A Keyword-based Approach to Conspiracy Video Classification

Raf van den Eijnden
STUDENT NUMBER: 983572

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

Thesis committee:

Michal Klincewicz
Wendy Powell

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
June 2020

Acknowledgements

I want to express my gratitude towards Mark Alfano, Colin Klein, Adam Carter & Amir Ebrahimi Fard for lending us access to their wonderful YouTube-dataset. Also towards Michal Klincewicz for excellent supervision in difficult times. Special thanks goes out to Siebe Albers and Tolga Akyazi for thorough communication and support in coding and writing this thesis, and to Eden den Drijver for thorough proofreading and emotional support.

A Keyword-Based Approach to Conspiracy Video Classification

Raf van den Eijnden

Abstract
YouTube has recently gathered criticism, as it would be a catalyst for the spread of conspiracy theories and content that leads to radicalization among users. By maximizing view time the algorithm overlooks certain borderline inflammatory content, YouTubes algorithm leaves room for improvement and accurate detection of conspiratorial content becomes evermore important. This research aims to not only build a classification pipeline towards recognizing conspiratorial content, but also enrich a classification pipeline by weighing the input data towards the use of keyword vocabularies reminiscent of conspiracy theories. A new method is proposed in which the combination of two keyword extraction techniques are combined with two semantic similarity measures to bias TF-IDF input data, thus aiming to improve a Support Vector Machine classifier as opposed to an unweighted baseline. The keyword extraction methods accurately depicted key terms of conspiratorial content when compared to literature regarding conspiracies, showing some validity in this method. Optimal classification results were obtained when combining a Most-Distinguishing Features approach with the GloVe algorithm to calculate semantic similarity between our keywords and our document word vectors (F1 = 0.74), showing that this keyword methodology can be used to further improve classification of conspiratorial content.

1. **Introduction**

Quite some criticism has recently befallen the algorithm that is running YouTube's recommendation system, as multiple experts on the YouTube recommendation algorithm discuss its tendency to show more and more extreme content, fake news, and conspiracy videos (Ledwich & Zaitsev, 2019; Ribeiro et al., 2019). Former YouTube algorithm developer Guillaume Chaslot has recently expressed concern regarding the nature of the recommendation algorithm. Supposedly, the algorithm is set to maximize user view time using recommendation of videos that are of interest to users and overlooks some important factors and considerations in recommending content. This, in turn, would lead to a bias towards sensational content, conspiracy videos, and factually inaccurate content (Faddoul, Chaslot & Farid, 2020). Social media have earlier been proven to have a catalytic effect on the adherence of conspiracy theories and the emergence of online platforms has led to a wider spread of alternate beliefs (Shin et al., 2018; Samory & Mitra, 2018; Chaslot, Faddoul & Farid, 2020). The concerns raised by Chaslot sparked interest in the scientific community and was heavily brought to discussion in a paper by Ledwich & Zaitsev (2019). After conducting a detailed analysis of video recommendations received by various political YouTube channels, this study suggests that YouTube actually favors mainstream media outlets instead of independent inflammatory channels as suggested by Chaslot. Other studies regarding radicalization, such as a study by Ribeiro et al. (2019), support Chaslot's

claims by finding that mild right-wing YouTube users are often easily sent towards more extreme right-wing and conspiratorial content.

As the largest operating video platform today (Tufekci, 2019), distinguishing in conspiratorial and extremist content becomes evermore important, as the platform plays a crucial role in formation of opinion and behavior. On a societal level a distinction between various types of content might prove useful, as our largest video platform might be a catalyst for extremism and conspiratorial views through its recommendation system, since 70% of all watched videos is recommended content (Solsman, 2018). In August 2019, the FBI has even introduced conspiracy theories as a domestic terrorist threat, as it fuels violent events and can provoke those with existing extremist and conspiratorial beliefs (Faddoul, Chaslot & Farid, 2020). Even though YouTube has announced that their efforts to reduce these harmful recommendations in January 2019 has led to a reduction in view time from these recommendations, new studies suggest that these effects had some effect, however view time seems to be increasing on conspiratorial content once more (Faddoul, Chaslot & Farid, 2020). Having any improvement in an algorithm that is able to detect conspiratorial content therefor will prove invaluable in preventing exposure to unfounded conspiratorial views and misinformation. Therefore, this study aims to not only build a classifier that can distinguish between conspiratorial or non-conspiratorial content, but also test a keyword-based method to biasing a classifier towards conspiratorial content.

Conspiracy theory researcher Cass Sunstein mentioned in his 2008 paper 'Conspiracy Theories' that these theories often have as distinctive features that they consist of false beliefs, that they generally attribute extraordinary powers to certain agents but most of all that they are characterized by their self-sealing mechanisms. Another important feature of conspiracy theories was captured by Karl Popper, who described how the appeal of these theories lies in the attribution of otherwise inexplicable events to intentional action. As a more recent paper by Samory & Mitra (2018) states: key elements of a conspiracy theory consist of agents, actions, and their targets. Samory & Mitra suggested that the language used in conspiracy theories is different than language used in various other theories and statements and that these theories often have a distinctive vocabulary. They developed the notion of a narrative-motif being present in the language behind these conspiracy theories. Klein et al. (2019) debated that interactions between users of conspiratorial content can be analyzed on a linguistic level to distinguish between conspiratorial or non-conspiratorial content. Enough theories support the idea that a different vocabulary is in place in conspiracy theories and this linguistic difference can be of great interest in developing a classification algorithm. Taking a linguistic approach can provide insight in both the nature and language of conspiracy theories, as well as insight in building an efficient and accurate text classification pipeline.

Many Natural Language Processing (NLP) methods have been used to make semantical distinctions and support text classification pipelines. A study by Chung & Pennebaker (2008) found that language used in self-description by students led to distinguishing between seven psychologically meaningful personality types after using a NLP-based factor analysis. Another paper by Pestian et al. (2010) successfully distinguished between suicide notes and notes of healthy controls by using a semantically based NLP approach. More closely related to conspiracy videos, a 2008 paper by Mallouf & Mullen found that different political stances can be extracted from the use of language and vocabulary in their discourse, and found that NLP methods can be used to cluster important utterances and words related to certain political beliefs and stances. These studies amount to the idea that good feature extraction and selection, along with a semantically based approach can be a powerful tool to distinguish between two semantical categories by means of analyzing the language and vocabulary used in these categories. The main goal of this study therefore is, apart from building a conspiracy video classification pipeline, to establish language patterns in conspiratorial content in the form of a keyword vocabulary, and see if this vocabulary of keywords can be used to bias the used classifier towards use of specific words that are found in conspiratorial content. I hypothesize that implementing a bias

based on a vocabulary specific to conspiratorial content can make classifiers more precise and recall videos of a conspiratorial more efficiently. This study will examine two methods of constructing a keyword vocabulary specific for conspiratorial content and combine those with two biasing methods that implement a semantic similarity measure to those keyword vocabularies in a classification pipeline. This study aims to explore how enriching TF-IDF word vectors with semantical information, based on keyword vocabularies, can influence the performance of a conspiratorial text classifier, and thus will answer the question: "Can a conspiratorial text classifier be enhanced by using a keyword extraction method to implement a semantic bias in the dataset?".

