Automatic Analysis of Facebook Posts and Comments Written in Brazilian Portuguese

Juan Manuel Adán Coello, Bruno Augusto Junqueira

Brazil

Abstract

Social networks and media are becoming increasingly important sources for knowing people's opinions and sentiments on a wide variety of topics. The huge number of messages published daily in these media makes it impractical to analyze them without the help of natural language processing systems.This article presents an approach to cluster texts by similarity and identifying the sentiments expressed by comments on then (positive, negative and neutral, among others) in an integrated manner. Unlike most of the available studies that focus on the English language and use Twitter as a data source, we treat Brazilian Portuguese posts and comments published on Facebook. The proposed approach employs an unsupervised learning algorithm to group posts and a supervised algorithm to identify the sentiments expressed in comments to posts. In an experimental evaluation, a system that implements the proposed approach showed similar accuracy to that of human evaluators in the tasks of clustering and sentiment analysis, but performed the tasks in much less time.

Keywords: Social networks. Text mining. Machine learning.

1. Introduction

Due to the large and growing number of users, social networks have become a valuable source of information generated by people of all genders and ages, from virtually all social, cultural and economic backgrounds. The data available is a valuable source for knowing users' opinions, or feelings, about a variety of topics, such as people, events and products.

To give an idea of the size of this new source of information, it is worth remembering that, according to the site statista (http://www.statista.com), in January 2019 Twitter had approximately 330 million active users in the world, more than 18 million in Brazil, who produce hundreds of thousands of tweet every day. Facebook, on the other hand, has close to 2.2 billion active users worldwide, more than 130 million in Brazil, who comment, enjoy and share ideas, opinions and criticism. The volume of messages produced on these networks and their dynamics make it impractical to analyze them by purely manual processes.

Several research works have been conducted in recent years aiming at the sentiment analysis of messages published on networks and social media [1] [2] [3]. The main source of data for these works has been Twitter. This can be illustrated by some Google Scholar searches. In April 2019, when searching for articles containing "sentiment analysis" and Twitter in the title 1080 results are obtained, while when searching for

"sentiment analysis" and Facebook only 67 results are returned. For both media, the vast majority of these works are focused on the English language. Including the keyword Portuguese in the search, the results became four and one, respectively.

The predominance of Twitter as a data source is probably because it offers an API that in a very simple way offers practically unrestricted access to published messages and data associated with its users, such as the number of followers. In the case of Facebook, it is necessary to create an application and obtain the authorization of the users so that their data can be accessed. For news pages, this authorization is not required, although the creation of the application is still necessary.

The sentiment analysis studies described in the literature mostly use the Naive Bayes, Maximum Entropy and Support Vector Machine (SVM) algorithms. Recent works explore the approach known as deep learning with good results in terms of accuracy, when available a large number of examples for training, but at the expense of high training times [4] [5].

In most available systems for sentiment analysis, as [6], the user has access to an interface from which he can specify the topic of interest using keywords. Once the messages (tweet, posts) that contain the keywords are retrieved, the system determines the percentage of them with positive, negative and neutral sentiment expressed (other classifications are possible). A major drawback of this approach is that the specified keywords may be contained in messages dealing with various, possibly unrelated, topics. For example, when using the keyword 'bank', the system can retrieve messages dealing with economics, gardening or information systems. Even in less anecdotal cases, when all messages are related, they eventually deal with various aspects of the topic, which may be associated with different opinions and sentiments. For example, the use of keywords that describe an action of public officials (prosecutors, judges, police) investigating cases of corruption involving politicians and businessmen might return messages may carry a mostly negative sentiment when referring to the criminal agents (politicians and businessmen), but positive when referring to the agents carrying out the investigation (prosecutors, policemen), or the opposite. For this reason, particularly when the number of messages is high, it becomes important to group the returned messages based on their similarities, in order to evaluate the polarity of the group to which they belong.

It is also worth to mention that most works available in the literature that use Facebook as a data source generally focus on the sentiment analysis of posts or comments, such as [7] and [8], or sometimes in their clustering, such as [9], but not on both tasks.

