APPLICATION OF CRAWLING TECHNIQUES AND NATURAL LANGUAGE PROCESSING FOR THE ANALYSIS OF INFORMATION EXTRACTED FROM THE SOCIAL NETWORK TWITTER

[1]MARCO JAVIER SUAREZ BARON, [2]JOHAN DAVID DIAZ MENDIVELSO, [3]OSCAR GARCIA-CABREJO

[1]Associate Professor, Pedagogical and Technological University of Colombia
[2]Researcher, Pedagogical and Technological University of Colombia,
[3]Assistant Professor, Pedagogical and Technological University of Colombia
E-mail: [1]marco.suarez@uptc.edu.co, [2]johan.diaz@uptc.edu.co, [3]oscar.garcia04@uptc.edu.co

Abstract - This work presents the development of a computational model driven to analyzeinformation obtained from Twitter social network. The method used to extract comments from Twitter is bases on the Rest and Streaming APIs and a web scraping tool. These comments are indexed and sent to a non-relational database that handles large volumes of information. The ability to extract information from the tools was verified by creating a corpus with Twitter comments for social analysis. It is concluded that it is possible to extract information from tweets by applying tools designed for crawling and natural language processing NLP.

Keywords - Crawling, Information extracting NLP, Twits.

I. INTRODUCTION

In recent years it is inevitable to notice within the different organizations, societies or people the need to analyze different points of view presented in virtual conversations, which can occur within social networks, chats, forums, blogs, etc. [one]. The main economic, social, cultural, political sectors, among others; theseare a clear example that it is necessary to create more tools that allow the interpretation of all the information that circulates daily in the different internet sites [2].

The aforementioned need encourages to continue research based on different models exposed; The application developed and exposed in this article is the continuation and implementation of a model based on natural language processing implemented in the article "Social analysis applying natural language techniques to information extracted from Twitter" [3].In the article in question the articulation of different algorithms for manipulating and refining information together with NLTK (Natural Language Toolkit) libraries is presented, which helps to extract information from Twitter and achieve labeling by applying labeling techniques (tokenization); Models like the previous one are a substantial advance for the need to be covered, but it is necessary to create or articulate a more interactive model that manages to cover different needs not previously considered.

Due to the above, the development of a web application is presented that helps to implement the mentioned model in an architecture suitable for the use of a more general public, who wishes to analyze text based on key topics or interests, which can help with different models. visual or with reports that can be exported for the different users of the tool.

II. MATERIALS AND METHODS

For the development of a web tool based on algorithms and models already developed in previousresearch, it is necessary to first study the operation or step by step of the algorithms in order to create a precise and suitable architecture for communication between the different models [4].After the detection and technical analysis, a development methodology is proposed that helps to unify everything found and with this, successfully complete the development of the solution and test it in controlled environments with particular cases and related to end users [5].

Next, each of the stages necessary for the development of the solution is presented:

A. Initial Model

Previous research left behind models, tools and algorithms that helped find solutions for the analysis of text extracted from social networks [1], which is why the results given by the research presented in [2] were taken, to implement them within a tool. more interactive for different users.

When carrying out the analysis of the research already presented, a somewhat punctual development and of purely investigative use is found, due to the fact that the algorithms created by the authors and the implementation of the NLTK libraries [6], are used in low-level environments, what is necessary is its migration to a paradigm more oriented to a general user and not specific who has experience with the tool.

From a technical perspective, the algorithms of extraction, refinement, lemmatization, morphological

structuring, semantics [6]. This algorithmsare implemented in the Python language, using libraries of the same language and also external libraries such as those of the NLTK, with this it generates the need to create an architecture that is compatible with this language in mention.It is also recommended that the development pattern and methodologies is coupled to the needs required for the migration of all algorithms already implemented [7].

Faced with the migration of the algorithms, it is necessary that each one of the files used be reviewed and ordered respecting a model-view-controller design pattern, because it is a pattern that complies with a service-oriented model [8] and it is a most appropriate way to develop a web application that meets all quality standards, information security and that can be deployed or published for a general public.

It should be noted in this section that the same authors of the algorithms taken were the same ones who made modifications and updates to the real source codes.