2. Related Work

In this section an overview is provided with earlier studies regarding YouTubes recommendations, conspiracy theories and their language components, and some similar Natural Language Processing approaches are highlighted.

2.1 YouTubes Recommendation System and the Spread of Conspiratorial Content

Present day social media allows for an open exchange of ideas and beliefs. Many platforms, such as Facebook and YouTube, allow users to upload and share all types of content. In this environment all types of content, even extremist and conspiratorial videos, can be shared and watched by users. This in itself could pose a problem, but another issue is of greater concern. Many researchers and legislators have recently expressed their concern about YouTube actively recommending misinforming or conspiratorial content. As Zeynep Tufekci pointed out in her New York Times article "YouTube: The Great Radicalizer", YouTube has a tendency to point you towards more "hardcore" content. Videos about vegetarianism will lead to videos about veganism, videos about jogging to videos about ultramarathons, and videos about Hillary Clinton to more extreme leftist content. Recently, we received another example of how YouTube can catalyze the adherence to false beliefs, as during the recent COVID-19 pandemic critics noticed the emergence of many conspiracy beliefs about the pandemic being a hoax, a bioweapon, or a result of 5G technology (Ahmed et al., 2020; Imhoff & Lamberty, 2020). The "filter bubble" effect can lead to people remaining in a bubble of harmful or inflammatory content which can lead to self-radicalization (Alfano, Carter & Cheong, 2018). It seems that people can easily consume sources of misinformation and conspiracy theories when browsing the YouTube video pool, as they have a tendency to find information in different sources when facing circumstances that challenge the status quo (Shin et al., 2018). Recommendations from YouTube play a big role in the exploration of these alternate sources of information, as they account to a large part of viewed content on YouTube. YouTube has a tendency to show us more extreme content, as this maximizes the view time on videos in general, showing there might be more appeal in inflammatory and sensationalist content. This is also seen in a study by Song & Gruzd (2017), who showed that watching a few anti-vaxx videos can quickly lead to recommendations of similar videos. YouTubes ability to personalize their massive library of videos is crucial for the service and critical for its livelihood (Solsman, 2018), however some important harmful videos are overlooked in the process. YouTubes Chief Product Officer has also told us that machine learning algorithms and artificial intelligence is at the backbone of this recommender system. When these systems are trained with the sole cause of learning to increase user engagement some dangerous types of content have a chance to flourish and the algorithm tends to offer choices that reinforce what someone already likes or believes, thus being able to act as a catalyst for extremist ideas (Chaslot, Faddoul & Farid, 2020; Hao, 2019). A paper by Covington, Adams & Sargin (2015) described YouTubes methods and how deep neural networks are used to construct is recommendations. First, from the entire video corpus, some candidates for recommendation are generated based on the video category. Then

these are ranked using user history and context, other candidate sources, and video features to narrow it down to a dozen recommendations which will appear in the sidebar.

As of recently, YouTube is contesting this narrative of extremism with three counter arguments: (1) According to their Chief Product Officer it is not the case that extreme content drives a higher version of engagement; (2) The company claims that view-time is not the only key performance measurement; (3) Recommendations are made within a spectrum of opinions, leaving users with a choice to engage or not. While some of these arguments may seem valid, Facebook CEO Mark Zuckerberg has warned before that extreme content in fact does drive more engagement on his platform (Chaslot, Faddoul & Farid, 2020). Even though videos with very extreme content, such as explicit violence, child pornography, and animal cruelty will be removed (Agarwal & Sureka, 2015), there exists a grey area that can nonetheless be perceived as "radicalizing". This grey area definitely has higher engagement than other videos (Ledwich & Zaitsev, 2020). Another point made by Chaslot and colleagues is that YouTube is not at all transparent in the way their recommendation algorithm works, which makes it difficult to identify on what basis the algorithm operates. It is known that the YouTube algorithm uses deep neural networks loaded with inputted data, video watch time, likes, and comments on videos. However, insight in the operation of this algorithm and weighing of certain input metadata remains a mystery. An argument can be made that YouTube users are responsible for their own choice to content, however, the large proportion of watched videos through recommendations makes a case for being careful in these recommendations. Even though YouTube has made efforts to decrease the view-time on this gray-area radicalizing content (Chaslot, Faddoul & Farid, 2020) there seems to be an increase once more since the last efforts were made. Continuation of the study of YouTubes recommendation system is definitely in order and means to identify borderline content can be a great tool in this research area.

2.2 Conspiracy Theories and Language

In order to investigate a language-based approach to identifying conspiratorial content, the determinants of what makes certain use of language conspiratorial have to be defined. There has been much discussion of what counts as a conspiracy theory. Sunstein (2009) argued that conspiracy theories are a subset of the larger category of false beliefs that are characterized by their self-sealing quality. The very arguments that give rise to them make it difficult to rebut or even question them. Another big component, according to Sunstein (2009), is the attribution of extraordinary powers to certain agents. These agents supposedly plan, control others, and maintain secrets. Popper (1966) at an early stage in conspiracy research, argued that these theories often attribute intentional action to otherwise inexplicable events. While this remains true for some conspiracy theories, there are others that point towards events that are indeed the result of intentional action. This attribution error has often returned as a key component of conspiracies (Clarke, 2002; Dagnall et al., 2015; Dentith, 2014). Another theory regarding conspiracies is that these theories always contain a certain substructure: They consist of an agent (or an actor), an action, and targets of those actions. Often the agent is a large corporation or powerful entity and the action has self-sealing elements (Samory & Mitra, 2018). Most conspiracy theories suggest that mainstream accounts of political events are a ruse or an attempt to distract the public from a hidden source of power (Fenster, 2008). Another less described aspect of conspiracy theories may be found in theories such as one by Hofstadter (1964), that states that conspiracy theories often interpret political events in terms of a struggle between good and evil. Often, conspiracy theories are explained as a monological belief system, which is characterized by speaking only to themselves while ignoring context. These belief systems consist of beliefs that are evaluated according to their coherence with other beliefs in the system, rather than external data (Goertzel, 1994; van Prooijen & van Lange, 2014). Another key feature is that these systems operate in a nomothetic explanatory style which explains

events in terms of generalized explanations that appeal to a malign pattern of events in the world. People who have a tendency to adhere to these belief systems have been found to possess certain personality traits associated with mistrust and suspicion (Kofta & Sedek, 2005). Words and discourse related to paranoia, mistrust and negative attitude towards authority can be expected in conspiracy theories (Grzesiak-Feldman & Esjmont, 2008).

