes YouTube videos with conspiracy theories more likely to be preferred by users.

There are several previous pieces of research that built classification models for deciding whether a video transcript contains conspiratorial content. Feature extraction is a crucial part of the process, and TF-IDF vectorization is a common method for addressing this issue. Some relevant studies with TF-IDF will be introduced in the next chapter.

The TF-IDF vectorization is one of the most common and highly effective ways to the transformation of text data (Kim, Seo, Cho, & Kang, 2019), but it has several shortcomings as well (Guo & Yang, 2016). This led researchers to work on building improved algorithm. Some of these studies will be introduced in the *Related work* section. The novelty of this project is that there is no earlier study which aimed to improve the performance of conspiracy theory video detection by examining the different improvement of the original algorithm.

In order to achieve the goal of this study, two different improvements of the TF-IDF algorithm will be compared with the original vectorization method for detecting conspiratorial content on transcripts of YouTube videos. For the classification task, a dataset with a modest size (N=579) will be used. The raw data contains transcripts of YouTube videos with binary labelling (conspiratorial or not conspiratorial video).

Four classifiers will be used for the Natural Language Processing (NLP) task of this research project. These are Logistic Regression, Decision Tree, Naïve Bayes, and linear Support Vector Machine (SVM). These methods are one of the most popular and widely used methods text classification and they have similarly cheap algorithmic capacities (Pranckevičius & Marcinkevičius, 2017). These attributes make them suitable and attractive for comparing the different versions of the TF-IDF algorithm.

Citizens being misled by conspiracy theory videos are an obvious threat to the society. The societal and personal benefit of this research topic is that if we can improve the performance of the detection of conspiratorial videos and used this to moderate videos and make the recommendation system safer, we could prevent the radicalization of the users.

1.2 Research questions

The research question of this thesis project is the following:

> *RQ: To what extent do improvements of the TF-IDF algorithm increase the efficiency of the vectorization to classify conspiratorial contents?*

Breaking the main research question into two measurable pieces:

> *SQ1: To what extent does the TF-IDF-ρ algorithm increase the efficiency of TF-IDF vectorization to classify conspiratorial contents?*
>
> *SQ2: To what extent does the TF-IDF-CF algorithm increase the efficiency of TF-IDF vectorization to classify conspiratorial contents?*

1.3 Findings

The findings of this study suggest that for some, but not for all the classifiers the two improved TF-IDF algorithms can slightly improve the performance. Moreover, it seems to be that the linear SVM classifier with the TF-IDF-ρ vectorization is capable of reaching better performance than the other combination of the four classifications and the three vectorizations, while the two improved algorithms can make the greatest impact on the Decision Tree classifier. Further findings of this thesis will be presented later in the *Results* section.

2. Related work

2.1 Classification of conspiratorial videos

There were several studies in the past where the goal was to detect conspiracy theory videos based on their transcript. In this section, a short summary of these researches can be read. In accordance with the subject of this research, this section will especially focus on works that used TF-IDF vectorization in some part of their research process.

Alfano et al. (2020) examined different recommendation pathways on YouTube, with the endpoint being conspiracy theory videos. They manually rated transcripts of the videos based on 6 different categories. A labelling was made to decide whether the video is conspiratorial or not. Their original dataset contained 600 labelled transcripts. An online crawler was used to get the 100 most recommended YouTube videos based on the six different topics. They found that certain categories can lead significantly more frequently to conspiratorial content than other categories.

The thesis of van den Eijnden (2020) aimed to prove that conspiratorial content would be recognized to a greater degree by weighing the input data. First, an unbiased vector was created using the Delta TF-IDF method (Martineau & Finin, 2009). After that, the bias was applied by the Word2Vec (Rong, 2014) and GloVe (Pennington, Socher, & Manning, 2014) methods. A linear Support Vector Machine (SVM) technique was used for binary classification. The hypothesis was confirmed by the results as implementing an input bias into the conspiratorial vocabulary indeed improved the efficiency of the detection. It also showed the GloVe method outperforming Word2Vec approach.

Another recent thesis project paired text classification with sentiment analysis for the purpose of detecting conspiratorial videos (Tolgahan, 2020). The following classification techniques were executed: Logistic Regression, Support Vector Machine, Naïve Bayes and

Decision Tree. Weighting features with the help of sentiment analysis produced positive results with Logistic Regression being the most successful classifier.

These studies share some general limitations as they are working with very structured datasets (with big overlapping between their raw data), which may not be generalizable to other videos from YouTube or other sources, and they are also disadvantaged by the unbalanced characteristics of the data as the proportion of conspiratorial videos are low.

2.2 Term Frequency – Inverse Document Frequency

The extraction of the variables has a critical role in text classification tasks (Liang et al. 2017). Term Frequency – Inverse Document Frequency is a feature weighing scheme designed by Salton, Wong & Yang (1975). TF-IDF is an old method for vectorization, but its simplicity and effectiveness makes it popular as a starting point for more recent algorithms (Ramos, 2003)

The TF-IDF algorithm determines the relative frequency of a feature in a document. The term frequency (TF) part of the model calculates the term's frequency in a document. If it is higher, the TF-IDF will also be higher, because it means that it is a characteristic word for this document. The document frequency calculates how many documents does a feature appear in. If it is a greater value, it implies that the given word is not able to distinguish between the documents. For this reason, the document frequency is inverted (IDF), thus it lowers the TF-IDF weight.