B. Technological Architecture

The model taken from the research mentioned above, gives the starting point for the structuring of the tool and the technological architecture necessary for the construction of the desired solution. With what has been analyzed, it is detected that the most appropriate model is a model-view-controller design pattern as presented in Figure 1, which helps to properly organize each of the models taken from the detected algorithms and achieves the creation of services that They can be exposed by API-REST techniques and thus communicate the user part or a graphic part with each of the exposed functionalities.

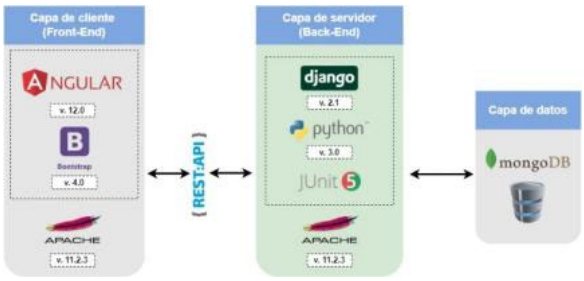


Figure 1. Technological architecture proposed

In the graphic part or the user part there are 2 frameworks, Bootstrap for the control of graphic styles or CSS and JavaScript that are handled within the application; and Angular that help control the flow of the application between each of its screens and also to consume and connect with the exposed services. In the second layer is the Back-End, which is programmed with the Django framework which allows coding each of the services necessary to fulfill each functionality found within the algorithms.This framework also helps to correctly implement the model-view-controller pattern. Finally, there is the data stage, which has a direct connection with the services and exposes a data persistence based on NoSQL, which will allow handling different results with different structures. It should be noted that each of the connections made within each layer are completely decoupled, and thus guarantee deployments on different servers that help maintain the stability of the application when overload or heavy processes are detected in any of the parts of the architecture.

C. Development Methodology

For the correct development of the web application, the SCRUM methodology is implemented, which helps to make a first approach with the real needs that exist within some organizations or within the scientific community, with this, to create a precise tool with a clear goal [9]

When establishing the needs, organized activities are proposed that guide the development of the software and the structuring of the proposed architecture, all within local environments that help to execute tests in controlled environments and different data can be visualized from each of the steps or processes executed. Within the algorithms or applying test models such as white box; step-by-step test execution ensures the correct validation of the application and creates guarantees for future users of the platform.

The implementation of the SCRUM methodology also guarantees that each of the aforementioned standards are met with respect to software development, as well as for each of the validations given in quality, security and usability factors that guarantee the proper use of the platform. in a productive environment.

III. RESULTS AND DISCUSSIONS

As a final result, the implementation of each of the models and the use of each of the materials or artifacts for the development of the web platform is presented, findings that were found in each of the development stages and how they influence the use of the platform within real and functional environments.

A. Architecture Implementation

Initially, each of the exposed algorithms is taken and separated in order to identify each model necessary to guarantee the persistence of the information and also the manipulation of the data generated by each of the processes, when executing this migration from a structured programming to an Object-oriented programming. At the end of this migration process, the objects or models shown in the diagram of figure 2 are defined.A tuple object is shown, which helps to persist within the database and correctly manipulate each tuple found within each analysis.

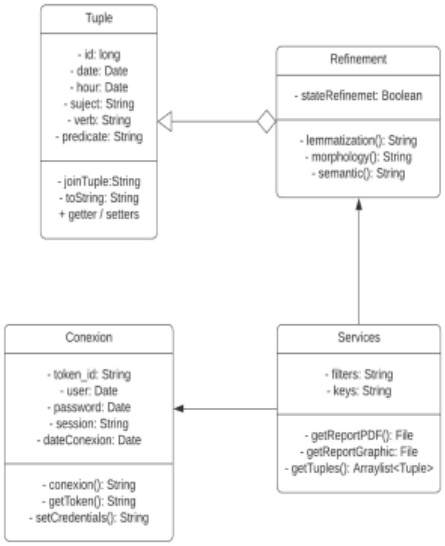


Figure 2. Model class diagram

For the Twitter information extraction model, a class is proposed that contains each of the methods and processes necessary for the connection with the social network.This helps to control possible errors with the connection and can also be sent in a more specific way by the users the topics of interest, topics or filters by which the users want to extract the information. This fulfills one of the main objectives of the tool, since it was sought that the users could filter in a more efficient and precise way all the information that was given in the first versions of the implementation of the models.