All of these conspiracy components lead to believe that we can be on the lookout for certain words and certain language specific to conspiracy theories. Self-sealing statements and words, such as "controls", "fabricates", "cover-up", can be expected in greater number in conspiratorial content (Keeley, 1999; Fenster, 2008). And a greater presence of words describing powerful agents such as "government", "FBI" & "CIA" is not unthinkable, and even often found in theories in conspiracy platforms such as the conspiracy subsection on Reddit (Samory & Mitra, 2018). Enough evidence points towards the use of specific words in describing these conspiracy theories.

2.3 Vocabulary-based Natural Language Approaches and Semantic Word Similarity Approaches

Quite some work has been done regarding the construction of keyword vocabularies in certain categories. Automatic keyword extraction is the process of identifying key terms, phrases, segments, and words from a document that can appropriately represent the topic of that document (Beliga, Mestrovic & Martincic-Ipsic, 2015). Keyword extraction can provide a compact representation of documents or categories and is therefore very useful in classification and clustering of documents (Zhang et al., 2008). Earlier work has used keyword lists to describe documents and use those lists for classification of text documents (Liu & Wang, 2007; Rossi et al., 2014). This method, known as classification through summarization (Shen et al., 2004) has been used to classify web-pages' content and is known to save computational power and reduce dimensionality in classification (Kim, Howland & Park, 2005). While other studies have used these keyword vocabularies as extra input in text classification (Li, Wen & Li, 2003) by presenting a classifier with both the raw text input, as well as the condensed input, using these keyword vocabularies to bias input data has been the limited subject of study. A paper by Ghosh & Desarkar (2018) has recently tried to identify terms that can strongly distinguish between classes, then weigh those terms more in a TF-IDF matrix. Another paper by McCallum & Nigam (1999) already found ways to create class-specific vocabularies and used those to directly categorize text data.

This study aims to collect class-specific features for conspiratorial content, thus creating keyword vocabularies containing features that strongly distinguish between conspiratorial and non-conspiratorial content, then use those to input a measure of semantic similarity in a TF-IDF based input vector. Where some studies have enhanced classical document representation through concepts extracted from background knowledge (Bloedhorn & Hotho, 2004), others have combined TF-IDF features with semantical similarity measures such as Word2Vec to outperform models using either of those methods alone (Lilleberg et al., 2015). A paper by Luo, Chen & Xiong (2011) has researched enrichment of TF-IDF feature values by weighting them using semantic similarity to category labels, however no enrichment using keyword vocabularies has been done. If this study is to enrich TF-IDF features (words) with semantical information (the similarity to keyword vocabularies), some important measures of semantical similarity have to be considered.

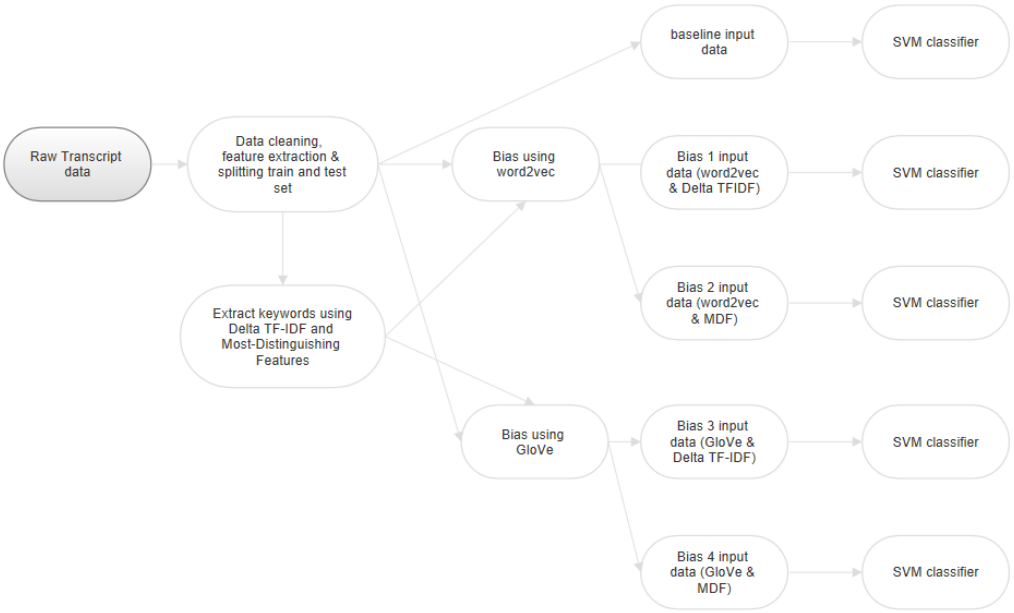
Some methods to measure semantical similarity between words have gained a lot of attention in recent years. First of all, a tool based on deep learning released by Google in 2013 named Word2Vec is considered. It provides an efficient implementation of the CBOW and Skip-Gram architectures for computing vector representations of words. (Rahmawati & Kodra, 2016). This architecture predicts the current word based on context and predicts surrounding words given the current word. It will construct vocabularies from the input data and learn vector representation of words. Then, two words from said vocabulary can

be compared in semantic similarity, which is based on context of words. Some work done by Lilleberg et al. (2015) enriched Word2Vec vectors with TF-IDF information for greater performance in text classification. Another method of capturing semantic similarity is done using the GloVe algorithm developed by Stanford University. This method is a prediction-based approach based on co-occurrence statistics of a very large corpus of words (Pennington, Socher & Manning, 2014). It measures a ratio between two words given their co-occurrence with a probe word and uses this for word embedding. The supposed advantage to the GloVe method is that it does not rely just on local context information of words, but also takes into account global statistics of word co-occurrence to obtain word vectors.

The two previously described measures of semantical similarity will be tested in this study namely by using these to measure word similarity between keyword vocabularies and the rest of the words in our documents. The exact proceedings of this method will be highlighted in the next section.