Aiming, therefore, to contemplate the requirements of the context presented, this article discusses the approach used to build BP (from Boca do Povo - People's Mouth in Portuguese), a system that allows users to indicate a topic of interest published in Brazilian Portuguese on Facebook . The returned posts will be grouped based on their similarities and, for each group, the comments to these posts will be labeled a

positive, negative or neutral. The results will be presented to the user through an easy-to-understand interface, who can, if desired, view the recovered posts and comments and their labels.

The rest of this article has the following structure: section 2 briefly presents the functionality made available to users; section 3 discusses the architecture of the system developed to implement the functionalities; section 4 presents the procedure used to evaluate the mechanisms adopted for clustering posts and classifying comments; section 5 closes the article with some final considerations.

2.BP: Functionality

BP offers its functionality to users through a Web interface. The initial screen of the system allows the user to indicate the term he wants to use for the search of posts stored in a local database of posts. Every post that contains the term will be retrieved. As the recovered posts will be clustered according to their similarity, the user must specify how many groups he wants to create. The user will also be able to inform in which pages he wants the search to be made. In this way, it is possible to restrict the number of recovered posts as well as to analyze the differences of sentiment in comments on a topic in posts of pages of different vehicles, since each can have a different audience.

Word (or tag) clouds are created to represent the clusters in order to facilitate the identification of the main topics they covered. The size of each word in the cloud is related to its frequency of occurrence in the group. The most frequent words are presented using larger fonts with more vivid colors to highlight them. If the user wishes, he can access the posts in a group and their comments by clicking on the respective tag cloud. Together with the tag clouds are displayed the totals of positive, negative and neutral comments made to the posts that make up the group.

The user also has access to a feature that allows him to label comments in order to increase the database used to train the classifier of sentiments.

3.BP: Architecture and Algorithms

The architecture of BP consists of five main modules: User Interface, Data Extraction, Posts Clustering, and Sentiment Analysis, described below.

3.1 User Interface

The system's user interface is a Web application created with Web Forms that allows the user to specify the keyword that will be used to search the posts of interest.

3.2 Data Extraction

The Data Extraction module consists of a background process that periodically searches for posts and their respective comments on Facebook pages. The retrieved posts and comments are stored in tables of a local DB to be used by other system processes, in particular by the clustering and classification processes.

Graph API is the application programming interface provided by Facebook for data extraction. It is used through HTTP requests that return the searched data in JSON format. In order to make requests to the Graph API it is necessary to have an access token, obtained when creating an application on Facebook. The Graph API has some limitations for performing searches with keywords, so that in order to make this possible, the posts are transferred to a local database where the search is performed.

3.3 Post Clustering

The posts clustering module gathers posts that are similar, seeking to make the formed clusters as different from each other as possible.

Clustering is performed using the unsupervised algorithm K-Means. Initially, the algorithm randomly chooses the posts that will constitute the centroids of the groups. The remaining posts are associated with the clusters whose centroids are closest. Then, the centroids of all clusters are recalculated and the posts are again assigned to the clusters whose centroids are closest. This process is repeated until no posts are moved from groups. The temporal complexity of the algorithm is $O(D.C.L_{Average}.I)$, where D represents the number of posts in the cluster, *C* the number of groups, $L_{Average}$ the average number of words on each post and *I* the number of times the process is repeated until the stop condition is reached. Once the clustering process is finished, a tag cloud is produced for each cluster.

Before starting the clustering, posts are pre-processed. They are transformed into tokens using spaces and punctuation marks as separators; all letters are transformed to lowercase and accents are removed. Stop words are also removed. The end result of post preprocessing is a term vector. Also, during post preprocessing it is formed the vocabulary to be used during clustering, consisting of a non-repeatable list of all the terms found in the posts.

To apply K-Means texts have to be represented so that they can be compared and the distance between them measured. In this way, it is possible to assign them to the group with the closest centroid. To do this, posts are represented as vectors in which each position corresponds to a vocabulary term. Each vector position will contain the tf_idf (term frequency - inverse document frequency) of the term corresponding to that position.

