Identification of Invalid Information about the COVID-19 Coronavirus Pandemic on a Social Network Platform

Georgios Lygeros
Hellenic Police, Department of Patras

Abstract
The outbreak of COVID-19 caused a parallel contagion which affected the sphere of information called infodemic. Social media as a popular communication channel, enhanced the phenomenon of misinformation causing multidimensional effects both in societal and individual level. Twitter as a web forum, host various types of false content that either deliberately or unintentionally were posted from experts, politicians or civilians. This democratized environment may offer the opportunity of opinion exchange but can maximize the consequences of misinformation. Conspiracy theories, false therapies and dystopian future prediction monopolized Twitters daily activity highlighting the need of a supervisory mechanism which would eliminate such content. In this paper, Machine learning techniques are implemented in order to detect fake COVID-19 related content. For this purpose, algorithms of Natural Language Processing (NLP) are utilized.
The data used to train the algorithms are derived from a publicly accessible dataset that contains tweets related to the current pandemic and were published in Greek language. These tweets were classified and annotated in three categories, true, irrelevant, or false. Once a sufficient number of data has been annotated, the most common words are visualized through word clouds for each category. In addition, a set of linguistic and morphological features were extracted from them by applying methods of converting texts into vectors, as well as features related to the subjectivity of the tweets' texts.

Keywords: COVID-19, Social Media, Misinformation, Fake news, Machine Learning

Introduction

One of the most popular hashtags in 2020 on Twitter was #covid19. However, with the emergence of the COVID-19 pandemic, political and medical misinformation has grown rapidly, creating consequences that can actually exacerbate the spread of the epidemic itself. Conspiracy theories, pseudo-scientific treatments and lawsuits are just two indicative categories of misinformation and fake news that have found fertile ground due to the evolving situation. This pandemic of misinformation could not leave our country unaffected.

Given the dangers of spreading fake news, it is essential to address the phenomenon in a timely manner. However, with knowledge of how information is disseminated on a social network such as Twitter, patterns and potentially malicious activities aimed at misleading users can be detected. Computer science and Machine Learning are proving to be well suited for this purpose.

By utilising Machine Learning methods in the effort to detect fake news, the process can be automated, reducing the time required in the effort to control information to detect fake news but also helping to stop its spread.

The aim of this paper is to investigate the automated categorization of Tweets published on the Twitter platform, which are written in Greek and have as their thematic content the pandemic of COVID-19 and its evolution. The categorization is based on the computation of a set of morphological, semantic, PoS (Part of Speech) and statistical features, which are obtained by applying advanced NLP (NaturalLanguageProcessing) techniques to the text of the tweets. The categories into which it is desirable for tweets to be classified are derived based on the validity of their content i.e. whether they contain true, false or irrelevant information in relation to the pandemic. The automatic classification of tweets based on these characteristics is investigated through the application of Machine Learning algorithms.

The contribution of this work lies in the fact that for the first time, as far as we know, an attempt is made to detect false news and automatically categorize news originating from tweets written in Greek concerning the COVID-19 pandemic. Our language makes such an analysis a challenging task, which is why related work is quite difficult to come across. Also, in order to speed up the research, a web-based tool was built which enables the mass categorization of tweets. This tool makes the process of tagging tweets for a volunteer easier and more manageable and also brings about the acceleration of the pace of the process. In addition, the morphological and semantic features extracted are added features which are obtained by applying the TF-IDF method to the dataset. This addition adds new information that other implementations have not been taken into account.

Literature review

Many researches have been conducted focusing on the issue of the automatic categorization of fake content. A lot of them present the challenges that the specific scientific field is facing regarding the implementation of the NLP. These challenges are connected with the mining and processing of datasets and the performances of the models (ShuKai *et al.*, 2017) και (Oshikawa, Qian & Wang, 2018). Buntain and Golbeck (2017) used Twitter in order to detect fake news trying at the same time to identify the most important characteristics that compile the picture of fake news. There are many initiative that attempted to evaluate the trustworthiness of a particular tweet (Qazvinian *et al.*, 2011) or a user (Kang, O'Donovan & Höllerer, 2012), while others have focused more on temporal reputation propagation dynamics (Kwon *et al.*, 2013). Zervopoulos *et al.* (2020) approached the problem of automatic text categorization regarding valid and invalid news based on Twitter platform and concerning protests in the Hong Kong region. The authors of the specific work implemented various Machine Learning (ML) algorithms regardless the language of the Twitter post.

Another issue that NLP works have to overcome is the labelling processes that are followed. Labeling is a crucial step as the generated results are strongly connected with the quality of the labeling process. The available dataset usually is not categorized and the categorization is a part of the implementation.

