Semantic Framework for Practicing Data Science in Public Health Organizations during the Covid-19 Pandemics

Radmila Juric
USN, Norway
rju@usn.no

Elisabetta Ronchieri
INFN CNAF, Italy
elisabetta.ronchieri@cnaf.infn.it

Abstract

This paper proposes a semantic framework based on software architectures for accommodating data science practices to the needs of Public Health Organizations (PHO), during the covid-19 pandemics. The goal is to create an environment suitable for deploying data science on an ad-hoc basis, upon the selection of data generated by governments, non-government organizations, public databases and social media, but guided by PHO own needs and expertise. It is important to run predictions, through learning technologies, which may depend on circumstances and situations relevant for PHO in the particular moment and thus enable better decision making in the time of the pandemic. The proposed software architecture relies on its deployment within integrated development environments and plug-ins/APIs towards software tools, and libraries for (a) data gathering and preprocessing, (b) predictions with learning technologies (c) reasoning with semantic technologies and (d) including human intervention to aid in understanding the situation in which PHO questions may be answered. The illustration of the proposal is uses the sentiment analysis of twitter data relevant to covid-19 and classification of tweets with machine learning.

1. Introduction

The covid-19 pandemic has opened the door to a variety of data processing, which helped to disseminate information relevant for understanding and managing the pandemic. The success in disseminating the data and results of its processing is in the efficient, almost ad-hoc generation/collection of data, particularly from social media, and their quick processing with software technologies. This research is a continuation of our interest in:

a) creating software architectures across different types of data processing and software applications using various technologies [1,2,3] and

b) using twitter data for decision making across problem domains [4,5,6].

Merging a) and b) will give computational models which may extract the semantic from a variety of data and directly enhance decision making based on its computational results [1,7,8]. Obviously, this research is triggered by computer scientists, in order to explore which types of computations we would need in managing the pandemics, and which types of data would be suitable for using existing or creating new software solutions. At the same time, the research focuses on the needs of PHOs in the time of pandemics. It illustrates the feasibility of using data science practices and Machine Learning (ML) algorithms for PHO own needs and thus making PHO efficient and independent from heavy weight software technologies and data scientists.

The undoubted contribution of the twitter platform, towards knowledge dissemination and sharing is well known [5,6, 7,8,9]. Twitter has become a global source of information used in many domains and healthcare management is not an exception. Even in the light of privacy laws and GDPR, it is unlikely the twitter platform would lose the prestigious role of having a powerful data repository within the reach of everyone and anyone. Therefore it should be included in data processing of any PHO if they need instant and ad-hoc access to tweets relevant to covid-19.

In order to extract semantics from live and user generated data, a variety of algorithms from Natural Language Processing (NLP) has gained popularity and thus enabled the manipulation of the text data. Twitter sentiment analysis in particular has become a sine-qua-non of our modern computational demands across problem domains [10,11]. It has been supported by ideas from the NLP, vocabularies and semantic lexicons, and focused on statistical predictions. The undisputed popularity of ML allows running predictions at any time and for almost any purpose and therefore there is no limit to obtaining predictive inference for PHOs.

However, these views could be challenged by computing scientists. Considering that the extraction of semantics from any type of data, including tweets, is rather risky, with NO safe and tested computational models in the background, may create problems.

Problem 1 - Practices, claimed to belong to the Data Science field, are not accessible to a wider community.

You need to be an "expert" in data science in order to prepare data and run algorithms. However, we cannot assume that a PHO, in the time of the pandemic, had a small army of data scientists, working around the clock, and trying to make sense of any data. PHOs need lightweight software applications which would allow running prediction and sentiment analysis according to their own way of thinking, and deciding on which data should be used for prediction and which algorithm would work for them. In times of emergencies, for example, PHOs may react instantly and use twitter data by customizing its processing and adapting strict rules defined in sentiment analysis [12,13], to their needs.

Problem 2: Automating sentiment analysis through widely available software tools, without knowing exactly how these tools create results, and which exact algorithms are housed behind various functions these tools offer, should NOT be encouraged [14]. For example: how do we know, in cases of automated feature selections for a ML classifier, that the tool would act according to the semantic stored in data? Are NLP algorithms and lexicons enough for defining an ML classifier's features? Where is the proof for this?

Problem 3 - The fact that we can easily compute, and run ML algorithms for getting predictive inference upon the abundance of data, does not mean that we produce trustworthy results of computing [15,16] because:

(a) We must make sure that the semantic of data has not been affected before its processing starts. In cases when the semantic of data changes, we should be aware of it and act accordingly.

(b) We have to ensure that the chosen algorithm(s) is(are) suitable for the chosen data and would create results relevant for the problem domain.

It is very difficult to address Problems 1-3 above and a silver bullet solution might not exist for the time being. Therefore the way forward might be to depart from standard practices of data science and allow human intervention in finding answers to questions PHOs may have. For example, if a PHO would be interested in performing a sentiment analysis, the best possible way of managing the semantic of tweets, would be through ML classifications and the careful management of choices of features in definitions of ML classifiers, where the PHO should have their own say.

