SentiMeter-Br: Facebook and Twitter Analysis Tool to Discover Consumers' Sentiment

Renata Lopes Rosa, Demostenes Zegarra Rodriguez, Graca Bressan
Department of Computer Science and Digital Systems
University of Sao Paulo, SP - Brazil
Email: rrosa@usp.br, demostenes@larc.usp.br, gbressan@larc.usp.br

***Abstract*—Brazilian Consumers' Sentiments are analyzes in a specific domain using a system, SentiMeter-Br. A Portuguese dictionary focused on a specific field of study was built, in which tenses and negative words are treated in a different way to measure the polarity, the strength of positive or negative sentiment, in short texts extracted from Twitter. For the Portuguese dictionary performance validation, the results are compared with the SentiStrength tool and are evaluated by three Specialists in the field of study; each one analyzed 2000 texts captured from Twitter. Comparing the efficiency of the SentiMeter-Br and the SentiStrength against the Specialists' opinion, a Pearson correlation factor of 0.89 and 0.75 was reached, respectively. The polarity of the short texts were also tested through machine learning, with correctly classified instances of 71.79% by Sequential Minimal Optimization algorithm and F-Measure of 0.87 for positive and 0.91 for negative phrases. Another contribution is a Twitter and Facebook search framework that extracts online tweets and Facebook posts, the latter with geographic location, gender and birthdate of the user who posted the comments, and can be accessed by mobile phones.**

Keywords*—*consumer sentiment; Twitter; Facebook; machine learning; social web analysis tool; support vector machines;

I. Introduction

Nowadays, people express their sentiments and opinions through social networks and micro-blogs very commonly. There are many sentiment analysis tools for texts posted at micro-blogs, but most of these tools dictionaries are only in English and it is important to consider different people's consumerism vision according to each country and each city.

Analysis tools of emotional texts based on word lists, are best known as ANEW [1], OpinionFinder [2], SentiWordNet, WordNet and SentiStrength [3], of which only the latter analysis tool has support for the Portuguese language and it considers only the unigrams. Sentimeter-Br dictionary covers unigrams (single element/word), bigrams (two adjacent elements) and stopwords (words in a search can be considered irrelevant).

Each field of study requires a specific dictionary, as well as lists of stopwords (e.g., the, of) [4], which need not be analyzed because they do not add value to the performance analysis. Slang and expressions according to each country also need to be considered. It is also important to define the field to be studied,to build a correct dictionary because an only single word may expresses a positive or a negative value or even no kind of emotion.

In [5], an architecture for analyzing in the smartphones field is assembled, analyzing consumers' vision in Twitter, but the study analyzes only one specific micro-blog. Using social networks to analyze sales and features of smartphones or other objects is justifiable because the amount of information is faster to collect, and more data can be gathered.

In [7], a tool captures Twitter data and the polarity of the reviews is analyzed, including slangs that, albeit widely used in social networks, are excluded by different word analyses that have been already mentioned above. A generic dictionary is used in this study. In [8], semantic analysis tools are studied, showing the difficulty to analyze texts from Twitter because there are many slang words and expressions of emotion in the form of symbols. It also shows that some words are not useful to analyze feelings, the so-called stopwords.

The contribution of this work is building a dictionary with the use of regional slangs, emotions, negative words and different verb tenses that have not been considered in other works. A different metric was used, depending on the tense and negative words in the text. The most frequently words were extracted from Google Trends [6] in the last four months to be used in the dictionary.

This work is compared with the SentiStrength tool that has several limitations. The SentiStrength estimates the strength of negative and positive sentiment in short texts. In this work, we joined these two values and turned into a single one.

The polarity of the dictionary was validated by the machine learning technique. The Weka (Waikato Environment for Knowledge Analysis) [9] software was used as a tool for the data analysis.

The algorithms used in Weka were Bayesian networks (Naive Bayes and Bayes Multinomial), Decision trees (C4.5) and Sequential Minimal Optimization (SMO). These algorithms were used to train the data and to decide if a sentence has a positive, negative, neutral or spam value [10].

We built a Twitter and Facebook search frameworks that can be accessed by mobile phones. Thus, these mobile users have access to promotions spread over social networks. The Facebook search framework is complete because it considers the user' s geographical location and their birth dates, if they have configured this information in their Facebook' s user accounts.

Section II provides a theoretical revision of sentiment analysis. Section III deals with the SentiMeter-Br architecture. Section IV presents the machine learning algorithms used in this work. Section V presents the Twitter and Facebook search framework. Section VI presents the results and discussions and finally, Section VII presents conclusions and the future works.