While TF-IDF is an effective method, it also has several shortcomings. One of it is that it is blind to synonyms or any other connection between words (Berger & Lafferty, 1999). Furthermore, it categorizes words and their plurals as separate features, which decreases the word's TF-IDF weight by a small margin (Liang, Sun, Sun, & Gao, 2017). Another problem, which will be taken under scrutiny in this thesis, is that TF-IDF ignores the class labels of the training data, which can lead to inadequate weighting (Forman, 2008).

Forman also presents an illustrative demonstration of the problem: let's take a look at a binary classification problem with a dataset where a word *X* appears in 80% of the cases with a positive label and another feature *Y* occurs only in 3% of the positive cases and neither of them appear among the negative sample. This would mean that *X* would get a higher document frequency than *Y*, which will lead to a smaller IDF score, while it is obvious that *X* characterizes better the positive class.

This study assumes that addressing this shortcoming of the TF-IDF algorithm with implementing a modified algorithm can lead to the improvement of the performance on classifying conspiratorial videos. Several pieces of research have been made to improve the performance of the TF-IDF algorithm.

Two modifications of TF-IDF were selected for this thesis project (see their detailed presentation in the next two subsections). The aim was to choose improvement methods whose theoretical background makes it possible to expect them to improve the classification performance in the detection of conspiratorial videos. Another important aspect was to choose algorithms that are most likely to implement without error within the framework of a master's thesis.

2.3 TF-IDF-CF

Liu and Yang (2012) addressed the shortcomings of the TF-IDF algorithm by introducing a new parameter to represent the in-class characteristics. They named this parameter class frequency, which calculates the proportion of a feature appearance among documents from the same class. They also put a minor modification to the term frequency parameter by adding a logarithm function to it.

2.4 TF-IDF-ρ

Zhang and Ge (2019) offered another novelty to the standard textbook TF-IDF method called TF-IDF-ρ. Their primary focus was to create an algorithm that is effective with desensitized data. They added a new factor, discriminative strength to the formula, which gives higher weights to feature items with big discriminative strength, as they have a good ability to distinguish among the different types of texts. Based on the use of the special vocabulary of conspiratorial contents, it can be expected that the use of discriminative strength can be a useful improvement for solving this classification task.

3. Methods

In this section the TF-IDF vectorization method and the two chosen improvements of the original algorithm will be described. Furthermore, there will be a general explanation of the four classifiers.

3.1 Term Frequency-Inverse Document Frequency (TF-IDF)

The original equation of the Term Frequency-Inverse Document Frequency vectorization can be seen in Figure 1 (Salton, Wong, & Yang, 1975).

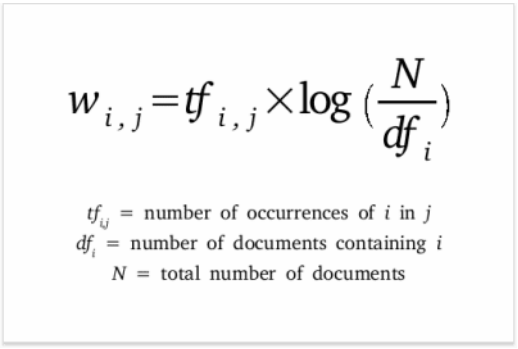

Figure 1. The original formula of the TF-IDF algorithm

As it can be read from the formula above, the TF-IDF matrix contains weights for the chosen word features in every document. The weight of a word *i* in the document *j* ($w_{i,j}$) is the product of the term frequency ($tf_{i,j}$) and the inverse document frequency ($idf_{i,j}$). The inverse document frequency is the logarithm of the total number of documents (*N*) divided by the number of documents containing word *i*.

It must be noted that in this study the above used IDF formula differs from the standard textbook notation (described above) as it is increasing the IDF value by one with the purpose of avoiding zero values. This is useful, because the additional terms that appears in all documents in a training set, will not be completely ignored by the algorithm.

3.2 TF-IDF-CF

The formula of the TF-IDF-CF algorithm (Liu & Yang, 2012) can be seen in Figure 2.

$$a_{ij} = \log(tf_{ij} + 1.0) * \log\left(\frac{N + 1.0}{n_j}\right) * \frac{n_{cij}}{N_{ci}}$$

Figure 2. The formula of the TF-IDF-CF algorithm

In the formula above, $a_{i,j}$ represents the TF-IDF-CF vector value for a word *j* in the document *i*. $tf_{i,j}$ represents the traditional term frequency value, but in this modified equation the final TF score is increased by 1 and then taking the logarithm of the sum. *N* represents the total number of documents. $n_{c,i,j}$ represents the number of documents where the term j emerges in the same category (*c*) that the document *i* belongs to. $N_{c,i}$ shows the amount of documents within *c* that the document *i* belongs to.