Data science is practised by individuals, using an enormous amount of software tools, libraries and APIs, in non- standardized processes. Therefore defining either a computational framework or software architectures, which would generate software solutions that can assist PHO in decision making during the pandemics might be the way forward. These software applications should be PHO friendly and allow running ML predictions with the involvement of PHO,

The paper is organized as follows. In the related work we overview a situation in publishing after more than 18 months of the pandemics. The number of publications in this field is rapidly growing as we write. However, there are no papers which come close to our idea of creating an operating environment, PHO friendly, where PHO expertise is a decisive factor on how to use data science to answer their own questions. Section 3 outlines the background and scenario of this research and define research questions. Section 4 gives the proposal as a semantic framework, which accommodates software architectures, for processing data according to PHO needs. It also illustrates the way of identifying data sources and data preparation for using predictive technologies. Section 5 and 6 illustrate two experiments and section 7 gives the excerpt of the implementation of the proposed software architecture, which prepares data for classifications and runs predictions with ML classifiers. Experiment 2 in section 6 shows the impact of human (PHO) intervention on the definition of a classifier and the accuracy of predictions. Section 8 debates the results and concludes.

2. Related work

At the time of writing this paper there were no publications that focus on frameworks and software architectures in a computational setting, for enhancing decision making in the time of covid-19. This might be explainable for two reasons. First, big data technologies could help in such circumstances and they themselves dictate the format and architectures of software solutions, which very often cannot be bespoke. Second, software tools designed for running predictions are freely available, as a single point of computing. They are deployable in Linux and Windows, available to everyone and require no previous knowledge in either statistics or computing. This might seem appealing, but it is extremely dangerous to make decisions on such premises, because it would be impossible to judge the level of accuracy of such computing and know risks associated with it.

Over the last decade, there is evidence that twitter data and sentiment analysis have been used in the management of infectious diseases, due to the unfortunate spread of the Zika and E-bola viruses. The authors of [11] illustrate their sentiment analysis on tweets, related to Ebola, by classifying them into positive, negative and neutral through NLP. Classification of tweets was carried out by comparing them with the "bag of words". Their visualization does give some insight into the impact of the disease on people. Reference [17] illustrates the processing of nearly 60 million tweets that captured the spread of the Ebola and Zika viruses. A topic modelling technique

was used, provided by Mahout, a Hadoop-based ML library. Reference [18] introduces an algorithm for understanding the spread of Ebola, by predicting outbreaks of zoonotic diseases before they appear. The authors claim that they can predict which species are capable of causing future outbreaks, because any given species may be a disease reservoir according to the training data of likely reservoirs. Finally, reference [19] focuses on the search for small molecule inhibitors of the Ebola virus (EBOV) and uses ML for something completely different. They generate Bayesian ML models with viral pseudo-type entry assay and the EBOV replication assay data. Computationally created scores of the MicroSource library of drugs select those likely to be potential inhibitors.

There are a few publications, which influenced the evaluation of sentiment analysis and its practices. The state of the art of opinion mining and sentiment analysis as systemized in [10] is still very much of interest to us. It contains a few of their ideas on ML based analysis, focusing on the preparation of the training data sets. In [20] the authors give a good insight into the optimization of ML based approaches for sentiment analysis, run upon tweets related to the HPV vaccine. Their proposed SVM model is still attractive and should be considered for initial testing of ML precision(s) in studies of a similar nature. In [21] the authors give a rationale for the usefulness of the SVM in Tweet analysis. We also looked at the data collection and preparation processes for running ML from [22, 23].

Over the last 18 months, we have witnessed an explosion of publications using ML and predictions/classifications for assisting in decision making in the pandemics. We are not in a position to review all of these publications, but the topic modelling in tweet sentiment analysis is still prevalent, and these papers are now coming from every corner of the world. A comprehensive review of sentiment analysis and its applications in fighting COVID-19, is in [24].

A classical and comprehensive sentiment analysis of tweets related to worldwide information on the spreading of covid-19 is available in [13] with proposed Naïve Bayes (NB) Hierarchy and the NB probabilistic algorithm that uses Bayes theorem to compute sentiment distribution over the data. In [25] the authors give a detailed explanation of ML classifiers in order to demonstrate insights into the progress of fear-sentiment over time as covid-19 spreads. They use descriptive textual analytics supported by textual data visualizations. It is worth looking at their overview of two ML classification methods, but only in the context of textual analytics. Top concerns of tweeter owners during pandemics are available in [26] and in [27] the authors aimed to detect and characterize user-generated conversations that could be associated with covid-19 related symptoms, using twitter generated data.

There is only one paper, which really attracted our attention. It identifies relevant tweets in real time, which support situation awareness [28]. Obviously, the publication may look irrelevant for twitter sentiment analysis during the pandemic, but it does show specific ways of thinking when we need to react to changes in the content of collected tweets. The authors of [28] are able to determine semantically irrelevant information by retraining classifiers. They improve the classification process, by allowing users to correct iteratively the relevancy of tweets. One of the major reasons why this publication could be of interest to readers of this paper is the level of user involvement in the process of defining classifiers suitable for running ML in various circumstances.

3. The Background and Research Questions

Let us assume that a particular PHO wanted to monitor a situation with the development of the covid-19 spread across the world and include the analysis of incoming tweets to aid their decision making.