The *tf_idf* of a term *t* for a document *d* (a post) is calculated by multiplying the frequency of occurrence of the term in the document, $tf_{t,d}$, by the inverse frequency of the term in all documents, idf, as shown in (5).

$$tf\_idf_{t,d} = tf_{t,d} * idf_t \quad (5)$$

The computation of the frequency of a term t in a document *d*, $tf_{t,d}$, is done applying (6), where $T_{t,d}$ is the number of occurrences of term *t* in post *d* and T_d the total number of terms in post *d*.

$$tf_{t,d} = \frac{T_{t,d}}{T_d} \quad (6)$$

The inverse frequency of occurrence of a term *t* in the document collection, idf_t, is a measure of the amount of information the term carries. A term carries a lot of information if it is rare in the collection of documents, and little information if not. The calculation of idf_t is made using (7), where *N* is the total number of documents (posts) in the collection and df_t the frequency of occurrence of the term *t* in all documents. The *idf* is calculated only once for each term in the vocabulary, since it does not depend on a specific post.

$$idf_t = ln\frac{N}{df_t} \quad (7)$$

The comparison of vectors representing two posts is done using the cosine similarity, which consists of calculating the cosine of the angle between the vectors to measure the similarity between them, as shown in (8). The cosine similarity of two vectors may vary from 0 to 1. If is equal to 1, the angle between the vectors is 0 ° and therefore the posts are identical.

$$cossim(P, C) = \frac{P.C}{||P||||C||} \quad (8)$$

This calculation is performed *n* times for each post, *n* being the number of groups to be formed. The highest similarity found between the post being grouped and the current centroids determines which group the post will be assigned to. After assigning all the posts to the groups, a new centroid is calculated for each group considering the average of the *tf_idf* of the posts belonging to each group. The process of comparing posts and calculating the position of the centroid is repeated until the stop condition is reached, that is, until there is no change in the distribution of posts in the groups.

The tag cloud is the chosen way to present to the user the formed clusters. The purpose of the tag cloud is to represent the content of the documents in the group in an intuitive and easy-to-view way. A tag cloud

displays only the most frequent terms of the posts belonging to a group. Terms are differentiated by the color and size of the font, to show their relevance in comparison to the other terms in the group. The forty-two most frequent terms of the post groups are shown. This amount has been determined empirically, having shown itself visually clearer after testing with other amounts.

3.4 Sentiment Analysis

This module classifies comments into the categories "positive", "negative" or "neutral" using the Naive Bayes supervised machine learning algorithm.

3.4.1 Comment Preprocessing

Before applying the sentiment classification algorithm, comments have to be preprocessed, in a process similar to what was done for post clustering, but more elaborated.

One of the aspects to consider in text preprocessing is the number n of adjacent words, or n-grams, that will be used to compose the terms to be analyzed by the algorithm. BP uses unigrams, since we empirically observed that the quality of the classification using unigrams and bigrams is very similar, and the use of bigrams requires a significantly higher processing load, confirming what had already been verified in [10]. Therefore, the initial step of pre-processing is to separate all words considering as delimiters spaces and special characters.

As a social network, Facebook allows users to tag other users by entering their names in a comment. This way, the marked user will be notified about the comment and the post in which their name was quoted or marked. Since names are not relevant to determining the sentiment expressed in comments and are present in a considerable number of them, they are removed. Other words also considered irrelevant for classification (stop words) are removed. Words with two or fewer characters are also removed.

Emoticons, such as ":)" or ":(", and emojis, such as “☺” and “☹”, are commonly used to express sentiments and emotions and are therefore of great importance for sentiment analysis. These symbols are recognized and replaced by "emt_bom" and "emt_ruim", depending on the associated emotion.

Laughter, identified when certain terms such as "kkk" or "haha" are found in the text, are removed from the comments, because the feeling they express depends on the context.

A disadvantage of using unigrams is that they ignore negations; for example, in expressions such as "not good" or "not like". Words such as "good" or "like" alone indicate positive opinions, but "no" indicates the opposite. In BP this is treated by adding "n_" in front of the words that are preceded by indicators of denial, such as "no" and "neither".