For the test data, it is not possible to calculate the class frequency, as the class of the item is unknown. Therefore, when calculating the TF-IDF matrix for the test data, the algorithm calculates the class frequency for every possible category (two in this thesis, as the label is a binary variable), and then uses the maximum value for the calculation.

3.3 TF-IDF-ρ

In the TF-IDF-ρ model of Zhang and Ge (2019) the calculation of the IDF score is modified. Their version can be seen at Figure 3.

$$\mathrm{TF-IDF-\rho} = \frac{n_{i,j}}{\sum_k n_{k,j}} * \log\left(\frac{N}{DF_i} * \frac{C}{C_i} + 1\right)$$

Figure 3. The formula of the TF-IDF-ρ algorithm

In the equation ρ is the discriminatory power of the feature and $\rho = c / c_i$, where c is the total number of categories in the corpus and c_i is the number of classes where the word appears. $n_{i,j}$ is the number of the word t_i appearing in the document d_j.

3.4 The four classifiers

Deng et al (2019) summarize different effective methods for text classification. Based on their paper the following classifiers will be used for this project: Logistic Regression, Decision Tree, Naïve Bayes, and Support Vector Machine.

4. Experimental Setup

In this section, the experimental setup of this research project will be described. First of all, the used dataset is described. Then the second step will be the explanation of the process of data cleaning. After that, the form of the word representation will be introduced. The visualization of the experimental pipeline can be seen in Figure 4. The models and evaluation metrics are discussed in the next chapter. The designed models to classify the conspiracy ratings are visible in the figure.

The used Python code is available on GitHub on the following link: https://github.com/balla-a/dss-thesis

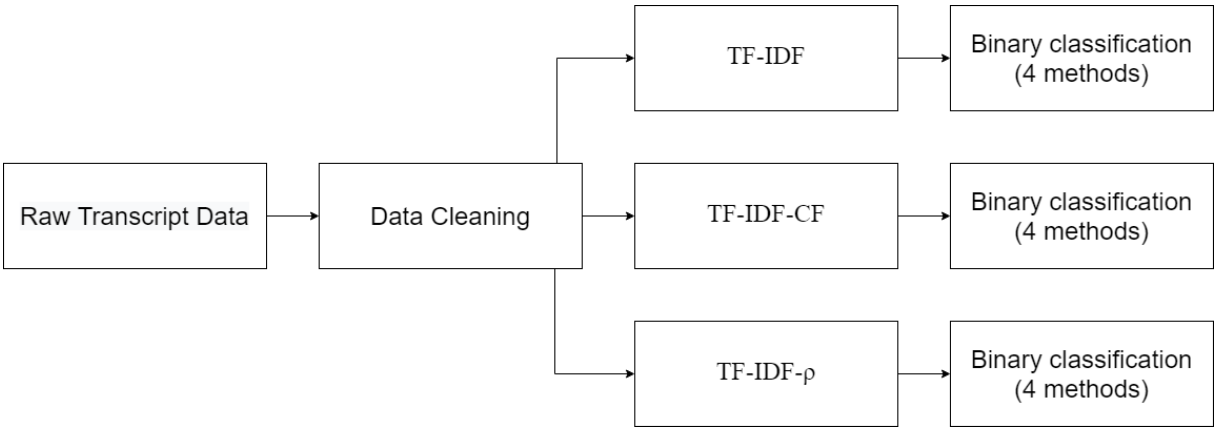

Figure 4. The project's experimental pipeline

4.1 Dataset description

The raw dataset contains text transcripts of labelled YouTube videos. The coding was made with a three-point scale, where the ranking indicates the conspiratorial claims of the videos:

(1) non-conspiratorial content
(2) conspiratorial with falsifiable arguments
(3) conspiratorial with unfalsifiable arguments

The complete dataset contains three already cleaned subsets from previous research projects and a new data collection was also performed by the author of this thesis and four other students. The sum of the three subsets from previous works has 482 unique transcripts. van den Eijnden (2020) gives a more detailed description of the collection, categorization, and cleaning process of these previous datasets.

For this project, 100 new video transcripts were gathered (20 by each student). The aim was to moderate the unbalanced characteristic of the available data by collecting conspiratorial videos (either with falsifiable or unfalsifiable type). The collection method was the following: on YouTube, searches were made with different conspiratorial-related keywords (e.g. “flat earth, “Bilderberg group”, “Obama antichrist”, “9/11 truth”, “covid-19 5G”). The keywords were chosen on a convenience basis. Only videos with available transcripts were collected ensuring that no duplication should be from previous datasets.

All of the transcripts were judged by three independent reviewers. Every judgment confirmed that the collected videos contained conspiratorial content. When there were disagreements about the falsifiability of the arguments, the final rating was by a majority decision. At the end of the process, three videos were removed, because they became unreachable on YouTube before their transcripts were extracted. This means 97 new cases were collected for this semester’s thesis projects. The inter-rater consistency was checked using Fleiss’ Kappa with a satisfactory outcome ($k = 0.781$, $z = 13.5$, $p < 0.0001$).