3. Experimental Setup

The experimental procedure consists of a few steps and in general looks as follows:



After cleaning the data, we bias the input data 4 times using different approaches explained in upcoming sections. All input data and code can be found on https://drive.google.com/drive/folders/121MzpcE1_GH0WY_3bwHnYvkYH5uc1wGW?usp=sharing. Here the final_testing.py script is the final script for the experiment done in this study, and my_functionsmodule_forthesis.py contains all hand-written functions used to run the final script.

3.1 Dataset

Raw Dataset description: The dataset used for this study consists of a combination of two similar datasets. The first dataset consists of 480 transcripts of YouTube videos, which have been ranked in three categories: non-conspiratorial, conspiratorial content, and conspiratorial content including unfalsifiable self-sealing statements. Transcripts have been ranked by three different raters and inter-rater reliability was measured using Fleiss'

Kappa ($k = .445$, $z = 27.5$, $p < 0.0001$). These YouTube videos have been collected by searching in six topics; firearms, gurus, tiny houses, fitness, natural foods, and martial arts. Each of these topics had five seed terms. The top five recommended videos in these searched videos will automatically be watched via a web crawler, then, this process was repeated five times. The 100 most recommended videos for each topic (600 in total), have then been ranked. 120 videos were removed from the set because no transcript was available and another 101 have been removed as they were duplicates. One was removed due to being unlabeled, resulting in 378 ranked transcripts. This part of the dataset was supplied by the research team of Mark Alfano and gathered from an OSF data repository associated with a forthcoming paper. The second dataset was gathered by me, Tolga Akyazi, and Siebe Albers. It consists of 50 transcripts, ranked in the same way as the first dataset, with an inter-rater reliability check using Fleiss' Kappa ($k = .698$, $z = 11.5$, $p < 0.0001$). These videos have been collected in using a few seed terms known to relate to conspiracy theories (5G, reptilians, coronavirus) as well as some seed terms unlikely to relate to conspiracy theories (archery, DJ'ing). Also, from these videos some recommended videos were used. Timestamps were added where possible conspiratorial utterances were found, so discussion on the distinguishing criteria was facilitated. From this dataset, 4 examples have been removed due to being deleted by YouTube at a later stage or due to missing transcripts. After the collection of the data, the multiclass labels were reduced to two labels; non-conspiratorial (consisting of previous non-conspiratorial category) and conspiratorial (including both types of conspiratorial labels) to make binary classification possible. This research focusses mainly on the distinction between conspiratorial and non-conspiratorial content. The distinction between the inclusion of self-sealing statements or not has been considered for later research.

The two gathered datasets were combined, resulting in 424 video transcripts in total. The complete dataset contains 335 videos (79.0%) with a label "non-conspiratorial" (0) and 89 videos (21.0%) with a label "conspiratorial" (1). These were split across train and test sets, with respectively an 80% and a 20% division. This happened in a stratified manner to ensure labels were evenly split across the sets, and both training and test set include data from both datasets used. An interpersonal robustness check between me, Tolga Akyazi, and Siebe Albers was done to ensure data had the same shape, to ensure that the correct transcripts were linked to the correct video labels and to find possible data entry points that were destined for removal. Here we considered data that had empty transcripts, only phonetic transcript data (such as "[music]"), and data with or no rating attached to them.

Data cleaning: Some NLP-methods were used to clean the text data of the transcripts. All words were processed via Spacy's en_core_web_sm parser (Honnibal & Montani, 2017). All non-alphanumeric characters were also deleted and tokens with a length of one or two characters have been removed as well. Then, all remaining words were lemmatized, lower cased, and tokenized for further processing. Some words were manually removed from analysis. Most of them were functional words created by the Spacy parser such as "-pron-", indicating a pronoun, or non-sensical words such as "aaaba", who were later found to conflict with the keyword extraction method. This resulted in a total of 631,035 words with a mean word count of 1473.62 per document.

Feature Extraction: The features extracted from these cleaned transcripts were done using SciKit Learn's TF-IDF vectorizer. The maximum amount of features in this vectorization process is later tested and optimized as a model hyperparameter. This results in a TF-IDF vector matrix, with a row for each transcript in our training data set. Our test set was later transformed to match the constructed TF-IDF vector based on our training data. This is done to ensure that our test data will be handled as unseen data in our classification process.

Keyword Vocabulary Construction: As mentioned before, this study aims to construct

keyword vocabularies consisting of words related to conspiracy theories to later bias the classifier towards use of these words. These keyword vocabularies have been constructed in two different ways; in the first method the TF-IDF values of each word was calculated, and compared between the classes. The difference was calculated between the average of all TF-IDF values of words belonging to transcripts in one class and the other class. If TF-IDF values differ greatly, one can assume that that word is unique to a certain class and therefore is a greatly distinguishing word between the two classes. The largest differential scores (deltas) were calculated, resulting in a top of most distinguishing words. The amount of keywords in this list has later been used as a hyperparameter in the tuning of the classification pipeline. Some examples of this keyword selection method are: "coronavirus", "federal", "censor", "freedom", and "Monsanto". This seems to intuitively match topics and words used in present day conspiracy theories and even captures some of the self-sealing elements described by Sunstein via words like "censor". The first keyword construction method described here is based on the Delta TF-IDF method described by Justin Martineau and Tim Finin in their 2009 paper. They aimed to use differential TF-IDF scores between positive and negative sentiment classes to further increase effectivity of the TF-IDF method and have done so with great success. The second keyword construction method used the original TF-IDF features extracted from the training set to train a basic Support Vector machine classifier. This classifier provided by SciKit-Learn has a built-in function to return the most distinguishing features found in the training process, and returns a list of words as most distinguishing features. The features that belong to the conspiracy class were then used as our keyword list. The length of this keyword list has also been tested as a pipeline hyperparameter. Some examples from this keyword extraction method are similar, but not entirely the same as the first method: "Epstein", "information", "gender", "conspiracy", and "government". These also seem to match popular conspiracy topics, and even describes agents such as the government often used in conspiracy theories. This method will hereafter be mentioned as the Most-Distinguishing-Features method.

Feature Enrichment using Keywords: The unbiased TF-IDF vector described in the previous section was used as a baseline for this study. After this, four other conditions were constructed by applying a bias in two different ways using the two keywords list that were constructed.