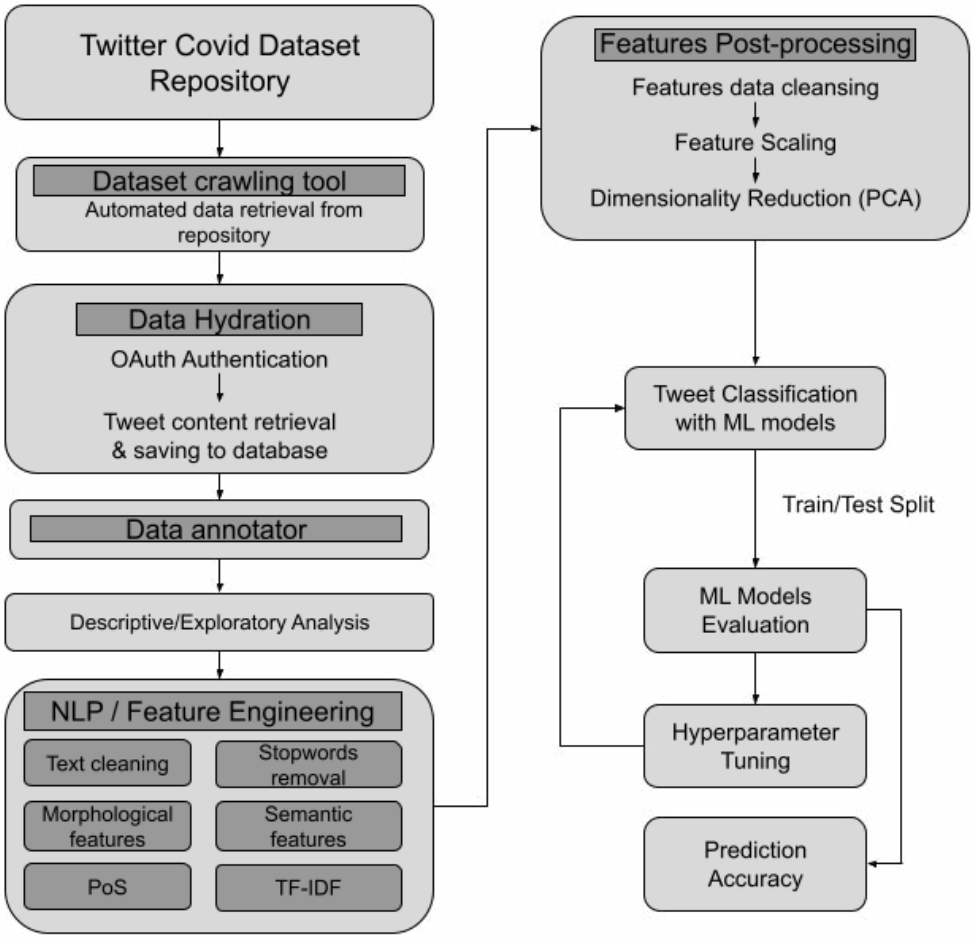
Recently a number of COVID-19 oriented datasets have been published whose data are not categorized based on the criterion of the detection of misinformation (Cui & Lee, 2020; Memon & Carley, 2020; Qazi *et al.*, 2020). Although there are datasets which characterize the data as disinformation data (Brennen *et al.*, 2020), datasets containing data with "true" news about COVID-19 are also rare. Memon and Carley (2020) focused on the characterization of disinformation communities on the subject of COVID-19 through data collected from Twitter.

Methodology

System Architecture and Implementation

The system implemented in this thesis can be analysed according to the diagram in Figure 1. Firstly, the identifiers of the tweets of interest are retrieved from the online repository hosting the dataset. For this purpose, a special crawler tool was created, which can search the data in the repository using criteria such as the date and language of the tweets. Then, based on the tweets' identifiers, a connection to Twitter's API for hydration of the tweets is made and the tweets are stored in the mongoDB database. The process of tagging the data into three categories follows. To facilitate this time-consuming process, a web application was implemented which downloads the content of the tweets stored in the database and through appropriate interface tools allows the quick selection of the most appropriate category to which each one belongs

Figure 1. System Implementation



After the tagging process is completed, a descriptive analysis of the data is performed for basic understanding and drawing conclusions about their structure and finding characteristic patterns. This is followed by a process of applying natural language processing methods, through which the available texts from each tweet are cleaned of unnecessary elements or elements that do not contribute meaningfully to the sentence and a set of features is extracted from the morphology, topic and words of each text. In addition, the TF-IDF method is applied through which the most significant words in the available dataset are identified.

This is followed by the process in which the extracted feature set, together with the categories of significant words obtained by applying TF-IDF are added to a single feature matrix. Post-processing techniques such as scaling and extraction of the most significant features by PCA method are applied to this matrix to reduce the dimensions of the matrix. After this step, the features matrix is divided into train-set and test-set which are then used to train and evaluate the machine learning algorithms being tested. Finally, for each algorithm, the basic parameters are optimized to obtain more accurate results.

Data collection

To solve the problem of identifying and categorizing tweets, a portion of the COVID-19 Twitter chatter dataset was used (Banda *et al.*, 2021). This dataset started to be generated from March 11, 2020, yielding over 4 million tweets per day. Daily hashtags, references to other users, emojis and their frequencies have been included. To make the dataset easier to use, the language in which the tweets are written is included in addition to the unique identifiers (IDs) of the tweets. The full dataset covers all languages; however, the most prevalent ones are English, Spanish and French. The dataset includes 903,223,501 tweets and retweets. In addition, a clean version without retweets is provided (226,582,903 unique tweets). For the convenience of NLP applications, the top 1000 most frequently en-

countered terms are additionally provided, as well as the top 1000 bigrams and trigrams, which are stored, separated by day, daily in an online repository on the github platform.