This means that the final dataset for this research consists of 579 non-empty transcripts without any duplication or missing labels. The new database is still unbalanced as 60.10% of the records are non-conspiratorial, while 20.21% of them are conspiratorial with falsifiable arguments and 19.69% of them are conspiratorial with unfalsifiable arguments. *Figure 5* shows the distribution of the three categories on a bar chart.

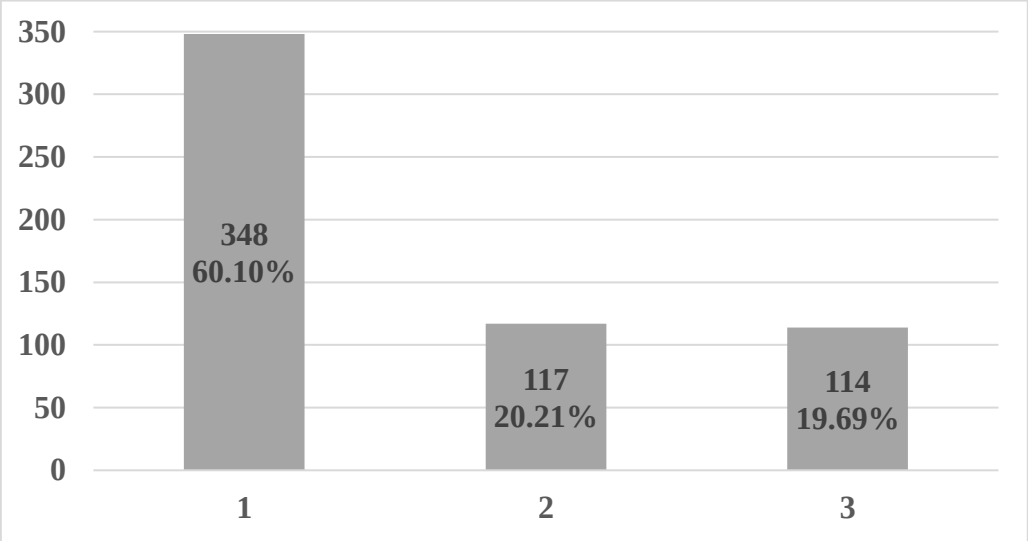

Figure 5. The distribution of the categories

4.2 Data cleaning

The aim of the data claiming for text data is to transform the items (documents containing words and sentences) into a feature vector, where each unique word represents a single variable (Kadhim, 2018). This process is called tokenization, and this is the first step of the pre-processing of the raw text data.

For this research project, the work of van den Eijnden (2020) stood as a sample during the data cleaning. The process contained various Natural Language Processing methods. The punctuations were removed and only the alphanumeric characters remained among the tokens. Every word in the data were lowercased.

Spacy's en_core_web_sm parser (Honnibal & Montani, 2017) was used for the data cleaning. At the end of the data cleaning process, the final dataset had a mean word count of 953 per transcript. The maximum word count in the database is 31 516 words. There are 24 931 unique words in the corpus.

4.3 Feature extraction

In order to reduce the number of words, the TF-IDF vectorizer function was used from the sci-kit package. This algorithm transforms the word counts into importance values in the given corpus. The TF-IDF vectors representing the biggest ratio of importance were selected. 2000 features with the most distinguishing value were used during the later steps of the process. The value 2000 was decided after some consideration which will be discussed in the *Hyperparameters* section of this thesis.

4.4 Relabelling of the dataset

It was mentioned previously that the database contains three categories:

(1) non-conspiratorial content
(2) conspiratorial with falsifiable arguments
(3) conspiratorial with unfalsifiable arguments

For this thesis project, all of the transcripts were relabelled the following way:

(0) non-conspiratorial content
(1) conspiratorial content

It means that the target variables with the value of 1 were transformed into 0, and the value of 2 and 3 was transformed into 1. Using multiclass labels would lead to a more difficult interpretation of the comparison of the different TF-IDF and relabelling also leads to more balanced database. However, with 39.90% of the data being from the minority group, the dataset still has a mild imbalanced nature.

4.5 Hyperparameters

In this section, the hyperparameter tuning is presented. After splitting the cleaned dataset into training, validation, and test sets, the hyperparameter tuning was carried out on the validation test. Based on the academic literature three hyperparameters were selected for this thesis project.

The number of the most relevant features were chosen based on their TF-IDF values as the first hyperparameter. It is a highly useful hyperparameter to tune because some words with very low frequencies can lead to adding a misleading factor into the categorization.

The number of feature values of 100, 200, 400, 700, 1000, 2000, 4000, 7000, and 10 000 have been tested (see Appendix A: Tuning of the maximum number of features). Taking into account that the maximum F1 scores for the different classifiers and the running time of the code, the feature number of 2000 was chosen as the best solution. Therefore, it will be used for further analysis.

The next hyperparameter is the C parameter of the Support Vector Machine classifier. The C parameter defines the number of samples inside the margin add to the comprehensive error. C = 0 would mean that the sample cases which are inside the margins are not castigated. On the other hand, an infinite value of the C parameter would lead to extremely hard margins.