For the control of the refinement and text processing algorithms, a management model is created which will help organize each of the extracted and formalized methods in the developed source codes or implementation of the NLTK libraries.All this to have as the last file o source resource a class that helps as a controller of the services that will be created for the consultation of the same by the users from the web model that will be shown to the users.

By having the models defined, two separate projects are created, one that independently controls all the requests by the user and another that will expose the services from an application server so that they can be used by the graphic part. Specific connections are also made with the application manager, MongoDB data, where the data corresponding to the queries made to the social network will be stored.This is useful to create a source for the generation of reports or graphic creation that help to visualize and analyze the information.

B. Tool Development

Each of the projects recommended by the proposed architecture are deployed on local application servers in fully controlled machines for greater monitoring and control during the development of the web application and the execution of the corresponding tests.

Initially, each of the data models to be obtained is programmed. Then a revision of the versions of each of the libraries used in the proposed models is carried out. These are updated to current and stable versions and source codes that are obsolete are updated. This helps the tool to be cross-browser compatible and stable for a longer time. By having the proposed libraries and models updated, development continues to separate the connection configuration with Twitter and the extraction methods, establishing parameterized models that help make them more dynamic for queries. The natural language refinement and processing algorithms are also processed in order to define them in a more parameterizable, stable, and reliable way. When they go through this process, they are moved to a special file where each tuple will go through its respective text processing.

As a last step in the server part or system functions, the controller file was created which has each of the endpoints, by which the created services will be called. Among them is the sending service of each one of the filters and keywords sent for the execution of the information extraction or also the services that will function as a call to export a report or execution of a graph that interests a user.

On the other hand, the proposed methodology allows the development of applications of this type and that comply with a controller view model, to be developed in a way that separates the graphic part, which was developed based on prototypes designed with specialized platforms as shown the figure 3.



Figure 3. Main screen of the tool developed for text analysis

Prototype-guided design helps to find a more precise solution and without wasting a lot of time on web development [Reference prototype methodology], this prototype design was taken to the development of each of the screens in the chosen and mentioned technologies (Bootstrap + Angular), each of the screens were programmed and improved according to the needs that also arose during the development of the services.

To finalize the development of the tool, each of the services was integrated with the graphic part, where

each of the actions executed by the client within the tool points to a service exposed with each of the processes required by the models. of text processing coupled to the solution.

C. Reports and analysis modules

It should be noted that the tool developed has a module which will help users to have historical information searched for and used within the tool:This helps to generate specific reports of the tuples that were found within the extraction of Twitter information. These reports or structured information will help users to have a better handling of it and that they can use it within other tools if they see it necessary. The information can be exported according to the desired report and in formats such as PDF, XMLS, txt or Json. A clear example can be the word cloud report with the most matches within the analyzed tweets, as shown in Figure 4.

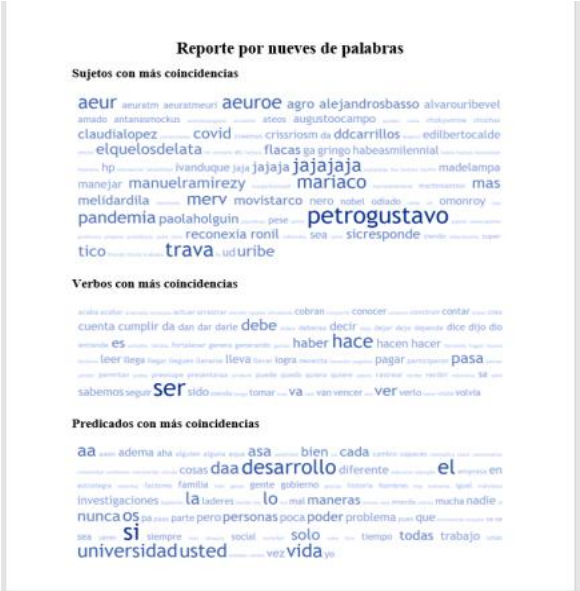


Figure 4. Document in PDF format with the word cloud report.