For the first and second bias Gensims Word2Vec was used and trained on the transcripts in our training data. The objective of Word2Vec is to generate vector representations of words that carry semantic meanings for further NLP tasks. This method, also known as a word embedding method, is based on the distributional hypothesis where the context for each word is in its nearby words. Semantic similarity of words is therefore computed by looking at neighboring words and assigning vector values to each word based on its neighbors (Mikolov et al., 2013; Ma & Zhang, 2015). Having this Word2Vec model trained on our training data, we can then use this model to compute cosine similarity between words. This cosine similarity thus represents the semantic similarity between two words, based on having similar words in contextual proximity. This model was trained looking at the 10 neighboring words for each word with a dimension size of 150 making 10 iterations over our data. This is quite standard for small datasets such as ours (Dupre, Lesaint & Royo-Letelier, 2018). This model was then used to compute cosine similarity between each word in our TF-IDF feature space and the words in the keyword lists, then averaging the similarity to each keyword per list. This results in a vector with the same length as our number of TF-IDF features, containing similarity scores to our keyword lists. A higher number in this vector represents a greater similarity of a certain word in the TF-IDF vector to the keyword lists. This vector of similarity is then multiplied by each row of both train and test data in our TF-IDF matrix, with each row representing a transcript, thus ensuring that words that relate to our keyword lists on a semantic contextual level have more weight in later classification. This creates our first and second

test conditions: One using Word2Vec in combination with the Delta-TF-IDF keyword approach to bias the TF-IDF vector and one using Word2Vec in combination with the Most-Distinguishing-Features approach to bias the TF-IDF vector.

For the third and fourth biased feature spaces, a different approach to calculating word similarity between the TF-IDF features and the keyword lists was used. This method, called GloVe, is not trained on our training data, but on about six billion tokens from Wikipedia. This model, originally created as an open-source project by Stanfords' Pennington, Socher, and Manning in 2014, is a model for distributed word representation. Training for this model was performed on aggregated global word-word co-occurrence statistics from a very large corpus (Wikipedia 2014) and the resulting representations showcase interesting linear substructures of the word vector space. GloVe is seen as one of the state of the art application to find semantic relations between words. This model, trained on a huge corpus, can provide a cosine similarity measure similar to Word2Vec, albeit more robust as the amount of data it is trained on has many more entries, spanning across a wide array of data, not just conspiracy videos such as the Word2Vec model trained on the dataset used in this study. The pre-trained model was download from the website of Standford University and the choice for the GloVe.6B.100d model has been made as it was the smallest model available and all other models resulted in unnecessarily large computational effort while not improving results. In a very similar measure this pre-trained model was used to compute cosine similarity between each word in our TF-IDF vector and the words in both of the keyword lists. This was also averaged per list, resulting in a vector with the same length as our number of TF-IDF features, describing a value of similarity between each of our TF-IDF vocabulary words and the keyword list. Once again, this vector of similarity was then multiplied by each row of both train and test data in our TF-IDF matrix, with each row representing a transcript, thus ensuring that words that relate to our keyword lists on a semantic level have more weight in the classification process. Hereby the third and fourth test conditions were created: One using GloVe in combination with the Delta-TF-IDF constructed keyword list to bias our TF-IDF vector, and one using GloVe in combination with the Most-Distinguishing-Features approach to bias our TF-IDF vector.

All previous steps result in five TF-IDF vectors as training input for classification, and five TF-IDF vectors that will be used as unseen test data. The training data for classification is again split in a training and a validation set with a random 80%/20% distribution.

3.2 Method & Models

Classification: The classifier used in this study is a linear Support Vector Machine, as it is a well known method for handling text data and has known many applications in recent years (Joachims, 1997; Basu, Watters & Shepherd, 2003; Sarkar et al., 2015). This classifier determines the best decision boundary between vectors that belong to a given group, and vectors that do not. They are used in binary classification tasks, such as the one in this study. These Support Vector Machines are known to perform well when there is no great amount of training data available, as is the case in this study, and therefore seems well-suited for the task. An SVM model is a feature representation of the examples as points in a hyperplane, mapped so that examples of either category are divided by a gap. When presented with new examples, these examples are mapped on either side of that gap, thus resulting in a classification in either category. This method is known to handle text categorization well and is praised for being computationally inexpensive. The original method is invented by Vapnik & Chervonenkis in 1963 and is still widely used today.

Hyperparameters and Tuning: Some variables were considered as hyperparameters for the classification pipeline. Each of these hyperparameters was manually tuned to find optimum performance in the validation stage of classification.

The first hyperparameter consists of the number of keyword in our keywords list. It is

not unimaginable that an optimal number of keywords in these lists will be present. A too large number of keywords may lead to the inclusion of words that do not relate enough to conspiracy theories to make a difference and a too small number may lead to calculating similarity to very specific words or notions, thus specifying the keyword vocabulary beyond the notion of conspiracy theories. Keyword vocabularies with respective size of 20, 50, 100 & 200 words were constructed and applied to the training data. Then these were tested on the validation data to measure performance using F1-score. Results are displayed below in table 1 and led to tuning this hyperparameter to a length of 50 keywords for both keyword methods. Notice here that the baseline doesn't change, which is logical considering that the keyword lists have no influence on the baseline input data.

Table 1
Best performing keyword-lengths for classification. F_1 scores reported for each of the tested input data.

	F1 score				
Number of keywords	Base line	Bias 1	Bias 2	Bias 3	Bias 4
20	0.70	0.74	0.72	0.71	0.69
50	0.70	0.74	0.75	0.72	0.72
100	0.70	0.74	0.74	0.71	0.71
200	0.70	0.74	0.74	0.71	0.71

The second hyperparameter that was tuned in this classifier is the maximum amount of features in the TF-IDF vector. This parameter can be effective to tune, as the data may have some exceptionally rare words, which might add unwanted dimensions to inputs in the future. We might not want to consider the whole corpus during the TF-IDF transformation to enhance the performance of the classifier. This can make the input data incredibly noisy and may reduce the predictive accuracy of machine learning algorithms (Wang, Wang & Chang, 2016). Having relatively more relevant features can reduce dimensionality and improve overall classification performance. The tuning of this hyperparameter is also done manually with maxima of 1000, 4000, 7000 and 10000 features for our TF-IDF vectors before biasing them and train the classifier on our training data. Performance on the validation data was measured using F1-score. Results are displayed below in table 2 and led to tuning this hyperparameter to a maximum of 7000 features.

Table 2
Best performing maximum features in TF-IDF vectorization for classification. F_1 scores reported for each of the tested input data.

	F1 score				
Maximum Features	Base line	Bias 1	Bias 2	Bias 3	Bias 4
1000	0.68	0.62	0.62	0.72	0.69
4000	0.70	0.74	0.74	0.71	0.72
7000	0.70	0.74	0.74	0.72	0.71
10000	0.70	0.74	0.74	0.71	0.71

The third hyperparameter that was tuned belongs to the Support Vector Machine. The C parameter tells the SVM optimization how much you want to avoid misclassifying each training example, and can greatly influence the amount the model overfits. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all points classified correctly and vice versa. C controls the trade-off

between achieving a low error on the training data, and minimizing the norm of the weights. A high value of C implies we allow fewer outliers. C was manually tuned with values of 1, 3, 5 and 10. Performance on the validation data was measured using F1-score. Results are displayed below in table 3 and led to choosing a value of 1 for C.