The hyperparameter tuning was made with C = 1, 3, 5, and 10. The F1 score was calculated on the validation set. Based on the results (see Table 1) C = 5 was chosen as the best performing parameter, so this will be used for further MSV classification reports.

F1 scores (macro avg)			
Parameter C	**TF-IDF**	**TF-IDF-ρ**	**TF-IDF-CF**
1	.78	.82	.78
3	.80	.83	.78
5	.81	.83	.79
10	.77	.82	.79

Table 1. The F1 scores for the hyperparameter tuning of the C value

The last hyperparameter is the maximum depth of the Decision Tree classifier. The minimum value of it is 1, which means the nodes are only expanded for one step, while if the value is set to None, the nodes are being expanded until all leaves are pure or contain less than the minimum sample split size. The values of 1, 3, 5, 9, and None were tested on the validation set with the help of the F1 score. Based on the results max_depth = 1 was chosen as the best hyperparameter (see Table 1Table 2).

F1 score (macro avg)			
max_depth	**TF-IDF**	**TF-IDF-ρ**	**TF-IDF-CF**
1	.76	.77	.75
3	.73	.72	.74
5	.77	.73	.69
9	.74	.73	.71
None	.74	.73	.69

Table 2. The F1 scores for the hyperparameter tuning of maximum depth

For the two remaining classifiers of this project (Linear Regression and Naïve Bayes) hyperparameter tuning was not conducted and they were addressed with their default values.

4.6 Evaluation metrics

Handelman et al. (2019) summarize the most common evaluation methods for machine learning problems. Based on their work the use F1 score is planned to be applied. The F1 score is the harmonic mean of precision and recall.

$$F_1\text{-}score$$

$$\frac{1}{\frac{1}{2}\left(\frac{1}{\text{recall}} + \frac{1}{\text{precision}}\right)}$$

Figure 6. The formula of the F1 score

The formula of the F1 score can be seen in Figure 6. Recall or sensitivity is the true positive rate, so the correctly classified conspiratorial contents. Precision or positive predictive value is the proportion of true positive values among all the values labelled as positive.

It is worth mentioning that the data is slightly unbalanced because there are more non-conspiratorial (60.10%) videos than conspiratorial videos (39.90%) in the database. This means that accuracy is not meaningful for our case, even if it is also a widely used evaluation metric for classification tasks.

4.7 Software and packages

The code for this thesis project was made by Python 3.8 programming language. The coding process took place in the Jupyter Notebook (Version 1.1.4) server-client web application. Several Python libraries and packages were used for the code. The complete list of them is the following:

- csv
- en_core_web_sm (Honnibal & Montani, 2017)
- math
- NumPy (Harris et al., 2020)
- os
- Pandas
- pickle
- scikit-learn
- SciPy (Virtanen et al., 2020)

Some of the charts and tables were made with Microsoft Excel (Version 2008). The whole project has been hosted on a PC with a 3.5 GHz Intel i7 Core.

5. Results

In this section, the results of the classifications will be presented. First, it will be analysed whether the two modifications of the TF-IDF vector succeed to improve our performance for this binary classification task. Also, the comparison of the four classifiers will be described. Secondly, it will be examined that to what extent does the two modified TF-IDF algorithms improve the classification performance.

F1 score was chosen as the primary evaluation metric for this research project, so the performance will be judged based on this metric. The full classification reports with precision and recall scores can be seen in Appendix B: Classification reports. The results are shown for the 12 classifications (3 vectorizations * 4 classifiers) in Table 3.

	F1 score (macro avg)			
	Log. Reg.	**SVM**	**Naïve B.**	**Tree**
TF-IDF (baseline)	.73	.81	.56	.76
TF-IDF-ρ	.82	.83	.79	.77
TF-IDF-CF	.74	.79	.78	.75

Table 3. The F1 scores of the classifications.

Overall, the TF-IDF-ρ vectorization with the linear SVM classifier yielded the best result (F1 score = .83). The TF-IDF-ρ vectorization reached higher F1 scores in all of the four classifications than the baseline and also proved better than the TF-IDF-CF method. The TF-IDF-CF performs better than the baseline in two cases (Logistic Regression and Naïve Bayes), while underperforms with the other two classifiers.

The linear SVM classifier outperformed the three other methods with every input vector. SVM's performance on the baseline also outperformed almost all of the improved TF-IDF's performance with the other three technique, except the Logistic Regression with the TF-IDF-ρ.

If the rate of improvement is taken under scrutiny, it can be seen that in the 6 cases where an improved TF-IDF was able to perform better than the original TF-IDF, there are only three cases where the modification was able to perform substantially better results (see Table 4. *The F1 score improvement using the modified TF-IDF vectors*). The greatest improvement was reached by the TF-IDF-ρ algorithm with the Naïve Bayes classifier (+41.07%), while the TF-IDF-CF also achieved a huge progress (+39.29%). Using Logistic Regression, the TF-IDF-

ρ managed to improve the result with a notable margin (+12.33%). It is also worth mentioning that this is the only classifier, where the disadvantage of the TF-IDF-CF is more than 2.5% against the TF-IDF-ρ (10.96%).