Description and download of data

The information related to the tweets within the datasets stored on github consists only of their unique attribute (tweet ID), the date and time of publication, the language they are written in and the country they originated from.

For the purpose of this Thesis, tweets published between November 1, 2020 and December 30, 2020 were used. To achieve the collection of these tweets, a tool was built using the Python language that allows automatic storage of tweets from github to the local computer storage in CSV format. This tool also gives the possibility to select the range of dates for which the collection of tweets is desired and the language in which they are written. In this particular case, the dates are the range mentioned earlier while the language is Greek whose abbreviation in which the tweets are stored on github is "el". Important to say is the fact that the tweets are collected from the clean versions of the dataset so there is no concern about duplicates of the tweets being collected. A total of 61,147 tweetIDs were collected.

Tweethydrator

Obviously, the information stored locally is not sufficient for the purposes of this Thesis as the full texts of the tweets need to be available in order to perform the appropriate analysis. To achieve this a tool was built to interface with the Twitter API. Through this tool, the data of the tweets included in the local data is identified, their content is downloaded locally to the server running the tool and then stored in a mongoDB database.

Data annotation

The algorithms used in this Thesis require the prior knowledge of the category a tweet belongs to depending on how valid their thematic content is in order to be trained correctly (to perform accurate training) and then generate models that will categorize new tweets accurately. In the context of this Work, the discretization of tweets into three categories was chosen. As there was no professional support in the tagging process, only three categories were chosen which characterize a tweet according to its content. These categories are:

1) Tweets which are objectively true. Such tweets are for example news posts related to the pandemic, news about Sars-Cov-2 virus, posts related to pandemic containment measures and so on.

2) Tweets that are considered false. False posts are defined as posts that are related to the virus but have controversial or satirical content. However, this classification is difficult as it requires knowledge of the facts that are being reported.

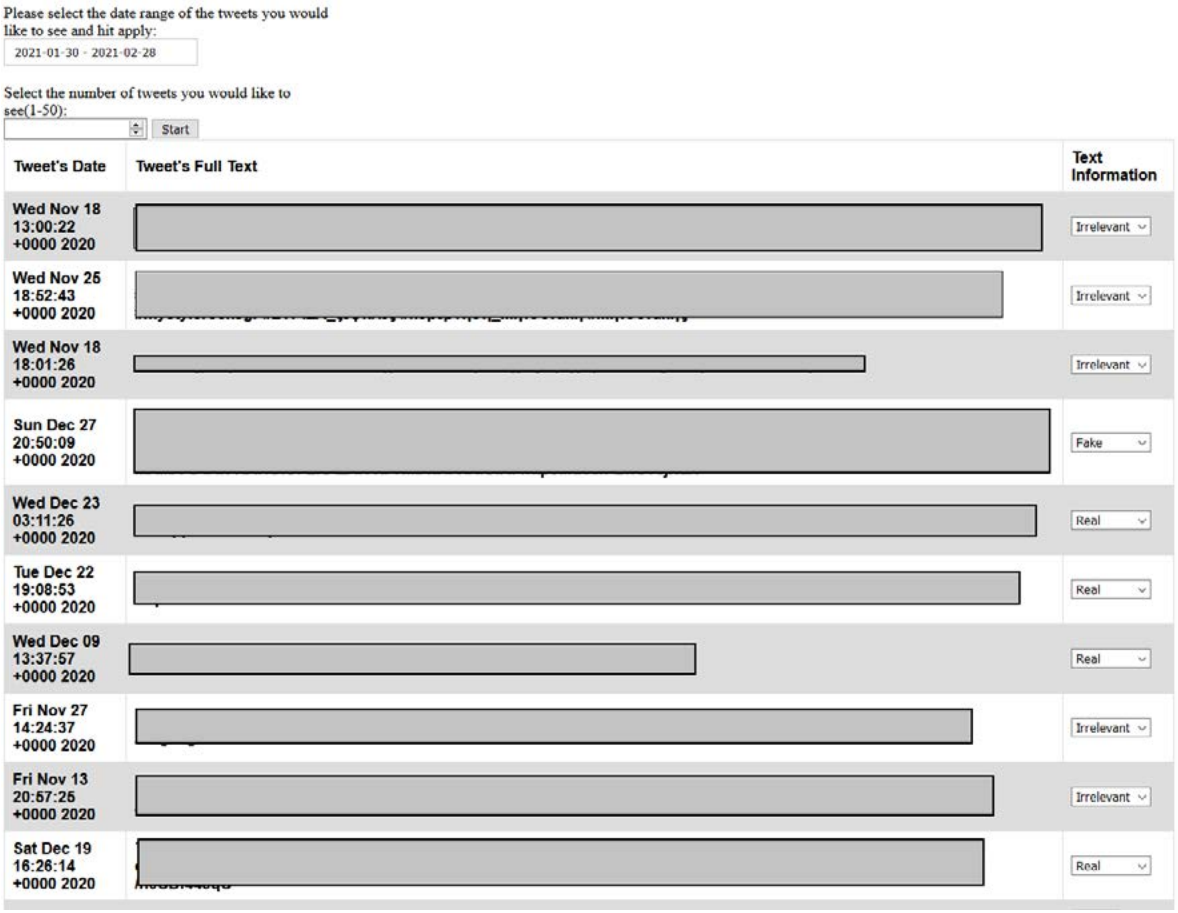
3) Finally, publications which are not related to the development of the events of the pandemic are classified as irrelevant. For example, the post "With #COVID19 only Playstation and Netflix" is easily classified as irrelevant.