Considering that the twitter platform has been recognized as an official intelligence source, for a variety of activities on the Internet, which include malicious use and urgent and live discoveries of potential dangerous exploits of software vulnerability [29], then it is worthwhile to monitor messages on twitter, which could impact PHO decision making during pandemics. Monitoring incoming tweets is rather simple from the computational point of view and requires a lightweight software application, which could also be developed in android and iOS operating environments and thus make an initial assessment of incoming tweets [7,30]. What is important is that the PHO must know exactly which terms should be used in filtering tweets in order to find out if there is information, available on the platform, which might affect their decision making. This is not a difficult task for experts within any PHO.

Inference extraction from tweets can have various formats, but ML and classifications may achieve one important outcome for a PHO and answer: *how likely is it that a particular tweet and its re-tweets contain information, which can have a direct impact on PHO decision making*? Therefore, there should be answers to these questions for incoming tweets:

1) Do they bring new information, knowledge, facts or even opinions, which are of interest to a PHO, and might require investigation?

2) Do they give data/knowledge, which are either of no interest to the PHO, or is an PHO aware of them, and thus they would not need to be investigated further?

These bullets 1) and 2) are ideal for classifying tweets and performing predictive inference, based on ML techniques. 1) and 2) above might also resemble the problem of performing a classical or traditional sentiment analysis upon the collected tweets, with one huge difference and only one potential similarity:

(i) This indeed is a text mining problem and belongs to the world of NLP algorithms, which are to be found in ML classifications and data preparation. They are particularly important in finding sentiment in tweets.

(ii) However, 1) and 2) above do not require to perform a sentiment analysis, because we are not looking for predominant topics in tweets or general sentiment. We are looking at classification of tweets for finding out if something important, for the PHO, is being shared on twitter, which would either affect public health or needs further research / investigation.

Consequently, we should run ML algorithms upon incoming tweets in order to predict if they belong to 1) or 2) above. If we wished to address this, we must summarize what has been outlined in Problem 1-3 from the Introduction. Firstly, acquiring predictive inference in everyday operations of the PHO is not a straight-forward task and does not have standardized techniques, which would guarantee trustworthy and reliable results [15]. These predictions depend on the way the PHO (a) define a training and testing data set, for ML classification and (b) their way of preparing twitter data, to be suitable for a chosen classification. Second, PHOs may not have excessive financial resources to employ human power (*data scientist)* therefore the proposal from this study must address both: the complexity of data science practices in interpreting the meaning of twitter data and the power of human intervention, from the PHO side in order to run adequate ML classifiers.

4. The Proposal

4.1. The Semantic Framework for PHO

Figure 1 is an overview of the proposed Semantic Framework for practising data science in PHO. The word "semantic" is added to increase computational intelligence with reasoning. Thus Semantic technologies are an aid in decision making, which is until now supposed to be based solely on learning and prediction technologies. Figure 1 also allows access to and processing of numerous types of data of various origins, which would be accessible individually to PHO anyway. However, in this framework a software application, built from the proposed framework embraces accessing and processing all data, under the umbrella of one software application, which would be run by the PHO.
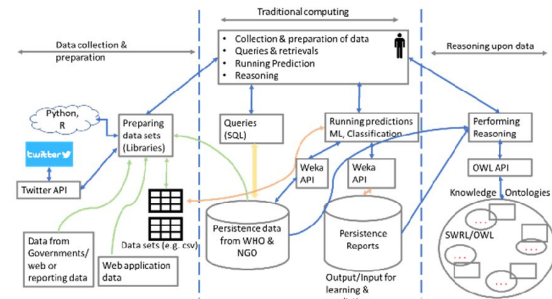

Figure 1. Semantic Framework with Software Architectures for Processing Covid-19 data

The Framework is very close to the notion of a layered and component-based software architecture, which follows separation of concerns and places computational software components in the separate layer to distinguish them between user interfaces and components which store data. Therefore, the figure is self-explanatory, particularly if we take into account that software components are technology specific, and we can distinguish between:

a) traditional SQL queries (the middle of the framework),
b) using Python's libraries [31] to process data sets generated as CSV files from twitter streams (left part)
c) using OWL-API [2,3] for populating ontologies with the data from the persistence (and csv files), and reasoning in feature selection (in future work)

However heterogeneous software components from Figure 1 are, there are numerous plug-ins and APIs which enable data processing and sharing across these technologies. Considering that software architecture generated from the semantic framework is component based and layered, it would be easily deployable within J2E [32] or .NET [33] environments and as such we should have no issues in creating a software application from the proposal. One of the options is to use Integrated Development Environment (IDE) such as NetBenas or Eclipse, set up relevant APIs, check plug-ins and create software components as defined in the Figure 1: UI, Computations and software components which contain relevant data and results of computing.