Table 3
Best performing C for classification. F_1 scores reported for each of the tested input data.

	F1 score				
C	Base line	Bias 1	Bias 2	Bias 3	Bias 4
1	0.70	0.74	0.74	0.72	0.72
3	0.69	0.71	0.69	0.67	0.67
5	0.66	0.73	0.73	0.67	0.67
10	0.66	0.73	0.73	0.67	0.67

This concludes the hyperparameter tuning for our models. Most of the hyperparameters selected are parameters that operate on the level of the input data, as this is the place where the most significant difference between the experimental conditions happens.

Model Evaluation: Model evaluation will be done using three evaluation measures. First of all, macro averaged precision will be calculated. This describes the proportion of positive identifications of the conspiracy class that has been done correctly. This measure has been selected as it signifies how much of the videos classified as conspiracy have been done so accurately. The second evaluation measure used is macro averaged recall. Recall measures the proportion of actual positives that were identified correctly, and signifies how many conspiracy videos went unnoticed by the classifier. Lastly, F1 scores have been calculated, signifying the harmonic mean between the precision and recall scores. This can give us an overall idea of how the classifier performed in regards to both precision and recall. Accuracy measures will not be included as they are often a skewed representation of performance in datasets with a large class imbalance such as this one.

Software & Package Source Acknowledgements: All experiments and code in this study was written and tested in the 2018.2.4 version of the JetBrains PyCharm IDE, running on a Python 3.6 interpreter background. Some important packages for the processing of the data in this study used are: Numpy for the organization and transformation of data (Oliphant, 2006), Pickle for compressing large data variables so they can be retrieved at a later point for a more fluent coding process (as standard python package, the source code is available on: https://github.com/python/cpython/blob/3.8/Lib/pickle.py), OS (van Rossum & Drake, 2007) for making file loading and switching between folders possible, and matplotlib (Hunter, 2007) to deliver visual insight in the dataset and results. Feature extraction processes where greatly helped by Spacy (Honnibal & Montani, 2017) which extracted tokens, lemmatized our text, and acted as a parser for our documents. SciKit Learns TF-IDF vectorizer (Pedregosa et al., 2011) was used to construct the TF-IDF features, representing each word with a value representing the inverse proportion of the frequency of that word in a document, to the proportion it appears in the entire corpus. All the classifiers were also gathered from SciKit Learn. Finally, Gensim delivered the Word2Vec algorithm (Rehurek & Sojka, 2010), and Stanford delivered the GloVe algorithm (Pennington, Socher & Manning, 2014). All other functions, such as the dataset shuffler, the two keyword extraction functions, and a data cleaning and organization function were hand-written for this study.

4. Results

Table 4 shows the performance of our classifier on our validation dataset, whereas table 5 shows the performance on our test set filled with unseen data. Each table shows results of our baseline compared to our 4 input bias methods. Bias 1 signifies the Delta TF-IDF keyword selection method in combination with Word2Vec word similarity, bias 2 signifies the Most-Distinguishing-Features approach in combination with Word2Vec, bias 3 signifies the Delta TF-IDF keyword method in combination with Glove word similarity, and bias 4 signifies the Most-Distinguishing-Features approach in combination with Glove word similarity.

Table 4
Performance of the classifiers on the validation set.

Performance Measure	Baseline	Bias 1	Bias 2	Bias 3	Bias 4
Precision	0.71	0.74	0.74	0.69	0.69
Recall	0.69	0.74	0.74	0.73	0.73
F1-score	0.70	0.74	0.74	0.71	0.71

As we can tell from the results in the validation stage, validation precision, recall and F1 all improved over the baseline using either bias 1 or 2. Even though recall improved versus baseline results using bias 3 and 4, we can see that precision has not reached the same result. The overall score of the model, F1, improves in all bias conditions versus the baseline.

Table 5
Performance of the classifiers on the test set.

Performance Measure	Baseline	Bias 1	Bias 2	Bias 3	Bias 4
Precision	0.73	0.78	0.78	0.76	0.76
Recall	0.62	0.67	0.67	0.69	0.71
F1-score	0.64	0.70	0.70	0.72	0.73

As seen in table 5, a significant increase in all performance measures can be seen when implementing a bias versus the baseline. Where bias 1 and 2 have no difference in performance among them, we can see a great increase in both precision, recall and F1. Bias 3 seems to perform even better overall when looking at F1-scores and bias 4 outperforms other methods.

5. Discussion

This study aimed to improve classification performance of text classifiers by implementing a bias based on semantical similarity to extracted keyword lists. This study tested the hypothesis that conspiratorial content would be recognized to a greater degree by weighing input data. The results described give some evidence that enriching TF-IDF feature data with features based on semantical similarity to class-keyword vocabularies is present. Implementing a bias towards certain words often used in conspiratorial content seems like a valid strategy to improve a classifiers recall and precision rates. As expected, GloVe outperforms Word2Vec in this case, and seems to capture word similarity to a higher degree as Word2Vec does, which is not unthinkable seeing as it is trained on a much larger corpus with a more comprehensive model structure. There is also some degree of

increased recall between the Delta-TF-IDF method and the Most-Distinguishing-Features method, where the latter seemed to be most effective. Some difference was expected, seeing as both keyword lists were composed of some different words. Though both keyword selection methods rely on TF-IDF feature information, a firm improvement in using keyword extraction methods seems present. All these findings support the notion that conspiracy theories can be understood better in terms of certain key words used and that conspiracy theories adhere to a certain vocabulary. By aiming to capture this vocabulary and using that as a basis to enrich our feature data with semantical information, this study succeeded to improve a classification pipeline designed to recognize conspiracy videos. This shows, that in a small and unbalanced dataset as the one used in this study, performance can be improved by operating on a feature level by implementing additional information of interest.