It should be noted at this point that the selection of the curves based on which to discretize the categories of the dataset is a complex task that requires a good knowledge of the problem and therefore the correctness of the categories and the method of characterization of the tweets can be reviewed.

However, for tagging another problem exists. Selecting tweets one by one and adding tag is an extremely time-consuming process. For this reason, an online annotator tool (online Tweet Annotator) was built using the Python language with the help of its library, flask, or which is oriented towards web application development.

Through this application a user can select a date range for which he/she wishes to tag tweets and also the number of tweets he/she wishes to display on his/her screen. These tweets are randomly selected from the database as long as they are within the range selected by the user and have not been tagged previously. The tweets are then displayed on the website showing their full text. The tweets are then stored in the mongodb database, updated by adding their categorization to their information, following a user selection.

Figure 2. Online Tweet Annotator



For the purposes of this paper, a total of 3931 tweets were tagged. Of these tweets, 1906 belong to the real category, 1017 to the fake category and 1018 to the irrelevant category. The distribution of annotated tweets is shown in Figure 3.

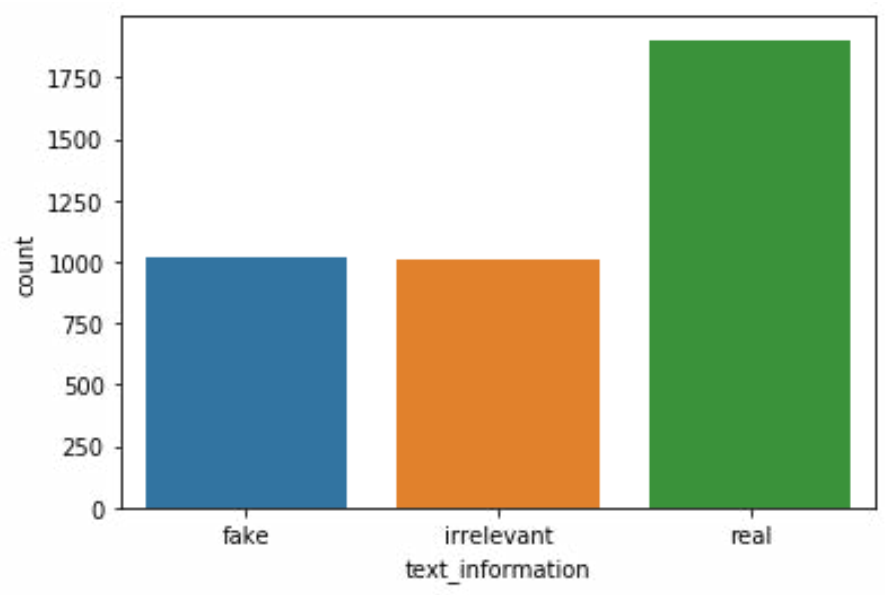
Descriptive analysis of the data

In order to understand the available data, check their validity and draw primary conclusions about the categories chosen to discriminate the dataset (fake, irrelevant, clear), a descriptive analysis of the available data is performed. In particular, it is checked whether there are incomplete fields or fields that have been lost due to an error. In addition, descriptive graphs are extracted which depict various statistics such as for each category the number of words contained in the tweets as well as the length in characters.

Extraction of features

To extract the features, first those tweets are retrieved from the mongoDB database that have been categorized and then converted into a Pandas Data Frame to make them easier to manage. The fields chosen to be retrieved from the tweets' information are their full text, their publication date and the category they belong to. Through an iteration structure that runs through the entire content of the Data Frame, various preprocessing techniques are performed, through appropriate functions, to extract the features effectively. The extraction of all features is done through special functions implemented for this purpose, in combination with functions from the spaCy and NLTK libraries in cases where this is necessary.

Figure 3. Distribution of annotated tweets



Text cleaning and extraction of morphological features

Two basic functions were implemented with which to perform text cleaning. Text cleaning is defined as the stripping of URLs, emojis, entities (mentions), and hashtags from their content. This was implemented with the help of a function that was implemented which cuts out the above according to specific regexes that cover the criteria by which they appear. What this function returns is the content of the texts of the tweets without the parts that have been cut off with the clean_text nomenclature, which is very useful for extracting various features as in some cases it is necessary for the texts to be in this format.