If users do not want to export the information and manipulate it independently, the tool has a module that will show graphs that help visualize data such as the number of tuples found according to the time of day, according to the typical or keyword, among others.The previously mentioned characteristicsare included in order to help users by generating a real-time display of the information that has been exported or that has been found up to the time of the query.

These report modules substantially help users to have a key input for decision making or function as dashboard pieces that work as data visualization tools within the different types of dissemination, whether academic or professional.

D. Tests on the tool

The execution of the tests is an essential part because it is the stage that reviews the step-by-step execution of the system, from the extraction of the tweets themselves, to the creation of the respective reports.The tests were carried out by creating messages within each of the processes to show each step of its execution.For example, it can be seen that the last extracted tweets are shown in the server console, which helps to compare some results within the servers (Figure 5).At the same time flat files are generated with all the results in order to generate specific filters and confirm that the information displayed within the platform is correct.

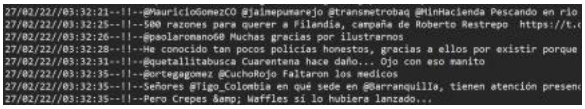


Image 5. Tweet extracted.

It should be noted that examples of white box or black box tests such as those shown above were executed continuously throughout the system analysis process, where it was verified that the results complied with each of the filters proposed by the user.

E. Future work

The tool is proposed as a complete solution for the different users who wish to carry out an analysis of the information that circulates on the social network Twitter. At this moment it has been tested and executed on local environments.The next step is to create a production environment which has public links and user modules which provide privacy for each one and queries can be differentiated.

It is also important that after the deployment to production and use of the first users, surveys and analysis of the use of the application are carried out, to see how viable the integration of extracting information from other social networks is.This is done so that the tool increasingly can contribute to the different challenges presented within organizations or people.

IV. CONCLUSIONS

The development of tools such as the one exposed in this article are necessary due to the great increase in the circulation of information on the internet and social networks.It is also notorious that more and more internet users increase, as could be seen in the last years of the pandemic,andlarge number of people has been forced to be part of all these networks of information and comments.

Algorithms for processing text or other information such as voice or video have boomed in recent years, which sees the need to always continue updating all research carried out in order to implement current models that help cover faster the needs of users.The publication of tools and the incentive to use them,

supports the real dissemination of knowledge by creating information networks that can help improve refinement processes and, at the same time, the development of new tools that cover other types of needs, but based on the same development model proposed in this article.

REFERENCES

[1] ZUCCO, Chiara, et al. Sentiment analysis for mining texts and social networks data: Methods and tools. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2020, Vol. 10, not 1, p. e1333.

[2] ÖBERG, Fredrik; OHMAN, Gustav. Improving Decision Making Through the Use of BI&A and a Data-Driven Culture. 2020.

[3] MENDIVELSO, Johan David Diaz; BARON, Marco Javier Suarez. Social analysis applying natural language techniques to information extracted from Twitter. Science and technology, 2019, vol. 24, no 3, p. 496-503.

[4] GRUHN, Volker; STRIEMER, Rudiger. The Essence of Software Engineering. SpringerNature, 2018.

[5] FARHAN, AR Shehab; MOSTAFA, GM Mostafa. A methodology for enhancing software security during development processes. In 2018 21st Saudi Computer Society National Computer Conference (NCC). IEEE, 2018. p. 1-6.

[6] MENDIVELSO, Johan David Díaz, et al. Advances and challenges of computational methods and models applied to the analysis of information in social networks. Links magazine, 2019, vol. 16, no 2, p. 298-309.

[7] WASHIZAKI, Hironori, et al. Studying software engineering patterns for designing machine learning systems. In 2019 10th International Workshop on Empirical Software Engineering in Practice (IWESEP). IEEE, 2019. p. 49-495.

[8] GORE, Himanshu, et al. Django: Web Development Simple & Fast. Annals of the Romanian Society for Cell Biology, 2021, vol. 25, no 6, p. 4576-4585.

[9] SUBIH, Muhammad Asaad, et al. Comparison of agile method and scrum method with software quality affecting factors. Int. J. Adv. Comput. Sci.Appl, 2019, vol. 10, not 5, p. 531-535.

★★★