	F1 score improvement (baseline = 100%)			
	Log reg	**SVM**	**Naïve B.**	**Tree**
TF-IDF-ρ	+12.33%	+2.47%	+41.07%	+1.32%
TF-IDF-CF	+1.37%	-2.47%	+39.29%	-1.32%

Table 4. The F1 score improvement using the modified TF-IDF vectors

6. Discussion

The research question of this thesis was that to what extent do improvements of the TF-IDF algorithm increase the efficiency of the vectorization to classify conspiratorial contents.

TF-IDF-ρ outperformed both the baseline and the TF-IDF-CF technique. This method uses a discriminatory power value ρ which doesn't calculate with the proportion of a feature word in a set of documents within the same class. It means that even if there is one feature word in a large corpus of the training set, it will affect the discriminatory power of the feature the same way if they are present with a high frequency in another category. This suggests that this method's performance could drop if the used database became larger because there would be a growing possibility that even a feature word with an obvious connection to conspiratorial will occur accidentally in other categories. Therefore, examining the performance of the TF-IDF-ρ in databases with the same characteristic, but with altering size would be a possible interesting path for future researches.

The TF-IDF-CF weighting method produced its best and only convincing performance with the Naïve Bayes classifier. The previous study of the proposers of the TF-IDF-CF algorithm also has their best performance compared to the TF-IDF baseline with the Naïve Bayes (Liu & Yang, 2012), so this thesis project confirms this previous result. However, it should be mentioned that they used a more balanced dataset and evaluated their result with the accuracy metric. Their model also shows significant progress with the MSV classifier, while this study was not able to reproduce this finding. But their work didn't report the parameters of the SVM classifier, which makes it harder to compare the results.

The high level of heterogeneity among the topics of the conspiratorial videos can also explain the relatively modest results of the TF-IDF-CF algorithm. The multiple data gathering didn't focus on a single kind of conspiracy theory (e.g. the flat earth) but collected all kinds of

conspiratorial contents. While there can be a general vocabulary which is characteristic for any many types of conspiracy theory contents, different topics can have different language. The TF-IDF-CF calculates class frequency, so a feature that only has high distinctiveness for a given conspiracy theory would score low on the class frequency weight. It can be mentioned that TF-IDF-ρ doesn't have this shortcoming as it needs only a single appearance in a category to be calculated in the discriminatory power.

Using multiclass classification with different labels representing different conspiracy theories would allow the specific vocabulary of a theory to have a high value of the class frequency. This approach would be an interesting path for a future study, but it would require a database with much more conspiratorial transcripts.

It is also important to remember that there are many more modifications to the original TF-IDF algorithm. While these two variations were chosen because the ideas behind them suggested that they can be relevant in the case of conspiratorial videos and were computationally cheap and easy to interpret. But it shouldn't be ignored that there is a possibility that other improvements can also be useful for developing the performance of our categorization. Based on the positive results of this research, using another improved TF-IDF method for the classification task of conspiratorial videos looks to be a worthy path for future researches.

The linear MSV classifier proved to be the best for this classification task. This is not surprising as this method is known for the good handling of datasets with a small amount of data and imbalanced proportion between the classes. For this project, the aim was to choose popular and widely used classification methods with a low level of calculation resources. But previous studies reported that Neural Networks would be the best choice for TF-IDF vectorizations (Lam & Lee, 1999). It suggests that in the future it would be useful to examine the possibilities of improvement by modified TF-IDF algorithms with Neural Network classifiers.

Finally, it can cause a methodological problem that YouTube tends to remove videos with claimed conspiratorial contents (Thompson, 2020). which is appropriate at the societal level, but makes it harder to collect data for researches, and building a detection model with high ecological validity.

7. Conclusion

The main focus of this study was to improve the performance of detection of conspiratorial YouTube videos using their transcripts. The novelty of this project compared to previous researches addressing this issue was that different alterations of the widely used TF-IDF algorithm were implemented and their performance was examined.

The research question of this thesis project was to find out that to what extent do improvements of the TF-IDF algorithm increase the efficiency of the vectorization to classify conspiratorial contents. Two previously designed improvements of the textbook algorithm were chosen for this project: TF-IDF-CF and TF-IDF-ρ. Therefore, the two sub-questions of this thesis were that to what extent does the TF-IDF-ρ and the TF-IDF-CF algorithm increase the efficiency compared to the baseline. This was explored with four common classifiers.

The findings show that for this kind of dataset, the TF-IDF-ρ proved to be successful in improving the performance of the TF-IDF algorithm. For some classifiers, the TF-IDF-CF also showed good results. The biggest improvement from the baseline was achieved with the Naïve Bayes classifier, while the TF-IDF-ρ also showed good performance with Logistic Regression. The baseline method cannot outperform scientifically any of the modifications.

Detecting conspiratorial contents can prevent the users of YouTube from radicalizing content, which gave societal relevance for this and for future studies in this area. The possible path of future researches was suggested. They contained implementing further improvements of the TF-IDF algorithm and using a database with different characteristics and also experimenting with effective but computationally more expensive methods, like the Neural Network classifiers.