Then all alphabetic characters are converted to lower case, the punctuation signs and accents are removed, as well as the adjacent repeated letters, such as in "wooord" which becomes "word". These actions are taken to ensure that multiple occurrences of the same word are detected, even in situations where there are typing errors.

3.4.2 Training the Classifier

In the training stage a set of comments previously classified by a human being is used to calculate the probability of each vocabulary term belonging to one of the classes considered (positive, neutral and negative)..

The computation of the probability that a term *t* belongs to a class *c* is done using (1). It consists of dividing the number of times the term appears in the comments of class *c* by the total number of terms of all the comments contained there. This calculation is made for each class considered (positive, negative and neutral) and for each term present in the vocabulary, thus obtaining, at the end, three distinct probabilities for each term of the vocabulary. As it is possible that a term is not present in all classes, some probabilities would result in zero, which would be a problem when it is necessary to perform the product of these probabilities, or use their logarithm, as is the case during the classification process described below. In order to avoid this problem it is used the smoothing of Laplace, which consists of adding 1 to the numerator and to the denominator of (1).

$$P(t|c) = \frac{T_{tc} + 1}{\sum_{t' \epsilon V}(T_{ct'} + 1)} \quad (1)$$

In (1), *t* represents the term whose probability is being calculated, T_{tc} is the number of times that term *t* appears in the comments of class *c*, and *V* is the vocabulary.

The Naive Bayes classifier also takes into account the a priori probability of each class, calculated by (2), which finds the ratio between the number of comments in the training set belonging to the class and the total number of comments available in the training set.

$$Priori(c) = \frac{N_c}{N} \quad (2)$$

In (2), *c* represents a class, N_c the number of comments belonging to class *c* and *N* the total number of comments in the training set.

The temporal complexity of the Naive Bayes algorithm in the training stage is $O(D.L_{Average})$, where D represents the number of comments and $L_{Average}$ the average number of terms per comment. The number of classes and terms in the vocabulary is not included because they are constant and have very low values in relation to the number of comments and the average number of terms.

As with the training stage, before going through the classification module, comments are pre-processed. To determine the class to which a comment will be associated, the probability of the comment belonging to each of the classes considered is calculated, with the comment being associated with the most probable. This can be done as shown in (3).

$$class = \underset{c \in \mathbb{C}}{arg\,max}\, Priori(c) \prod_{1 \le k \le n_d} P(t_k|c) \quad (3)$$

In (3), $\mathbb{C}$ is the set of classes considered, c represents a specific class, n_d the number of terms in the comment d and t_k is the term k of the comment d being analyzed. In this equation, conditional probabilities are multiplied, indicated by the term $P(t_k|c)$, one for each term in the comment.

Probability values for terms are generally very low, since each term does not repeat much compared to the total number of terms present in the thousands of comments used for training. Multiplying those probabilities could generate a very low value, resulting in underflow, that is, a value so low that it would not be possible to represent. To solve this problem, we can use the mathematical property log(xy) = log(x) + log(y), as shown in (4), so that instead of multiplying the probabilities we add their logarithms. As the logarithmic function is monotonous (if x > y then log(x) > log(y)), the highest probability found by (4) will also be the highest probability found by (3).

$$class = \underset{c \in \mathbb{C}}{arg\,max}[log\, Priori(c) + \sum_{1 \le k \le n} logP(t_k|c)] \quad (4)$$

The temporal complexity of Naive Bayes, resulting from the application of (4), is $O(D.L_{Average})$, where D represents the number of comments and $L_{Average}$ the average number of terms per comment. As in the training stage, the number of classes is not considered because it is a very low constant value in relation to the number of comments and the average number of terms in each comment.

4.Evaluation and Validation

In this section, we present the procedure used to evaluate the performance of the sentiment analysis and clustering algorithms. The purity of the groups formed, the overall polarity of the comments [11] and the time required for the sentiment analysis are the metrics considered. The evaluation aimed to verify if the objective proposed for the research was achieved, i.e., that the grouping and the sentiment analysis performed by BP are close to those that would be performed by a human being, but performed much more quickly.