The next function implemented for clearing the texts converts the texts into lists of tokens. Before this is done some steps are taken to ensure that the tokens have the desired format. These are as follows:

- Deleting emojis from the text of the tweets
- Deleting all digits from the text and replacing them with the blank
- Deleting all the punctuation marks
- Deleting all stopwords (via spaCy and NLTK functions)
- Deleting tokens of less than 3 characters

The lists of words returned by this function contain the content of the text free of what was deleted during its execution with the words nomenclature.

The clean_text returned by the above function is used as input to the functions which extract most of the morphological features. To be precise the extraction of the features concerning, the length of the text, the count of the different punctuation marks ("?!", "?", "!", ".", ";" etc.) and their total number and finally tweet_entropy is done via functions implemented with "plain" Python in combination with the re library which provides an easy way to identify regex within the text. The tweet_entropy is extracted by a function implemented that does the mathematical calculation for the entropy of the text.

There are two more functions which take clean_text as input. The first one calculates the number of consonants and vowels present in a tweet and therefore the corresponding attributes are calculated through it. First, any suffix and tone of words are removed from clean_text and then each character is categorized according to whether it is a consonant or a vowel. As a result, their count is easily computable. The second function returns using "plain" Python the number of capitals, lowercase, digits, letters and the letter-to-digit ratio which also correspond to the corresponding attributes.

The attribute corresponding to the average word count per sentence comes from a function which, with the argument clean_text, initially converts the text content into tokens with the help of the NLTK library which supports this function in Greek as well. Then the punctuation marks are subtracted and from there the average is calculated with a simple calculation.

The last features that use clean_text to produce the text have to do with the number of consecutive consonants, the number of consecutive vowels and the number of occurrences of repeated identical characters. These three attributes are computed through a function that has similar logic to the function that computes the total number of consonants and vowels described above. The calculation of the first two features is calculated directly from lists of characters (consonants, vowels) generated at runtime, while the number of consecutive occurrences of a character (>3) is calculated from a list containing all characters of a text.

The function that returns the words is useful in extracting two features. The first attribute concerns the average length of words which is computed by a function that takes words as an argument. The first step that this function performs is to call the function that generates clean_text by giving the list of words as an argument. Then the calculation of the average is again done through a simple mathematical calculation. The second feature concerns the number of words present in a text which is quite easy to calculate through the length of the words list.

Finally, there are features which can be calculated from the original text of the tweets without any processing. For example, the attributes related to the number of urls, mentions and hashtags are generated through functions that check the content for words starting with "http{.....}", "@" and "#" respectively, which are numbered. The attribute having to do with the number of entities is calculated from the sum of the above. It is important to say that a check is performed to see whether a url is functional or not, so in the counting only the functional ones are taken into account. Stopwords are one of the attributes counted via a function that again has the original text of a tweet in its argument. This function returns the number of stopwords contained in a tweet with the help of the functions that extract them from the spaCy and NLTK libraries. Although these functions recognize most of the Greek stopwords, some were added that these functions do not recognize.

Exporting SemanticFeatures

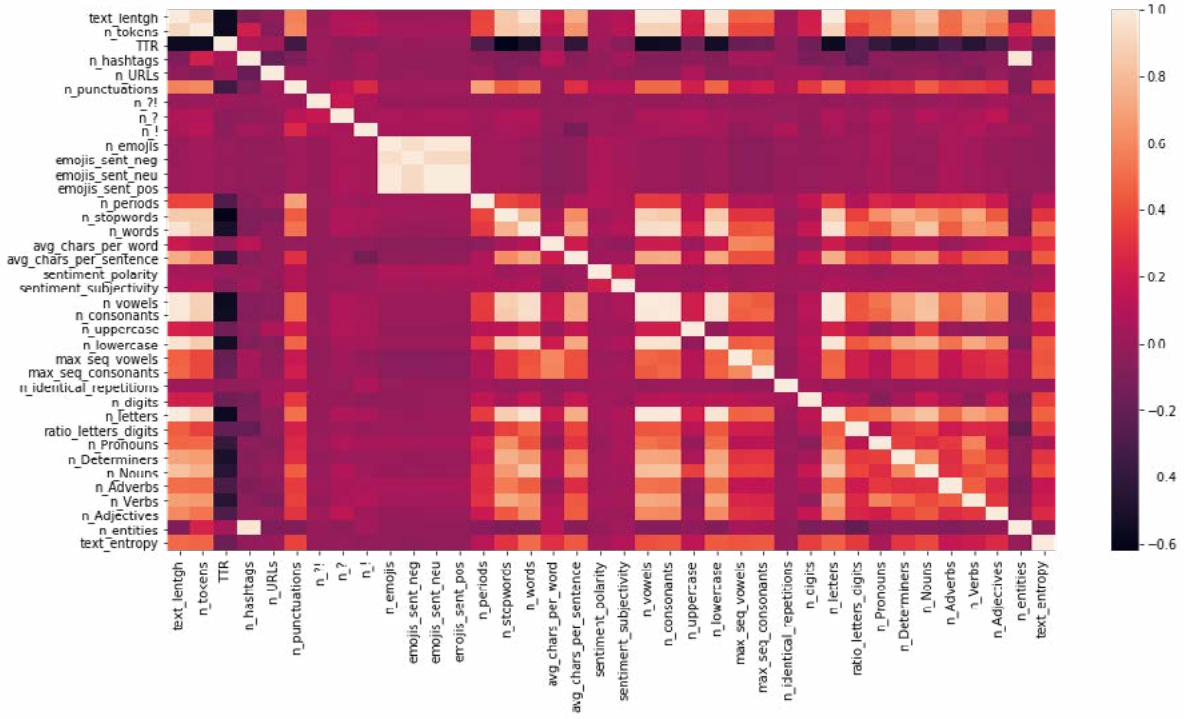
Exporting semanticfeatures is a difficult task to implement. However, as the field is constantly evolving, libraries have been developed which can extract such features in a partially automated way. For the purposes of this thesis, the spaCy library was used through which, with two simple commands, the objectivity of a text and its polarity can be extracted, and thus the corresponding features are computed directly. Also a successful metric, is the counting of positive, negative and neutral emoji contained in a text. The computation of these attributes is done in two steps. In the first stage all emoticons contained in a tweet are searched through a function and then in the second stage they are categorized into negative, positive and neutral emoticons and counted. The former are implemented through regexmatching, a very common way to find emoji as well as the categorization is done based on a list of emojis containing all possible emojis and their rating, which is provided by the emosent library.