Considering the different nature of computations between these three vertical parts of the framework, it is easy to conclude that the left part of Fig. 1 deals with data streams from e.g. social media and twitter, potential web data and structured data (in a tabular formats) from governments. All these sources would mostly require the use of libraries which can prepare and process data for further computations. It is very likely that the data will be prepared for running ML and classifications, but

it could also serve as a source of information which can be stored in a traditional database persistence (middle part of Fig.1). The data sets in a tabular format should be updated and it is up to a PHO to decide whether such type of data should find its place as an archive in SQL databases. The same databases may store any other results of computations and particularly ML predictions where this information can be processed further and possibly reused in ML algorithms. The access to ontologies and reasoning upon their concepts is also feasible and could be used for either improving the performance of ML through feature selection/labelling or interpreting the semantic of data in the framework.

There are a few interesting points in Figure 1.

The left part of the framework deals with data streams from social media, but also with web generated data, i.e. data from web applications. Data generated by governments are very often in tabular format, with rows and columns, but also in simple tables with numerical values inserted as a part of data collection. From this perspective, data sets in csv and .xsl formats would dominate, which means that twitter data will fit perfectly well. These data sets are crucial for running predictions, and shared across computations in Fig. 1.

The middle part of the framework is reserved for traditional computing with persistent data repositories which are either generated by the computations in this part (outputs and inputs for ML and predictions) or available from public sources, such as WHO. These are often archives, but if for any reason, we need to store information on data processing before we run ML or store results of ML algorithm accuracy, we should use persistent data repositories for them. Persistent repositories range from traditional MySQL database to spreadsheet documents and OWL ontologies.

The right part of the framework would include reasoning in order to bring more intelligence to PHO decision making, as shown in [1,2]. There are various possibilities of using SWRL enabled OWL ontologies [34]. One example would be to use results of data preparation (csv data sets), definition of the classifiers (such as features) and results of running ML algorithm in terms of their accuracy to populate the ontology and reason upon: *"which classifier and which feature selection would the best choice for a PHO"*. To be more precise, reasoning would be beneficial to determine features of any classifier, and thus the framework outlines which data generated within the framework could be relevant.

4.2. Identifying the Source of Data

In order to collect tweets, a PHO has two options. The first is to use Twitter API which, allows placing criteria for tweet extraction, but requires registering and opening an account and facing a few Twitter API limitations. The other option is to use TWINT, the Twitter scraping tool written in Python, which allows scraping Tweets from Twitter profiles without using Twitter's API [35]. By utilizing Twitter's search operators, TWINT would scrape Tweets from specific users, scrape Tweets relating to topics, hashtags & trends and issue queries to Twitter data allowing to scrape user's followers, Tweets the user likes(d), and who they follow. It requires no authentication, API, or browser emulation.

In this study the twitter data set was downloaded in the period between 2020-01-01 and 2021-08-28 using two methods. It started with the collection of tweets every day, which resulted in data set exceeding 1,000.000+ tweets in a very short period of time (estimated min 1000 tweets per day). The option of collecting tweets every four days proved to be more practical. It created a more manageable data set which could be processed more efficiently, might not need extra hardware power and could be more suitable for manual inspection, if required. The specified language was English because we wanted to experiment with both: complete automation in tweets analysis based on the algorithms from the NLP (and thus English language was imperative) and using human intervention in determining the sentiment and labeling of tweets. The keyword for tweet extraction was **covid** and for each tweet we collected *date, language, username, tweet, hashtags, nlikes, nretweets*. Total no of tweet is 51455.

The initial inspection of the content of the data set, supported by the count of most frequently used words in it, revealed something interesting: there are numerous "hashtags" in bodies of these tweets. The first 12 most frequently used hashtags are #covid (in 1200+ tweets), #cvoid-19 (900+), #coronavirus (200), # insiders, #getvaccinated, #vaccine, #smartnews, #newsbreak, #waremask, #deltavariant, #getvacinatednow, #newnormal (all of them appearing in 150+tweets). This information would be very important for a PHO: It shows various threads of Twitter debates on the pandemics and the level of interest in mentioning topics, such as "delta-variant" and "vaccination" in particular.

4.3. Data Set Cleaning and Preparation

Regardless of tools and techniques which can be used in tweet classifications, one of the most important tasks is to remove stop-words. The "nltk" package in Python has a list of stopwords stored in 16 different languages and it is very convenient to use it. Each tweet has been tokenized and then each token is checked with the stopwords list: only those tokens, which are not included in this stopword list, are not removed. At the end tokens are joined for each tweet. The further

cleaning is quite standardized and required if we wanted to use Python libraries. Therefore we have replaced hyphen, removed user names, hashtags, URLs, non-ASCII characters, punctuations, numbers and special characters, plus convert the texts into lowercase.

5. Experiment 1

Experiment 1 uses automation available for defining the sentiment from NLP algorithms. Its results are dependent on predefined steps of the collected and cleaned tweet analysis. Therefore, after the cleaning process, it is useful to identify the most popular tweeted unigrams related to COVID-19 (with ngrams procedure) and thus the most frequent words are shown in cloud of words (Figure 2) and in the bar chart from Figure 3.

Figure 2. Cloud of words according to the ngrams procedure with n set to 1.

It is also easy to find the most popular bigrams (pair of words) related to COVID-19 (Figure 4).