For the evaluation, 21 portals were used, including newspapers such as A Folha de São Paulo, organizations such as UN Brazil, and public institutions such as the Federal Senate.

4.1 Classification of Training Comments

The classification of the training set proved to be a difficult task, since each comment had to be manually classified and entered into the database with the respective classification. During this stage some basic rules were followed. The first was to classify a comment by what is written in it, even if there is no direct relation between the comment and the post to which it is associated. For example, a comment with a negative criticism of the government is considered negative, even if it is made in a post that reports an automobile accident. Another rule was to consider a neutral comment when it presented positive and negative characteristics in similar amounts.

The manual classification resulted in a training set containing 1431 comments, of which 932 were negative, 300 were positive and 202 were neutral.

4.2 Evaluation of Sentiment Analysis

The time required by BP to classify all comments in the evaluation set and the overall polarity obtained were compared with the classification times and overall polarity resulting from the analyses made by three evaluators. One of the evaluators was female, trader, aged 46 years, the other two evaluators were male, IT professionals, aged 29 and 22 years. The basic requirement for choosing the evaluators was their familiarity with Facebook.

Three posts were chosen, each containing at least fifty comments, found in the searches performed using the keywords "manifestações", "economia" and "lava jato", terms that dealt with popular issues in Brazil at the time of the evaluation.

After BP and the evaluators classified the comments, the polarity coefficient, r, was computed using (8).

$$r = \frac{C_{Positive}}{(C_{Positive} + C_{Negative})} \quad (8)$$

In (8), $C_{Positive}$ and $C_{Negative}$ represent the amount of positive and negative comments, respectively, so that *r* varies from 0 to 1; being equal to 1 when the sentiment of all comments is positive and 0 when all are negative.

The results obtained are shown in tables 1, 2, 3 and 4. In tables 1, 2 and 3, column BP represents the classifications performed by BP; columns A1, A2 and A3 represent the results for each of the three evaluators, column $\bar{x}$ is the mean of these three values and column σ the respective standard deviation. The first line (-) shows the number of posts classified as negative, the second (~) as neutral, the third (+) as positive and the last (r) the corresponding global polarity.

Table 1. Global polarity for "Lava Jato"

	BP	A1	A2	A3	$\bar{x}$	σ
-	72	68	69	65	67,33	1,70
~	3	7	6	10	7,67	1,70
+	3	3	3	3	3,00	0,00
r	0,04	0,042	0,042	0,044	0,043	0,001

It can be observed that the polarity coefficients obtained from the classifications made by the evaluators and by the system have very low values. On the other hand, the tables show that the polarity coefficient calculated for the classification made by BP has the same bias as the coefficients calculated for the classifications made by the evaluators, and that both show a great predominance of negative comments. Thus, one of the objectives proposed for the study described was satisfied, namely that the classification of comments performed by BP be similar to the classification performed by human reviewers.

Table 2 Global polarity for "manifestações"

	BP	A1	A2	A3	$\bar{x}$	σ
-	60	50	52	52	51,33	0,94
~	3	11	12	10	11,00	0,82
+	5	7	4	6	5,67	1,25
r	0,077	0,123	0,071	0,103	0,099	0,021

Table 3. Global Polarity for "ECONOMIA"

	BP	A1	A2	A3	$\bar{x}$	σ
-	53	52	56	54	54,00	1,63
~	0	4	1	3	2,67	1,25
+	4	1	0	0	0,33	0,47
r	0,07	0,019	0	0	0,006	0,009

Table 4 presents the times taken by the evaluators to conclude the classification of the 203 comments used in the evaluation and the time required by BP to perform the sentiment analysis for 245 random comments (it is not relevant which comments are analyzed, since their sizes, and therefore the processing times, vary little). As expected, the system performs the classification much faster than a human being. The time spent by BP does not include the training time of the algorithm, since this process does not need to be repeated for each classification event and can be performed occasionally and off-line.