Some notes in interpreting these results must be considered though. Firstly, some discussion regarding the keyword extraction method. This dataset has been constructed in a very structured way and might not be fully generalizable to the entire YouTube video pool. Therefore, our keyword vocabularies might also not be an accurate reflection of actual conspiracy vocabularies, as they may be much richer in words considering conspiracies in general and are probably far less specific (containing more words such as verbs and containing less words such as names). In further research, implementing keyword or vocabulary data from external resources could greatly support the idea that topic-specific keywords can improve classification. There is an abundance of external sources of conspiracy theories, such as the conspiracy subreddit, which can be text-mined for conspiracy related content to identify keywords. Another approach, apart from finding key words, is finding distinguishing *agent-action-target* trigrams in conspiracy such as those in the Samory & Mitra (2018) paper and use these to measure similarity to a corpus. These may be far more descriptive of conspiracy theories than loose words as Samory & Mitra suggested, and yield better results in optimizing classification. Whilst the keywords extracted in this research intuitively seem descriptive of conspiracy theories, as they describe a powerful agent or contain a self-sealing element, perhaps it could offer some insight to also zoom out on a sentence-based approach. Considering other linguistic features such as grammar could prove invaluable to the improvement of classifiers. This can also be applied to the feature extraction, as the basis for this study relies heavily on TF-IDF features, which are not omnipotent. Using more sophisticated features as a base, such as word embeddings, can provide some improvement in performance in NLP-processes such as this.

Another problem that rises with this topic-adhering approach to gathering video data is that it is never certain to say that these methods will work on a large scale. Also, the balance between conspiratorial content and non-conspiratorial content in the entire YouTube video pool can differ greatly from the distribution in our dataset, which could alter the effect of biasing input word vectors. Another point is that the ranking of the videos have all been done by people of a similar scientific background. The notion of what entails a conspiracy theory may vary among a more diverse pool of video raters. Through this paper, perhaps a lid has been lifted on research towards improving classifiers that try to grasp more difficult and complex concepts such as conspiracy theories. Of course applying these methods to other conceptual areas can be very interesting as well.

On the level of classification this study opted for a Support Vector Machine as it is well known to handle smaller and unbalanced datasets without much loss of performance. Even though the selection of this classifier seems justified, it also raises much questions about how other classifiers, such as Recurrent Neural Networks, Naïve Bayes classifiers, and other deep learning approaches might perform. Recurrent Neural Networks are supposedly more robust (Chen et al., 2017).

Taking a methodological approach we can also make some comments about YouTubes recommendation system. YouTube has a large corpus of data available and uses many machine learning and deep learning approaches, such as recurrent neural networks

(Covington et al., 2016). They also have a wide array of metadata available, as well as personalized profile information. This enormous amount of metadata, along with more sophisticated classification methods, can lead us to believe that the effects posed in this study may fall short in comparison to methods already employed by the YouTube algorithm. Perhaps considering personalized metadata in the future can be of great assistance in this field of research.

6. Conclusion

First and foremost, this study aimed to develop a well-rounded classification pipeline for conspiratorial content on YouTube. It explored a possible methodology for the bias weighting of input data towards semantic similarity to certain key words that are specific to conspiratorial content. The question asked in the beginning of the paper: "Can we improve a conspiracy video classifier by implementing a input data bias based on similarity to extracted keywords?" finds a somewhat positive answer when regarding the results of this study. Findings suggest that certain keywords can be descriptive of conspiratorial content and can even be used as input to further improve classifiers, at least to a certain degree. This supports the notion that conspiracy theories adhere to a certain vocabulary and findings regarding keywords suggest that narratives containing agents and self-sealing elements can be captured in keywords. There is a lot of room for further research, mainly in using more sophisticated approaches to determining features, classifiers, and data, however we can establish to some degree an effect of the methodology proposed in this study.

The identification of conspiratorial content on YouTube remains crucial in stopping the spread of inflammatory content, which invokes content bubbles in groups of already radical YouTube users. These types of content will lead to an increase in display of radical behavior (Alfano, Carter & Cheong, 2018; Tufekci, 2018; Chaslot, Faddoul & Farid, 2020). Of course YouTube has announced efforts to reduce conspiracy theories, however, any insight into the language of these theories and possible methods to detect them more effectively, contributes to these efforts. There is hope that tireless research and public criticism of YouTubes algorithm will further diminish view time of radicalizing content.

7. References

Agarwal, S., & Sureka, A. (2015). Applying social media intelligence for predicting and identifying online radicalization and civil unrest oriented threats. *ArXiv eprint 1511.06858, 2015.*

Ahmed, W., Vidal-Alaball, J., Downing, J., & Segui, F.L. (2020). Covid-19 and the 5g conspiracy theory: Social network analysis of twitter data. *Journal of Medical Internet Research, 22*(5), e19458.

Alfano, M., Carter, J.A., & Cheong, M. (2018). Technological seduction and self-radicalization. *Journal of the American Philosophical Association, 4*(3), 298-322.

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

Basu, A., Watters, C., & Shepherd, M. (2003). Support Vector Machines for text categorization. *Proceedings of the 36th Annual Hawaii international Conference on System Sciences.*

Beliga, S., Mestrovic, A., & Martincic-ipsic, S. (2015). An overview of graph-based

keyword extraction methods and approaches. *Journal of Information and Organizational Sciences, 39(1).*

Bloehdorn, S., & Hotho, A. (2004). Boosting for text classification with semantic features. *International Workshop on Knowledge Discovery on the Web,* 149-166. Springer, Berlin, Heidelberg.

Caselles-Dupre, H., Lesaint, F., & Royo-Letelier, J. (2018). Word2vec applied to recommendation: Hyperparameters matter. *In Proceedings of the 12th ACM Conference on Recommender Systems.* 352-356.

Chen, P., Sun, Z., Bing, L., & Yang, W. (2017). Recurrent attention network on memory for aspect sentiment analysis. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.* 452-461.

Chung, C.K., & Pennebaker, J.W. (2008). Revealing dimensions of thinking in open-ended self-descriptions: An automated meaning extraction method for natural language. *Journal of Research in Personality. 42*(1), 96-132.

Clarke, S. (2002). Conspiracy theories and conspiracy theorizing. *Philosophy of the Social Sciences. 32*(2), 131-150.

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtube recommendations. *In Proceedings of the 10th ACM Conference on Recommender Systems.* 191-198.

Dagnall, N., Drinkwater, K., Parker, A., Denovan, A., & Parton, M. (2015). Conspiracy theory and cognitive style: A worldview. *Front Psychol. 2015, 6,* 206.

Dentith, M. (2014) The philosophy of conspiracy theories. *Palgrave Macmillan UK, Macmillan, United Kingdom.*

Faddoul, M., Chaslot, G., & Farid, H. (2020). A longitudinal analysis of YouTube's promotion of conspiracy videos. *ArXiv eprint 2003.03318.*

Fenster, M. (2008). Conspiracy theories; Secrecy and power in American culture. *University of Minnesota Press.*

Ghosh, S., & Desarkar, M.S. (2018). Class-specific TF-IDF boosting for short-text classification: Application to short-texts generated during disasters. *In Companion Proceedings of the Web Conference 2018.* 1629-1637.