5.1. Running Sentiment Analysis

Before applying sentiment analysis, we have to determine positive, negative and neutral tweets. To do so, we considered the cleaned tweets with the stemming technique. We then apply VADER (Valence Aware Dictionary for Sentiment Reasoning) sentiment analyzer to determine tweets that can be considered to create positive, negative and neutral sentiments. However, the Vader lexicon, often used for text sentiment analysis is sensitive to both polarity (positive/negative) and intensity (strength) of emotion. It is available in the NLTK package and can be applied directly to unlabeled text data. However, VADER sentimental analysis relies on a dictionary that maps lexical features to emotion intensities known as sentiment scores. For each cleaned tweet the polarity is computed. When polarity is greater than ⅓ the tweet is labelled with positive sentiment. When the polarity is lower than ⅓ the tweet is labelled with negative sentiment. Otherwise it is set to neutral.

We then apply the Vader's sentiment analyzer to determine tweets sentiment, as given in Figure 5.

Figure 3 The words vs frequencies

Figure 4. Cloud of words according to the bgrams procedure.

Figure 5 Initial sentiments distribution

Figure 6. Sentiment distribution ("neutral" omitted)

Vader's sentiment analyser is pretty cautious, therefore it is unlikely that it will take extremely polarized values, which are strikingly visible in Figure 5. The percentage of tweets categorised as neutral is rather high. By removing the 'neutral' label we let the sentiments to be either 'positive' or 'negative', as shown in Figure 6.

With a polarized labelling which doesn't consider the neutral option the difference between negative and positive tweets is more clear.

At this point we can run ML algorithms to measure if we can classify future tweets into positive and negative. Multinomial Naïve Bayse, SVM and Logistic Regression are chosen, with precision, recall, and f1-score as ML metrics. It is important to note, that there is no access to feature selection. If we wished to understand the definition of the classifiers and the details of features, we can only use functions within Python libraries such as **collections.Counter()**. It reveals that there are 758322 words available in the datasets (no uniqueness) and the number of features used in the experiment 1 is 29301.

ML Algorithm	Avg Precision	Avg Recall	Avg F1-score
Multinomial Naïve Bayse	0.81	0.82	0.82
SVM SGDClassifier	0.85	0.85	0.86
Logistic Regression	0.86	0.88	0.89

Table 1: Results of running ML algorithms in Experiment 1

6. Experiment 2

In this experiment we used the same data set, but did not perform additional sentiment analysis, because we wanted to eliminate the dependence on NLP algorithms in the text analysis when defining classifiers and selecting features. Therefore, on the data set available from experiment 1, we run the same set of classifiers, upon a different selection of features. Initially, the choice of features was dependent on the most frequently used words, as available in Figure 3.

	Avg Precision	Avg Recall	Avg F1-score
Multinomial Naïve Bayse	0.93	0.92	0.92
SVM SGDClassifier	0.92	0.92	0.91
Logistic Regression	0.92	0.93	0.93

Table 2: ML results for the first set of features

When we use this set of features: **vaccine, death, hospital, variant, strain, ventilator, health, spread, positive, patient, kid, vulnerable, test** to label relevant tweets (3089 relevant and 28366 non relevant) and apply the same classifiers as in Experiment 1, the ML results, given in Table 2, show improvements.

This feature selection *vaccine, decease, pharmaceutic, stomach, pfizer, moderna, astrazeneca, transmiss, symptom, medicinal, hospital, death, variant, strain, ventilator, health, spread, positive, patient, immune, restriction, kid, child, vulnerable, test,* with the number of relevant/irrelevant tweets 25301/ 26154 renders results, for the same algorithms, as given in Table 3. SVM results do not improve (the others are not affected)

	Avg Precision	Avg Recall	Avg F1-score
Multinomial Naïve Bayse	0.93	0.92	0.92
SVM SGDClassifier	0.90	0.90	0.89
Logistic Regression	0.92	0.93	0.93

Table 3: ML results for the second set of features

This feature selection *'disease', 'symptoms, 'pharmaceutic', 'vaccine', 'pfizer', 'moderna', 'astrazeneca', 'side effects', 'transmission', 'medicinal', 'hospital', 'death', 'variant', 'strain', 'ventilator', 'health', 'spread', 'positive', 'patient', 'immune', 'restriction', 'kid','child', 'vulnerable', 'blood vessels', 'receptor', 'smell', 'digestion'*, gives the number of relevant/irrelevant tweets 20734/30721, but the result of algorithms precision, given in Table 4 improves by 0.02 for SVM.

ML Algorithm	Avg Precision	Avg Recall	Avg F1-score
Multinomial Naïve Bayse	0.93	0.93	0.92
SVM SGDClassifier	0.91	0.91	0.90
Logistic Regression	0.93	0.94	0.92