Table 4. Time to classify comments

BP	A1	A2	A3
00m 03s 062ms	26m 19s 329ms	14m 56s 854ms	18m 32s 015ms

4.3 Clustering Evaluation

The evaluation of the clustering algorithm was performed by two evaluators, the first female, trader, aged 46 years, and the second a male, 22-year-old student. The evaluation was performed considering three clusters resulting from searches using the again the terms "manifestações", "economia" and "lava jato".

Three lists of posts were presented to the evaluators, each list corresponding to a group, and for each list the tag cloud that represents the group. The evaluators had to indicate, for each post, if it was properly assigned or not, and for this he could consult the cloud of labels corresponding to the group.

The quality of the groups was checked by calculating their purity, p, expressed by (9) [12].

$$\mathrm{p}(\Omega, \mathrm{C}) = \frac{1}{\mathrm{N}} \sum_{\mathrm{k}} \max_{\mathrm{j}} |\varpi_{\mathrm{k}} \cap \mathrm{c}_{\mathrm{j}}| \qquad (9)$$

In (9), Ω is the set of groups $\{\omega_1, \omega_2, \ldots, \omega_k, \}$, C the set of topics addressed by each group $\{c_1, c_2, \ldots, c_j, \}$, ϖ_k a group of posts, identified through the tag cloud, c_j represents a topic contained by this group and *N* the total number of posts in all groups. The maximum value of the intersection between ϖ_k and c_j, denoted by $\max_j |\varpi_k \cap c_j|$ is the number of correctly grouped posts in each group. The metric purity ranges from 0 to 1, and the closer it is to 1, the purer the cluster.

The purity of each group was calculated according to the analysis of each evaluator, as well as the mean purity for the three clusters. The results are presented in tables 5, 6 and 7, where columns G1, G2, and G3 represent the three clusters generated by BP.

Table 5. Purity for “Lava Jato”

Comment correctly clustered	A1			A2			$\bar{x}$		
	G1	G2	G3	G1	G2	G3	G1	G2	G3
Yes	69	41	12	63	37	11	66	39	11,5
No	16	8	5	22	12	6	19	10	5,5
p	0,808			0,735			0,772		

Table 6. Purity for “Manifestações”

Comment correctly clustered	A1			A2			$\bar{x}$		
	G1	G2	G3	G1	G2	G3	G1	G2	G3
Yes	9	12	15	8	12	16	8,5	12	15,5
No	1	4	16	2	4	15	1,5	4	15,5
p	0,632			0,632			0,632		

Table 7. PURITY FOR “Economia”

Comment correctly clustered	A1			A2			$\bar{x}$		
	G1	G2	G3	G1	G2	G3	G1	G2	G3
Sim	21	21	63	20	20	64	20,5	20,5	63,5
Não	10	12	13	11	13	36	10,5	12,5	24,5
p	0,64			0,634			0,637		

4.4 Results Evaluation

The results of sentiment analysis indicate a very negative polarity of comments about news posts on Facebook. This may be an indication that people are more likely to opine when they are not satisfied, thus making negative comments. In addition, it should be noted that the news in the posts used in the evaluation tend to report negative facts to most people.

Clustering has proven to be a challenging task, which depends on several factors that influence the results. In particular, the number of groups to form. It is difficult to establish the "optimal" number of clusters to form because it heavily depends on the considered texts. Nevertheless, the results achieved are positive. As a secondary result, the tag cloud proved to be adequate to represent the groups.

5.Related Work

In this section, we review recent publications that deal with the development of methods and systems related to the main focus of this article, that is, to group by thematic similarity posts and comments related

to news written in Brazilian Portuguese, on Facebook, and determine the polarity of each identified group, in an integrated manner.

In [13] it is described a model for sentiment analysis using domain ontologies and case-based reasoning. The authors report results higher than NB and SVM for reviews of cameras and films, in English, extracted from the Amazon site.

A proposal to rank news affecting the coffee market as positive or negative is described in [14]. In experiments using a NB classifier, with news collected from the Web, mostly written in English, accuracies ranging from 52% to 68.53% were obtained.