Exporting PoS (Part of Speech) features

The calculation of PoSfeatures is done through the nlp_post_processing function implemented with the help of experts in the spaCy library, which has special tools for calculating such features. Initially, LexicalRichness is used through which a metric is computed that represents the "richness" of the text with respect to the variety of words used. Through this metric another function of LexicalRichness can be used which calculates the TTR of the text which corresponds to the corresponding feature. Then, three lists are generated containing tokens into which the text is divided, the PoS categorizations found and the lemmas (lemmas) generated respectively. These lists are generated via special spaCy text processing and parsing functions that support the Greek language and grammar. From the list of categorised PoS found, the features concerning the number of occurrences of pronouns, determiners, adjectives, nouns, adverbs and entities present in a tweet are directly computed. In nlp_post_processing there are also the functions that compute the objectivity and polarization of the text.

The set of extracted features and the degree of correlation between them is illustrated in Figure 4:

Figure 4. Correlation matrix of the key features extracted from the texts of the available tweets



Calculation of the TF-IDF feature

Next, the TF-IDF method is implemented using the TfidffVectorizer function of the scikit-learn library. To generate the text vectors needed to implement TF-IDF, a function is used to join the entries generated by nlp_post_processing. This is done so that the number of different words, and thus the number of feature dimensions, is significantly reduced as the TfidffVectorizer function generates features for each individual word it detects within the text. A set of features are thus extracted which correspond to the most significant words in the dataset. These features are combined with the previous morphological features, PoS and semanticfeatures, into a single featurematrix which has dimensions (3931 x 10923), i.e. a total of 10923 features are computed!

Algorithm training

Before training the algorithms, the collected features are transformed so that they all have the same range of values. The Standard Scaler class of scikit-learn is used for the normalization.

After the features are transformed, PCA is then applied to reduce the dimensionality of the features matrix, which is deemed necessary due to the number of features. PCA is parameterized to extract the minimum number of dimensions needed to maintain 95% of the variance. The results show that instead of 10923, only 2745 basic features are needed! It also turns out that, this data represents almost 25% of the size of the original feature smatrix. In a way, this is a kind of Feature Selection. Obviously, after dimensionality reduction, the feature matrix takes up much less space and this can significantly speed up a classification algorithm (such as classification based on the SVM algorithm).

Finally, the data is divided into train and test. The train data will be used to train the learning algorithm and the test data will be used to verify the results. For the separation, 80% of all available data is used as train data and 20% as test data.

The SVM algorithm is the first algorithm tested for its ability to classify tweets based on the classes defined. Initially the parameters of the algorithm are set which are as follows:

- Kernel: the “rbf” kernel is used.
- C: 1
- Gamma: Scale. this means that the gamma is derived from the ratio 1/ (<number of features> x <scale of trainmatrix>)

Hyperparameter Tuning is then applied using the GridSearch method (Bao and Liu, 2006). In this case, “line-

ar" is additionally tested as kernel, as well as different combinations of C and gamma values. Next algorithm tested is Random Forest. For Random Forest a similar procedure is followed. Initially training is done with the initial training parameters. Then using the function Randomized Search CV a set of combinations of the parameters is generated, from which the algorithm randomly selects which ones to train. Finally, training is performed using Multinomial Naïve Bayes. As this algorithm relies on the use of probabilities it cannot accept negative values. Therefore, in this case the data is normalized in the interval 0 to 1 using the Min Max Scaler class.

Results

This chapter presents the results of the descriptive analysis and the Machine Learning algorithms applied. First, within the descriptive analysis, some basic statistics characterizing the tweets are visualized.

Figure 5. Number of characters in a tweet depending on the category it belongs to.