References

Alexa. (2020, 10 22). "It is just a flu": Assessing the Effect of Watch History on YouTube's Pseudoscientific Video Recommendations. *"It is just a flu": Assessing the Effect of Watch History on YouTube's Pseudoscientific Video Recommendations*.

Alfano, M., Fard, A. E., Carter, J. A., Clutton, P., & Klein, C. (2020). Technologically scaffolded atypical cognition: The case of YouTube's recommender system. *Synthese*, 1–24.

Berger, A. L., & Lafferty, J. D. (1999). Information Retrieval as Statistical Translation. In F. C. Gey, M. A. Hearst, & R. M. Tong (Ed.), *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA* (pp. 222–229). ACM. doi:10.1145/312624.312681

Deng, X., Li, Y., Weng, J., & Zhang, J. (2019). Feature selection for text classification: A review. *Multimedia Tools and Applications, 78*, 3797–3816.

Faddoul, M., Chaslot, G., & Farid, H. (2020, 3 6). A Longitudinal Analysis of YouTube's Promotion of Conspiracy Videos.

Forman, G. (2008). BNS feature scaling: an improved representation over tf-idf for svm text classification. *Proceedings of the 17th ACM conference on Information and knowledge management*, (pp. 263–270).

Guo, A., & Yang, T. (2016). Research and improvement of feature words weight based on TFIDF algorithm. *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, (pp. 415–419).

Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., . . . Asadi, H. (2019). Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods. *American Journal of Roentgenology, 212*, 38–43.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., . . . Oliphant, T. E. (2020). Array programming with NumPy. *Nature, 585*, 357–362. doi:10.1038/s41586-020-2649-2

Honnibal, M., & Montani, I. (2017). Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *Unpublished software application. https://spacy. io*.

Kadhim, A. I. (2018). An Evaluation of Preprocessing Techniques for Text Classification. *International Journal of Computer Science and Information Security, 16*.

Kim, D., Seo, D., Cho, S., & Kang, P. (2019). Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *Information Sciences, 477*, 15–29.

Krekó, P. (2015). Conspiracy theory as collective motivated cognition. *The psychology of conspiracy*, 60–67.

Lam, S. L., & Lee, D. L. (1999). Feature reduction for neural network based text categorization. *Proceedings. 6th international conference on advanced systems for advanced applications*, (pp. 195–202).

Liang, H., Sun, X., Sun, Y., & Gao, Y. (2017). Text feature extraction based on deep learning: a review. *EURASIP journal on wireless communications and networking, 2017*, 1–12.

Liu, M., & Yang, J. (2012). An improvement of TFIDF weighting in text categorization. *International proceedings of computer science and information technology, 47*, 44–47.

Martineau, J., & Finin, T. (2009). Delta tfidf: An improved feature space for sentiment analysis. *Proceedings of the International AAAI Conference on Web and Social Media*, *3.*

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, (pp. 1532–1543).

Pew Research Center. (2012). YouTube & News - A New Kind of Visual News. *YouTube & News - A New Kind of Visual News*.

Pranckevičius, T., & Marcinkevičius, V. (2017). Comparison of naive bayes, random forest, decision tree, support vector machines, and logistic regression classifiers for text reviews classification. *Baltic Journal of Modern Computing, 5*, 221.

Ramos, J. (2003). Using tf-idf to determine word relevance in document queries. *Proceedings of the first instructional conference on machine learning*, *242*, pp. 29–48.

Ribeiro, M. H., Ottoni, R., West, R., Almeida, V. A., & Meira, W. (2019, 8 22). Auditing Radicalization Pathways on YouTube.

Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*.

Roose, K. (2019). The making of a YouTube radical. *The New York Times, 8*.

Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM, 18*, 613–620.

Thompson, C. (2020). YouTube's Plot to Silence Conspiracy Theories. *Wired, 10*.

Tolgahan, A. (2020). *Classification of Conspiratorial Content on YouTube - Measuring Influence of Sentiment Weighting on Classification Performance.* Master's thesis, Tilburg University.

Tufekci, Z. (2018). YouTube, the great radicalizer.

van den Eijnden, R. (2020). *A keywords-base approach to conspiracy video classification.* Master's thesis, Tilburg University.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., . . . Contributors, S. 1. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods, 17*, 261–272. doi:10.1038/s41592-019-0686-2

Yesilada, M., & Lewandowsky, S. (2021). A systematic review: The YouTube recommender system and pathways to problematic content.

Zhang, T., & Ge, S. S. (2019). An improved TF-IDF algorithm based on class discriminative strength for text categorization on desensitized data. *Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence*, (pp. 39–44).