In [15] machine learning algorithms are explored to identify aggressive comments on Twitter, after the tweets go through a pre-processing phase. The problem is similar to sentiment analysis, but instead of classifying the messages as positive or negative, they are classified as aggressive or non-aggressive, which requires specific training sets for the task. The classifiers with the three highest accuracies were Bayesian (70.94%), SVM (70.50%) and neural network (70.43%).

Unlike previous studies based on machine learning algorithms, in [16] it is presented a lexicon based approach, aimed at the sentiment analysis of comments in Portuguese posted in cancer related Facebook communities. The proposal is compared with other methods based on lexicon for sentiment analysis for the Portuguese language, as well as with tools for the English language. Before using the English tools, the comments in Portuguese are translated into English. In experiments reported in the article, the proposed method obtained an average accuracy of 65.78%, surpassing the other tools based on lexicon for the Portuguese language and the tools for the English language, the latter with poor performance.

An experimental study in which clustering techniques are applied to perform the task of sentiment analysis is described in [17]. The authors found that clustering algorithms of the k-means family clearly show advantages in well-balanced sets, which do not occur in unbalanced sets.

In [18] it is proposed a method for tweet sentiment analysis using an attribute clustering scheme based on the chi-square test ($\chi 2$), together with POS tagging and a multinomial Naive Bayes model.

In [19], a cluster-based method for sentiment analysis is proposed as an unsupervised solution independent of domain with competitive accuracy with supervised methods. The work proposes to find alternatives to the expensive process of manual data annotation required by supervised learning methods. The method was evaluated using several sets of review messages from various types of products, with acceptable results.

Afonso and Duque [20] also studied the use of clustering methods for the automatic classification of texts in Brazilian Portuguese for several domains. In the experiments described, the automatic classification methods produced results far from those obtained with manual classification.

In [21], it is presented a system for clustering texts in Brazilian Portuguese composed of two modules: one to form indices consisting of compound terms of texts and another to apply an evolutionary clustering algorithm that uses the indices to gather the texts in common topics. The experiments described show acceptable results, although the number of groups produced is far from the original number of topics created manually.

The above literature revision shows that most work deals either with clustering or sentiment analysis of texts, and not with an integrated process that involves clustering by thematic similarity and identifying group polarity, as proposed by the work object of this article. In the case of sentiment analysis, the number of publications dealing with the sentiment analysis of texts in Brazilian Portuguese published on Facebook is very small. On the other hand, the works we found that deal with clustering and sentiment analysis use clustering to perform sentiment analysis and not as a step in an integrated process of text clustering and polarity computation.

6.Conclusion

In the work described in this article, a system was created using two machine learning algorithms in order to group similar news posts published on Facebook and classify the sentiment associated with the comments made to the posts. BP, the Web system created, fulfilled the expectations and achieved the originally proposed objective: to group posts and classify comments with similar quality to that of people, but in less time.

The use of Facebook as a data source is an element to be highlighted in the work, since most recent research on sentiment analysis uses Twitter or Web crawling as sources. The reason for this is probably that Twitter is a source of short and simple text with unrestricted access. In the case of Facebook, it is necessary to create an application and gain user authorization so that certain data can be accessed. For Facebook news pages, this authorization is not required, although the creation of the application is still necessary.

In future work, we intend to explore new clustering algorithms as well as other machine learning algorithms to classify sentiments in order to try increase the accuracy of the system [22]. In this context, the construction of classification committees (ensembles) is a method that has presented good results in classification problems and should be explored [23].

7.References

[1] E. Cambria, B. Schuller, Y. Xia, e C. Havasi, "New avenues in opinion mining and sentiment analysis", IEEE Intell. Syst., nº 2, p. 15–21, 2013.

[2] R. Feldman, "Techniques and applications for sentiment analysis", Commun. ACM, vol. 56, nº 4, p. 82–89, 2013.

[3] M. Á. García-Cumbreras, A. Montejo-Ráez, e M. C. Díaz-Galiano, "Pessimists and optimists: Improving collaborative filtering through sentiment analysis", Expert Syst. Appl., vol. 40, nº 17, p. 6758–6765, 2013.