Table 4: ML Results for the third set of features

7. Excerpts from the Implementation

Figure 2 illustrates the excerpt from the implementation of the proposed framework. In this study we ran two out of four functionalities: *Preparing Data Set* (Tweets) and *Running Predictions* by preparing a classifier and running ML. This implementation allowed us to prepare data sets using Twitter API and then prepare the set fully supported by Python and its libraries. However, the use of TWINT, as suggested in section 4.2 does not change the architectural model, because it is a Twitter scraping tool working with Twitter streams. The preparation for the classifier, which included finding its features (by

analyzing the prepared data set) took two different pathways and thus we ran two different experiments on the prepared data set.
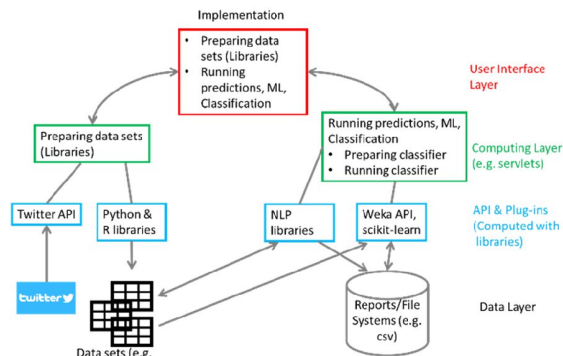

Figure 7. Excerpts from the Implementation

Figure 7 has been implemented within NetBeans IDE, we used JSP and servlets in the User Interface and Computing layer. However, API and plug-ins are sitting below them. They address particular needs of experiments: we either run computations with libraries (NLP, sickit-learn) or use OWL-API to access and manipulate ontologies (e.g. knowledge collected though experiments). Therefore, servlets from Figure 7 are rather simple because they manage tasks of our experiments, but we have a choice of libraries which were done "as we go" (not fully automated yet).

The first experiment involved the NLP processing libraries, which used their algorithms for the automatic preparation of the data set and labelling for classification, without human intervention. Figures 2-6 are results of such automation. They give visual information to humans. Consequently, decision on feature selection is automatic: humans have no final say in the decision related to tweet classification. This part of implementation was trivial.

The second experiment involves human intervention. We used the same data set from experiment 1 and did NOT further clean the data. The purpose was to experiment with feature selection by entering features through the user interface and accessing sickit-learn libraries. This means feature selection was dictated by humans (health experts from the PHO).

Our prototype is not commercially viable because it is there just to prove the concept. However, it shows that is it relatively simple to create a full scale application from Figure 7 where we utilize a synergy of traditional computations (with programming languages) with ready made libraries (from Python to sickit-learn), plug-ins if needed and API (accessing twitter streams and OWL ontologies).

Therefore experiment 2 involves PHO decision making on the definition of the classifier. Even if the PHO wishes simply to change the set of features, which were not detected with any of the NLP algorithms, and nothing else, it will be able to look at the results of classification and judge, from the professional point of view which classifier would be acceptable.

In this implementation, we have not experimented with a data set which was not "cleaned" with NLP libraries The semantic behind URLs and stop-words does not contribute too much to extracting the meaning from Tweets, without carrying out further computations. The final outcome of preparing the data source was placed in a .CSV file (left bottom part of Figures 1and 2).

The experiments used *pandas, matplotlib, nltk, collections, wordcloud, re, datetime, sklearn.linear_model* and our programming of servlets in Java was minimal. We did not to use the WEKA tool (not excluded in future)..

8. Results, Discussion and Conclusions

The study focuses on an operational environment suitable for PHO domain experts in order to perform the text analysis of collected tweets and make their own judgement if "*it is likely that a particular tweet and its re-tweets contain information, which can have a direct impact on the PHO decision making*. This means that the study does not embark on the analysis of the efficiency of any of these ML algorithms and levels of predictions. However, the results of our experiments are interesting. Experiment 1, which uses an automated approach to text preparation and sentiment analysis is a disappointment for many reasons:

a) NLP and the calculation of sentiments based on it divide tweets into positive and negative (pos/neg). In the time of any pandemic it would be impossible to explain what would pos/neg mean when people are losing lives. What would *pos* sentiment mean to PHO? Would it be better if a PHO defines a sentiment and moved away from pos/neg if sentiment analysis is still needed?

b) NLP algorithms with the analysis of data given in Figures 2-4 are interesting, but they do not show exactly how the content of the words in these clouds are being used in further processing.

c) Figure 5 is interesting, but it shows that we have an enormous amount of information in our data set which can not be defined as pos/neg. Should not our sentiment analysis STOP immediately at this stage? How reliable is Figure 6 where "neutral" sentiment is removed? Does it affect data processing?

d) No access to feature selection is a norm here, but the number of features, discovered with a library function is excessive. Do PHOs know the method of selecting the words in the data set into features?

In experiment 1, the precision of our simple ML algorithms, run upon the data set, is relatively low. We could blame the issues listed in a)-d) above, but until we get the answer to the questions above, we do not know if this classification is expected, normal, or completely wrong. It does not guarantee that PHOs will be able to classify incoming tweets successfully in future.

Experiment 2 is more promising. It is simply based on the premises that

- Humans define features by inserting the words which PHO finds RELEVANT for classification of the tweets (we are not interested in pos/neg)
- ML algorithms are run repetitively with different feature selection and their precision monitored.

The initial feature selection did follow the first 10+ most frequently used words in the data set. The list was changed by adding more words, misspelling some them and adding words which we have not found in the most frequently used words set. The precision of the same algorithms is far better than in experiment 1 PHOs are able to talk about RELEVANT and IRRELEVANT tweets and it is up to any PHO to decide how to play with feature selection. In this pandemic, the words of interest for PHO changes almost on a daily basis and the automation in tweet classification is not the answer for efficient decision making. With all respect to the NLP domain, in this study, NLP did not secure successful tweet classification for anyone.