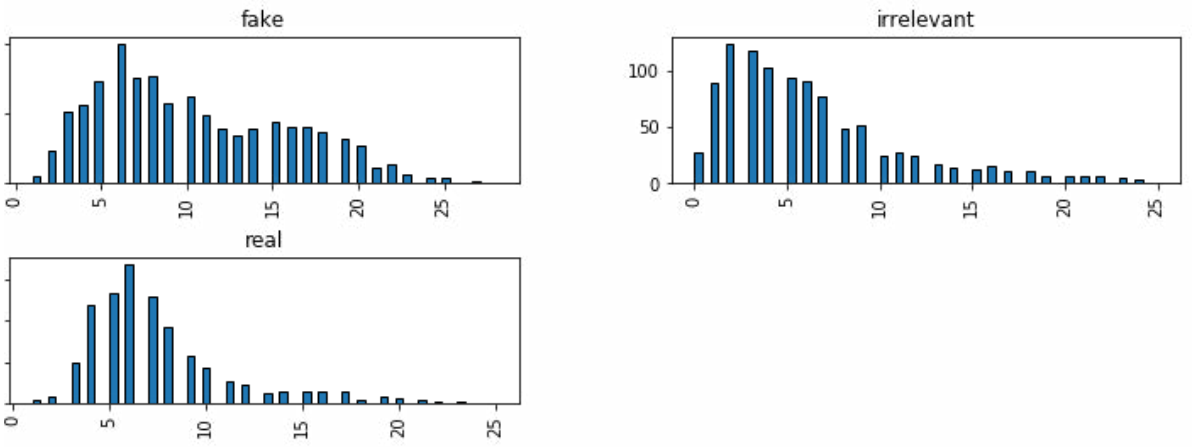
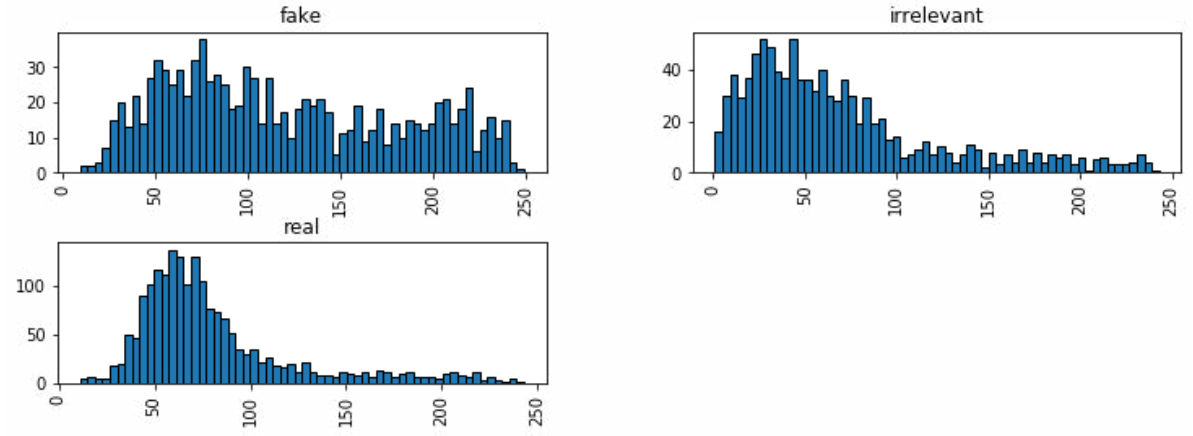


Figure 6. Number of words in a tweet depending on the category it belongs to.



As can be seen from Figures 5 and 6, there seems to be some correlation between the length of the tweet in words and the number of characters with the fake category, i.e. compared to the other two categories, many tweets have a larger number of words and characters.

In addition, visualization of the most important terms in wordclouds is done. More specifically, for each category the 50 most frequently encountered words are identified and visualized. In addition to the most frequent words per category, the most frequent words in the dataset are also presented, which are visualized in Figure 7.

Figure 7. Wordcloud for the whole dataset

In the following, the results of the algorithms tested for the categorization of tweets are presented and analyzed. As described in the previous chapter, the training of the set of algorithms is done using those features that correspond to 95% of the dataset. As the results of the algorithms show, although with a low percentage, it is possible to identify a tweet as real, fake or irrelevant based on the characteristics described in the previous chapter. Among the algorithms tested, the best performance is achieved by the Random Forest algorithm using the Randomized Search method as it better distinguishes tweets of each category compared to the other algorithms. Worst performance was achieved by the Multinomial Naïve Bayes algorithm.

Table 1. Summary results of the algorithms tested

Algorithm	Precision	Recall	F1-Score
SVM	0.68	0.53	0.51
SVM + Gridsearch	0.59	0.57	0.57
RandomForest	0.65	0.61	0.62
RandomForest + Randomizedsearch	0.66	0.61	0.62
MultinomialNaiveBayes	0.65	0.47	0.45

Conclusion

Currents' work aim was to approach computationally the posts in the social network of Twitter, assisting the task of identifying false or irrelevant posts related to the current pandemic COVID-19 and its spread in Greece. We focused on Twitter as its structure and main features allow the detection of trends which are related to the current events. The aim of the activities was to find morphological features of the tweets, as well as features related to the subjectivity of the texts, which allow automatic discriminating between false and true statements in order to categorize them according to their reliability.