[4] X. Zhang e Y. LeCun, "Text Understanding from Scratch", Prepr. ArXiv150201710 Cs, fev. 2015.

[5] M. Zhang, Y. Zhang, e D.-T. Vo, "Gated Neural Networks for Targeted Sentiment Analysis", 2016.

[6] P. Grandin e J. M. Adan, "Piegas: A Systems for Sentiment Analysis of Tweets in Portuguese", IEEE Lat. Am. Trans., vol. 14, nº 7, p. 3467–3473, 2016.

[7] A. Ortigosa, J. M. Martín, e R. M. Carro, "Sentiment analysis in Facebook and its application to e-learning", Comput. Hum. Behav., vol. 31, p. 527–541, 2014.

[8] S. S. Dasgupta, S. Natarajan, K. K. Kaipa, S. K. Bhattacherjee, e A. Viswanathan, "Sentiment analysis of Facebook data using Hadoop based open source technologies", in Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on, 2015, p. 1–3.

[9] V. Franzoni, Y. Li, P. Mengoni, e A. Milani, "Clustering Facebook for Biased Context Extraction", in International Conference on Computational Science and Its Applications, 2017, p. 717–729.

[10] B. Pang, L. Lee, e S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques", in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, 2002, p. 79–86.

[11] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter, e V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in twitter", in Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval, 2015.

[12] C. D. Manning, P. Raghavan, e H. Schtze, Introduction to information retrieval. Cambridge University Press New York, NY, USA, 2008.

[13] F. Ceci, A. L. Goncalves, e R. Weber, "A model for sentiment analysis based on ontology and cases", IEEE Latin America Transactions, vol. 14, nº 11, p. 4560–4566, 2016.

[14] P. O. L. Junior, L. G. de Castro Junior, e A. L. Zambalde, "Applying Textmining to Classify News About Supply and Demand in the Coffee Market", IEEE Latin America Transactions, vol. 14, nº 12, p. 4768–4774, 2016.

[15] L. P. Del Bosque e S. E. Garza, "Prediction of aggressive comments in social media: an exploratory study", IEEE Latin America Transactions, vol. 14, nº 7, p. 3474–3480, 2016.

[16] R. G. Rodrigues, R. M. das Dores, C. G. Camilo-Junior, e T. C. Rosa, "SentiHealth-Cancer: a sentiment analysis tool to help detecting mood of patients in online social networks", International journal of medical informatics, vol. 85, nº 1, p. 80–95, 2016.

[17] B. Ma, H. Yuan, e Y. Wu, "Exploring performance of clustering methods on document sentiment analysis", Journal of Information Science, vol. 43, nº 1, p. 54–74, 2017.

[18] Y. Wang, K. Kim, B. Lee, e H. Y. Youn, "Word clustering based on POS feature for efficient twitter sentiment analysis", Human-centric Computing and Information Sciences, vol. 8, nº 1, p. 17, 2018.

[19] M. T. AL-Sharuee, F. Liu, e M. Pratama, "Sentiment analysis: An automatic contextual analysis and ensemble clustering approach and comparison", Data & Knowledge Engineering, vol. 115, p. 194–213, 2018.

[20] A. R. Afonso e C. G. Duque, "Automated text clustering of newspaper and scientific texts in brazilian portuguese: analysis and comparison of methods", JISTEM-Journal of Information Systems and Technology Management, vol. 11, nº 2, p. 415–436, 2014.

[21] A. R. Afonso, "Brazilian Portuguese Text Clustering Based on Evolutionary Computing", IEEE Latin America Transactions, vol. 14, nº 7, p. 3370–3377, 2016.

[22] A. Ceron, L. Curini, e S. M. Iacus, "iSA: a fast, scalable and accurate algorithm for sentiment analysis of social media content", Inf. Sci., 2016.

[23] N. F. da Silva, E. R. Hruschka, e E. R. Hruschka, "Tweet sentiment analysis with classifier ensembles", Decis. Support Syst., vol. 66, p. 170–179, 2014.