The experiments described in this study are run by computer scientists, but their level of complexity is rather low and by no means could they present an obstacle to PHO domain experts. For monitoring tweets in real time, there are very light-weight software applications available, which can run on Android or iOS. For preparing a data set, there is an easily accessible Python ready-made function, which can easily be applied to the data set, the way PHO requires. Finally, the feature selection is a simple job of carrying out a debate between PHO domain experts and inspecting the set of most frequently used words in their data set, plus entering them into the application. The application from Figure 7 safeguards PHO activities and its implementation is rather trivial for software engineers. The basic knowledge of statistics and NLP is not a prerequisite for exploring text processing as described in Experiment 2. The most difficult job is to know exactly, how the feature selection in Figure 7 would secure answers to questions 1) and 2) from the introduction. We have proved that it is feasible to classify tweets successfully by following potential PHO rules on text processing applied to the data set.

The authors believe that if a software application, which deploys the ideas from this study (including user's involvement in feature selection) existed from the first moment when tweets on a "new virus" started appearing on Twitter in late 2019, PHOs all over the world could have grasped more relevant information much earlier than it appeared in the media. If we started "listening to tweets" in December 2019, when the first CNN news reader announced the existence of the new virus in China, we could all have had a different picture on "what is happening" which in turn would affect the decision on "what to do". If in the cyber security field we constantly listen to tweets [12], what is stopping us from listening to tweets for finding more about a potentially dangerous virus? Constant changes in feature selection, as shown in experiment 2 are a powerful mechanisms which every PHO can exercise. This is a very safe and easy way of monitoring what is relevant on twitter and what is not. PHO would not need an expensive data scientist for any of these tasks/epxeriments.

Our future work is centered on A) implementing a full scale software application from Figure 1 and B) collecting knowledge from feature selection and the precision of ML algorithms and deploying reasoning for deciding the best possible way of classifying incoming information on live data streams.

9. References

1. Tarabi, M., Juric, R. (2018) Software Architectures For Smart Applications In The Management Of Chronic Diseases: A Case Study Of Reversibility Of Diabetes 2, In Proceedings of the 51st HICSS Conference 2018

2. Che-Fu, C., Juric, R. Kim, I. (2021) Software Architectures and Efficient Data Sharing for Promoting Continuous Drug Re-purposing, In Proceedings of the 54th HICSS Conference, Jan 2021

3. Shojanoori, R., Juric (2015) ASECS: Assistive Self-Care Software Architectures For Delivering Service In Care Homes , In Proceedings of the 48th HICSS Conference, HI, US, Jan. 2015

4. Juric, R., Kim, I, Tesanovic, I., Sinkovic, A. (2014) Raising Awareness of Dental Pain from Tweets, in Proceedings of the SDPS 2014 Conference, Kuching, Malaysia, June 2014.

5. Juric, R., Kim, I. (2015) Can Twitter Transform Communities Affected By E-Bola, In Proceedings of the 20th International SDPS Conference, November 2015

6. Juric, R., Kim, I., Panneerselvam, H., Tesanovic, I. (2017) Analysis Of Zika Virus Tweets: Could Hadoop Platform Help In Global Health Management, In Proceedings of the 50th HICSS Conference, Jan. 2017

7. Juric, R., Kim, I. (2017) Software Architectures For Smart Applications Which Merge Ontological Reasoning With Big Data Analytics, In Proceedings of the 22nd SDPS 2017 conference November 2017

8. Shamoug, A., Juric, R., Paurobally, S. (2012) Ontological Reasoning as a Tool for Humanitarian Decision Making, in Proceedings of the 9th International ISCRAM conf. Canada, April 2012.

9. Pettai, K., Juric, R. Arntzen-Bechina, A (2015) The Power Of Microblogging In Disseminating Information In Humanitarian Crises: A Study Of Nepalese Earthquake, In Proceedings of the 20th Intern. SDPS Conference, November 2015

10. Joylin, B, Suma, A. T. P. And Victor, N., "Sentiment Analysis On Ebola Outbreak Using Twitter Data," Int. J. Pharm. Technol., 8(4), 2016, Pp. 25835–25842

11.] Vorovchenko, T., Ariana, P., F., van Loggerenberg, and Amirian, P., "#Ebola and Twitter. What Insights Can Global Health Draw from Social Media?", Big data in healthcare. Springer briefs in pharmaceutical science & drug development ed. Springer, 2017.

12.] Vorovchenko, T., Ariana, P., F., van Loggerenberg, and Amirian, P., "#Ebola and Twitter. What Insights Can Global Health Draw from Social Media?", Big data in healthcare. Springer briefs in pharmaceutical science & drug development ed. Springer, 2017.

13. Samuel, J.,Ali, G.G.M.N.,Rahman, M.M., Esawi, E. and Samuel, Y. "COVID-19 Public Sentiment Insights And Machine Learning For Tweets Classification", Information, 11(6), 2020

14. Ronchieri E., Juric, R. (2018) The Efficiency of Machine Learning in Finding the Sentiment in Tweets: A Study of Software Vulnerability, In Proceedings of the SDPS 2018 Workshop on Accountability of AI, Italy, Dec 2018.