For the purposes of our work, we used a ready-made dataset related to tweets that are related to the pandemic. For convenience, our study period is only the period between November 1, 2020 and December 31, 2020. Two tools were implemented, one to automatically retrieve from the dataset the identifiers of tweets that were published during the specific period and are written in Greek. In addition, another tool is implemented which for each of the collected identifiers retrieves their full content from the Twitter platform and stores it in a MongoDB database.

As Machine Learning classification algorithms rely on the use of training data for which the classification category is known in advance, a part of the work involved recording the category for the collected tweets. A total of 3931 tweets were manually classified into three categories, real, fake and irrelevant. To speed up this process, a new tool was created to allow for quick viewing of tweets and tagging.

From the data, by applying transformation methods these 38 features were extracted which are related to the linguistic morphology of the tweets, their subjectivity, sentiment analysis and the type of words used. Moreover, an innovation of this Work is the use of features obtained by applying the TF-IDF method to the texts of the collected publications. However, as the dimensions of the features were exploded, it was necessary to use dimensionality reduction using the PCA method.

The application of these features to Machine Learning algorithms for automatic classification showed unsatisfactory but encouraging results for solving the problem. The cause of the low accuracy of the algorithms is traced to two factors. On the one hand, the difficulty of discriminating even for skilled personnel between publications of different categories, as the distinction between correct and false news is based on factual knowledge. This cannot be achieved by exploiting only linguistic features. Also, the number of categories into which tweets are distinguished is certainly expected to affect accuracy, as specialized knowledge is also required for the number and characterization of the categories selected.

With these two findings as a starting point, the impetus for further research on the problem is given. To begin with, the extraction of knowledge about the subject matter and the use of this information in classification can be studied. Also important is the contribution to the validity of a publication of the credibility of the user who made the publication. Finally, the discretization of tweets into different categories as well as the characterization of larger volumes of data will certainly reveal new ways to achieve the final goal.

In conclusion, at a time when conspiracies and fear abound, the police science/ law enforcement have an opportunity to protect the public from misinformation, as well as to enforce the law to protect public health and public safety. This research is intended to help law enforcement authorities detect immediately false news and provide an understanding of what is driving the misinformation that might compromise public safety.

Gaining the first-mover advantage by distributing correct information about COVID-19 and going first, has been shown to reduce the influence of subsequent misinformation, as the first piece of information heard tends to be what sticks.

By detecting fake news in a timely manner, citizens will be able to be informed of the truth and not be misled by individuals or organizations whose purpose is to harm social cohesion, the state and the well-being of society in general.

References

- Banda, J. M. *et al.* (2021) 'A large-scale COVID-19 Twitter chatter dataset for open scientific research—an international collaboration', *Epidemiologia*. MDPI, 2(3), pp. 315–324.
- Bao, Y. & Liu, Z. (2006) 'A fast grid search method in support vector regression forecasting time series', in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 504–511.
- Brennen, J. S. *et al.* (2020) *Types, sources, and claims of COVID-19 misinformation*. University of Oxford.
- Buntain, C. & Golbeck, J. (2017) 'Automatically Identifying Fake News in Popular Twitter Threads', in *Proceedings - 2nd IEEE International Conference on Smart Cloud, SmartCloud 2017*, pp. 208–215. doi: 10.1109/SmartCloud.2017.40.

- Cui, L. & Lee, D. (2020) 'Coaid: Covid-19 healthcare misinformation dataset', *arXiv preprint arXiv:2006.00885*.
- Kang, B., O'Donovan, J. & Höllerer, T. (2012) 'Modeling topic specific credibility on twitter', in *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pp. 179–188.
- Kwon, S. *et al.* (2013) 'Prominent features of rumor propagation in online social media', in *2013 IEEE 13th international conference on data mining*, pp. 1103–1108.
- Memon, S. A. & Carley, K. M. (2020) 'Characterizing covid-19 misinformation communities using a novel twitter dataset', *arXiv preprint arXiv:2008.00791*.
- Oshikawa, R., Qian, J. & Wang, W. Y. (2018) 'A survey on natural language processing for fake news detection', *arXiv preprint arXiv:1811.00770*.
- Qazi, U., Imran, M. & Ofli, F. (2020) 'GeoCoV19: a dataset of hundreds of millions of multilingual COVID-19 tweets with location information', *SIGSPATIAL Special*. ACM New York, NY, USA, 12(1), pp. 6–15.
- Qazvinian, V. *et al.* (2011) 'Rumor has it: Identifying misinformation in microblogs', in *Proceedings of the 2011 conference on empirical methods in natural language processing*, pp. 1589–1599.
- ShuKai *et al.* (2017) 'Fake News Detection on Social Media', *ACM SIGKDD Explorations Newsletter*. ACM PUB27 New York, NY, USA, 19(1), pp. 22–36. doi: 10.1145/3137597.3137600.
- Zervopoulos, A. *et al.* (2020) 'Hong Kong protests: using natural language processing for fake news detection on twitter', in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pp. 408–419.