Goertzel, T. (1994). Belief in conspiracy theories. *Political Psychology, 15(4),* 731-743.

Grzesiak-Feldman, M., & Ejsmont, A. (2008). Paranoia and conspiracy thinking of Jews, Arabs, Germans, and Russians in a Polish sample. *Psychological Reports 102*(3), 884-886.

Hofstadter, R. (1964). The paranoid style in American politics. *Harper's Magazine.*

Honnibal, M., & Montani, I. (2017). Spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear, 7.*

Hunter, J.D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9(3),* 90-95.

Imhoff, R., & Lamberty, P. (2020). A bioweapon or a hoax? The link between distinct conspiracy beliefs about the Coronavirus disease (COVID-19) outbreak and pandemic behavior.

Joachims, T. (1997). Text categorization with support vector machines: learning with many relevant features. *Proceedings of the 10th European Conference on Machine Learning.* 137-142.

Keeley, B.L. (1999). Of conspiracy theories. *Journal of Philosophy, 96*(3), 109-126.

Kim, H., Howland, P., & Park, H. (2005). Dimension reduction in text classification with support vector machines. *Journal of Machine Learning Research, 6,* 37-53.

Klein, C., Clutton, P., & Dunn, A.G. (2019). Pathways to conspiracy: The social and linguistic precursors of involvement in reddit's conspiracy theory forum. *PLoS ONE, 14,* 11.

Kofta, M., & Sedek, G. (2005). Conspiracy stereotypes of Jews during systemic transformation in Poland. *International Journal of Sociology, 35,* 40-64.

Ledwich, M., & Zaitsev, A. (2019). Algorithmic extremism: Examining YouTube's rabbit hole of radicalization. *ArXiv eprint 1912.11211, 2019.*

Li, C., Wen, J., & Li, H. (2003). Data classification using stochastic key feature generation. *Proceedings of the 20th International Conference on Machine Learning.*

Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and word2vec for text classification with semantic features. *In 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing,* 136-140.

Liu, J., & Wang, J. (2007). Keyword extraction using language network. *2007 International Conference on Natural Language Processing and Knowledge Engineering,* Bejing, 2007. 129-134.

Luo, Q., Chen, E., & Xiong, H. (2011). A semantic term weighting scheme for text categorization. *Expert Systems with Applications, 38(11),* 12708-12716.

Ma, L., & Zhang, Y. (2015). Using Word2Vec to process big text data. *In 2015 IEEE International Conference on Big Data.*

Malouf, R. & Mullen, T. (2013). Taking sides: User classification for informal online political discourse. *Internet Research. 18*(2), 177-190.

Martineau, J.C., & Finin, F. (2009). Delta TF-IDF: An improved feature space for sentiment analysis. In 3rd *International AAAI Conference on Web and Social Media.*

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (1999). A Machine learning approach to building domain-specific search engines. *In IJCAI, 99,* 662-667.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representation in vector space. *ArXiv eprint 1301.3781.*

Oliphant, T.E. (2006). *A guide to NumPy*. 1, 85. USA: Trelgol Publishing.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12,* 2825-2830.

Pennington, J., Socher, R., & Manning, C.D. (2014). GloVe: Global vectors forword representation. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.* 1532-1543.

Pestian, J., Nasrallah, H., Matykiewicz, P., Bennett, A., & Leenaars, A. (2010). Suicide note classification using natural language processing: A content analysis. *Biomedical Informatics Insights.* 3.

Popper, K. R. (1966). The conspiracy theory of society. *Conjectures and Refutations: The Growth of Scientific Knowledge, New York. 4*, 165-168.

Rahmawati, D., & Khodra, M.L. (2016). Word2Vec semantic representation in multilabel classification for Indonesian news article. *2016 International Conference on Advanced Informatics: Concepts, Theory and Application.* 1-6.

Rehurek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.*

Ribeiro, H., Ottoni, R., West, R., Ameida, V.A.F., & Meira, W. (2019). Auditing radicalization pathways on YouTube. *ArXiv eprint 1908.08313, 2019.*

Rossi, R.G., de Andrade Lopes, A., de Paulo Faleiros, T., & Rezende S.O. (2014). Inductive model generation for text classification using a bipartite heterogenous network. *Journal of Computer Science & Technology, 29,* 361-375.

Samory, M., & Mitra, T. (2018). 'The government spies using our webcams'; The language of conspiracy theories in online discussions. *Proceedings of the ACM on Human-Computer Interaction, 2*(CSCW), 1-24.

Sarkar, A., Chatterjee, S., Das, W., & Datta, D. (2015). Text classification using support vector machine. *International Journal of Engineering Science Invention. 4*(11), 33-37.

Shen, D., Chen, Z., Yang, Q., Zeng, H., Zhang, B., Lu, Y., & Ma, W. (2004). Web-page classification through summarization. *In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* 242-249.

Shin, J., Jian, L., Driscoll, K., & Bar, F. (2018) The diffusion of misinformation on social media: Temporal pattern, message, and source. *Computers in Human Behavior, 83,* 278-287.

Solsman, J.E. (2018). YouTube's AI is the puppet master over most of what you watch. *CNET, January, 10.*

Song, M.Y., & Gruzd, A. (2017). Examining sentiments and popularity of pro- and anti-vaccination videos on YouTube. *In Proceedings of the 8th International Conference on Social Media & Society.* ACM, New York, NY, USA. 17.

Sunstein, C., & Vermeule, A. (2008). Conspiracy theories. *John M. Olin Program in Law and Economics, 387.*

Tufekci, Z. (2018). YouTube, the great radicalizer. *The New York Times, 10.*

Tufekci, Z. (2019). YouTube has a video for that. *Scientific American, 320,* 77.

Van Prooijen, J.W., & Van Lange, P.A.M. (2014). The social dimension of belief in conspiracy theories. *Power, Politics and Paranoia: Why People are Suspicious of Their Leaders,* 237-255. Cambridge University Press.

Van Rossum, G., & Drake, F.L. (2009). Python 3 reference manual. Scotts Valley, CA: CreateSpace.

Vapnik, V., & Lerner, A. (1963). A pattern recognition using generalized portrait. *Automation and Remote Control, 24,* 6.

Wang, L., Wang, Y., & Chang, Q. (2016). Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods, 111,* 21-31.

Zhang, Z., Cheng, H., Zhang, S., Chen, W., & Fang, Q. (2008). Clustering aggregation based on genetic algorithm for documents clustering. *2008 IEEE Congress on Evolutionary Computation.*