Appendix A: Tuning of the maximum number of features

	F1 score (macro avg)												
	TF-IDF				TF-IDF-ρ				TF-IDF-CF				
Max. features	**Log reg**	**SVM**	**Naive B.**	**Tree**	**Log reg**	**SVM**	**Naive B.**	**Tree**	**Log reg**	**SVM**	**Naive B.**	**Tree**	**Avg.**
100	.79	.75	.74	.67	.78	.82	.74	.70	.72	.79	.75	.67	**.74**
200	.80	.74	.75	.62	.79	.77	.75	.57	.72	.75	.75	.68	**.72**
400	.78	.76	.74	.67	.81	.81	.74	.67	.71	.77	.71	.65	**.73**
700	.78	.76	.75	.68	.82	.82	.75	.68	.71	.78	.68	.67	**.74**
1000	.78	.74	.77	.76	.83	.82	.77	.71	.70	.78	.67	.60	**.74**
2000	.78	.75	.79	.67	.83	.82	.75	.72	.76	.78	.74	.68	**.75**
4000	.78	.75	.75	.70	.83	.82	.75	.71	.76	.79	.74	.69	**.75**
7000	.78	.74	.74	.68	.83	.81	.73	.71	.77	.75	.74	.71	**.75**
10000	.78	.73	.69	.69	.83	.81	.68	.69	.77	.75	.71	.69	**.74**
Average	**.78**	**.75**	**.75**	**.68**	**.82**	**.81**	**.74**	**.68**	**.74**	**.77**	**.72**	**.67**	

Table 5. The F1 scores for different number of maximum features

Appendix B: Classification reports

	precision	recall	f1-score	support
class 0	0.75	0.88	0.81	104
class 1	0.75	0.57	0.65	70
accuracy			0.75	174
macro avg	0.75	0.72	0.73	174
weighted avg	0.75	0.75	0.75	174

Table 6. The classification report of the baseline with the Linear Regression

	precision	recall	f1-score	support
class 0	0.88	0.83	0.85	104
class 1	0.76	0.83	0.79	70
accuracy			0.83	174
macro avg	0.82	0.83	0.82	174
weighted avg	0.83	0.83	0.83	174

Table 7. The classification report of the TF-IDF-ρ with the Linear Regression

	precision	recall	f1-score	support
class 0	0.88	0.83	0.85	104
class 1	0.76	0.83	0.79	70
accuracy			0.83	174
macro avg	0.82	0.83	0.82	174
weighted avg	0.83	0.83	0.83	174

Table 8. The classification report of the TF-IDF-CF with the Linear Regression

	precision	recall	f1-score	support
class 0	0.84	0.86	0.85	104
class 1	0.78	0.76	0.77	70
accuracy			0.82	174
macro avg	0.81	0.81	0.81	174
weighted avg	0.82	0.82	0.82	174

Table 9. The classification report of the baseline with the SVM

	precision	recall	f1-score	support
class 0	0.89	0.82	0.85	104
class 1	0.76	0.86	0.81	70
accuracy			0.83	174
macro avg	0.83	0.84	0.83	174
weighted avg	0.84	0.83	0.83	174

Table 10. The classification report of the TF-IDF-ρ with the SVM

	precision	recall	f1-score	support
class 0	0.90	0.73	0.81	104
class 1	0.69	0.89	0.78	70
accuracy			0.79	174
macro avg	0.80	0.81	0.79	174
weighted avg	0.82	0.79	0.80	174

Table 11. The classification report of the TF-IDF-CF with the SVM

	precision	recall	f1-score	support
class 0	0.97	0.31	0.47	104
class 1	0.49	0.99	0.65	70
accuracy			0.58	174
macro avg	0.73	0.65	0.56	174
weighted avg	0.78	0.58	0.54	174

Table 12. The classification report of the baseline with the SVM

	precision	recall	f1-score	support
class 0	0.87	0.77	0.82	104
class 1	0.71	0.83	0.76	70
accuracy			0.79	174
macro avg	0.79	0.80	0.79	174
weighted avg	0.80	0.79	0.79	174

Table 13. The classification report of the TF-IDF-ρ with the Naïve Bayes

	precision	recall	f1-score	support
class 0	0.91	0.70	0.79	104
class 1	0.67	0.90	0.77	70
accuracy			0.78	174
macro avg	0.79	0.80	0.78	174
weighted avg	0.82	0.78	0.78	174

Table 14. Table The classification report of the TF-IDF-CF with the Naïve Bayes

	precision	recall	f1-score	support
class 0	0.77	0.91	0.83	104
class 1	0.82	0.59	0.68	70
accuracy			0.78	174
macro avg	0.79	0.75	0.76	174
weighted avg	0.79	0.78	0.77	174

Table 15. The classification report of the baseline with the Decision Tree

	precision	recall	f1-score	support
class 0	0.77	0.91	0.84	104
class 1	0.82	0.60	0.69	70
accuracy			0.79	174
macro avg	0.80	0.76	0.77	174
weighted avg	0.79	0.79	0.78	174

Table 16. The classification report of the TF-IDF-ρ with the Decision Tree

	precision	recall	f1-score	support
class 0	0.76	0.91	0.83	104
class 1	0.82	0.57	0.67	70
accuracy			0.78	174
macro avg	0.79	0.74	0.75	174
weighted avg	0.78	0.78	0.77	174

Table 17. The classification report of the TF-IDF-CF with the Decision Tree