15. Ronchieri E., Juric, R. Canaparo, M. (2019) Sentiment Analysis for Software Code Assessment, in 2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), Manchester, UK.

16. R. Juric, N. Danylchanka (2020) Reliability of Training Data Sets for ML Classifiers: a Lesson Learned from Mechanical Engineering, in HICSS 53 Proceedings, January 2020.

17. Moise, I., (2016) "The Technical Hashtag in Twitter Data: a Hadoop Experience," in IEEE Proc. of Big Data

18. Han, B., "The Algorithm that's Huntig Ebola," IEEE Spectrum, 2015.

19. Ekins, S., Freundlich, J., Clark, A. M., Anantpadma, M., Davey, R. A., and Madrid, P., "Machine Learning Models Identify Molecules Active Against the Ebola Virus in Virtro," in F1000 Res., 2015.

20.]Du, J., Xu, J., Song, H., Liu, X. and Tao, C., "Optimization on ML based approaches sentiment analysis on HPV vaccines related tweets," J. Biomed. Semantics, 8(9), 2017.

21. Ahmad, M., Aftab, S., Muhammad, S. S., and Ahmad, S., "Sentiment Analysis of Tweets Using SVM," Int. J. Comput. Appl., 7(5), 2017.

22. Ghenai, A. and Mejova, Y., "Catching Zika Fever: Application of Crowd sourcing and machine Learning for Tracking Health Misinformation on Twitter," arXiv, 2017.

23. Melo, C. F. O. R., Navarro, L. C., de Oliveira, D. N., Guerreiro, T. M., Lima, E. O., Delafiori , J. et al., "A Machine Learning Application Based in Random Forest for Integrating Mass Spectrometry-Based Metabolomic Data: A Simple Screening Method for Patients With Zika Virus,"Front.Bioeng. Biotechnol., 6(31), 2018.

24. Alamoodi, A.H., Zaidan, B.B.., Zaidan, A.A., Albahri, O.S. , Mohammed, K.I., Malik, R.Q. Almahdi, E.M. Chyad, M.A. , Tareq, Z,.Albahri, A.S., Hameed, H., Alaa, M. (2021) Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review, in Exp.Sys. With Appl. 167 (2021)

25. Manguri, K. H., Ramadhan, R. N. and Mohammed Amin, P. R., "Twitter Sentiment Analysis on Worldwide COVID-19 Outbreaks",Kurdistan Journal of Applied Research, 5(3), May 20, pp. 54-65,

26. [28] Mackey, T., Purushothaman, V., Li J., et al. Machine Learning to Detect Self-Reporting of Symptoms, Testing Access, and Recovery Associated With COVID-19 on Twitter: Retrospective Big Data Info Surveillance Study. JMIR Public Health Surveill. 6(2):e19509., 2020

27. Abd-Alrazaq1, A., Alhuwail, D., Househ1, M., Hamdi1, M. and Shah, Z. "Top Concerns of Tweeters During the COVID-19 Pandemic: Infoveillance Study", J Med Internet Res Apr 21;22(4):e19016, 2020, doi: 10.2196/19016. Published online at https://www.jmir.org/2020/4/e19016/

28. Snyder LS, Lin YS, Karimzadeh M, Goldwasser D, Ebert DS. Interactive Learning for Identifying Relevant Tweets to Support Real-time Situational Awareness. IEEE Trans Vis Comput Graph. 2020 Jan;26(1):558-568

29. Juric, R., McClenaghan, K. M., Moholoth, O.C.(2019) Detecting Cyber Security Vulnerabilities through Reactive Programming, in Proceedings of the 52nd HICSS, January 2019

30. Juric, R., McClenaghan, KM, Moholoth, OC (2018) Analyzing Software Vulnerabilities with Reactive Programming, In Proceedings of the SDPS 2018 Workshop on Accountability of AI, Italy, Nov 2018

31. S. Thivaharan., G. Srivatsun. and S. Sarathambekai., "A Survey on Python Libraries Used for Social Media Content Scraping," *2020 International Conference on Smart Electronics and Communication (ICOSEC)*, 2020, pp. 361-366

32. J2E Technology https://www.oracle.com/java/technologies/appmodel.html

33. .NET Framework https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet

34. Almami, E., Juric, R., Ahmed, M.Z. (2016) Software Applications Built Upon SWRL Enabled OWL Ontologies, In Pros. of the 21st SDPS 2016 conf. Florida, December 2016.

35. Zacharias, C.; Poldi, F. TWINT - Twitter Intelligence Tool, 2017 - 2020. https://github.com/twintproject/twint

36. Li, J. and Liu, H., "Challenges of Feature Selection for Big Data Analytics,"IEEE Intell. Syst., 32(2), 2017, pp. 9–15.

37. Rustam F, Khalid M, Aslam W, Rupapara, V, Mehmood A, Choi GS (2021) A performance comparison of supervised machine learning models for Covid-19 tweets sentiment analysis. PLoS ONE 16(2): e0245909.