II. SENTIMENT ANALYSIS

Sentiment analysis, also known as opinion mining, has been studied by researchers, mainly in social webs, such as Twitter. It is a type of computational study of text in natural language, which aims to identify sentiment polarity and intensity of sentiments [11]. The sentiment analysis goal is to classify the polarity of a given text, helping to define if a sentence is positive, negative, or neutral. It is associated to a number in a -5 to +5 scale (most negative to most positive). Each word uses natural language processing or a word dictionary. Another research direction is the subjectivity or objectivity identification [12], but it will not be covered here.

Opinion mining can be used in different topics. A topic in which opinion mining can help is marketing intelligence to know more about people's consuming habits.

Opinion mining in textual data for marketing intelligence can be categorized into three types [13]:

- Early alerting: informs subscribers when a rare, but critical or even fatal condition occurs.
- Buzz tracking: follows trends in topics of discussion and understand what new topics arise.
- Sentiment mining: extracts aggregate measures of positive versus negative sentiment opinion.

This paper analyzes the sentiment mining type and allows learning about buzz tracking, by capturing the words used in tweets.

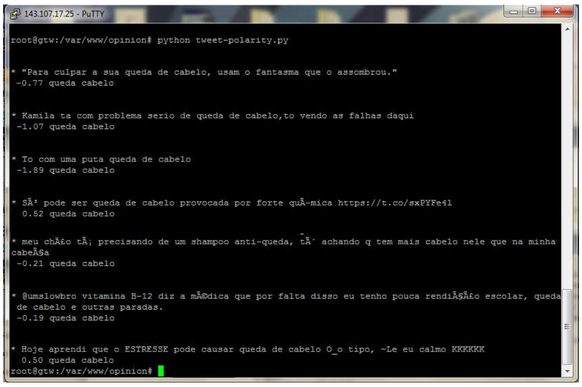
Sentiment analysis is not a simple task in social networks because the texts can be ambiguous. The use of slangs and ironies is difficult to decipher and to put on a scale as a positive or negative sentiment. So, it is important to have a specific dictionary according to the context because a word can have a negative or a positive value, as in the following texts:

- "Dry hair results from a number of reasons: negative value".
- "The carpet was cleaned and dried: positive value".

A. AFINN word list

There are several word lists to be used in sentiment analysis, with different scales for each word. One of them is the AFINN [14]. Each word in this list has a score from -5 (very negative) to +5 (very positive). Most of the negative words have a minus 2 score, and most of the positive ones have a +2 score. Only the strong obscene words have a -4 or a -5 score, and the entire word list has a bias towards negative words (1598 words corresponding to 65%) [15].

In this paper, a sentiment scale similar to AFINN was used, but with new words in the context to be analyzed, listed by Specialists and captured from Twitter.



```
root@gtw:/var/www/opinion# python tweet-polarity.py


* "Para culpar a sua queda de cabelo, usam o fantasma que o assombrou."
 -0.77 queda cabelo


* Kamila ta com problema serio de queda de cabelo,to vendo as falhas daqui
 -1.07 queda cabelo


* To com uma puta queda de cabelo
 -1.89 queda cabelo


* SÃ³ pode ser queda de cabelo provocada por forte quÃ-mica https://t.co/sxPYFe41
  0.52 queda cabelo


* meu chÃ£o tÃ¡ precisando de um shampoo anti-queda, tÃ´ achando q tem mais cabelo nele que na minha
cabeÃ§a
 -0.21 queda cabelo


* @umslowbro vitamina B-12 diz a mÃ©dica que por falta disso eu tenho pouca rendiÃ§Ã£o escolar, queda
 de cabelo e outras paradas.
 -0.19 queda cabelo


* Hoje aprendi que o ESTRESSE pode causar queda de cabelo O_o tipo, ~Le eu calmo KKKKKK
  0.50 queda cabelo
root@gtw:/var/www/opinion# 
```

Fig. 1. Sentiment strength value of tweets using the search word *hair loss* = *queda de cabelo* in portuguese.

III. SENTIMETER-BR ARCHITECTURE

Before collecting texts from Twitter, we do a preliminary screening for the most commonly used words in the Internet searches regarding the study area (hair cosmetics) through Google Trends in a four-month period. The dictionary, which is specialized in hair care (shampoos, hair loss, products for greasy hair), began to be formed by AFINN with words of common usage as good, confident, accident. Two specialists were sought to cite most commonly used words concerning about hair cosmetics with a suggestion of values from -5 to +5. These words were added to the dictionary. Their final values were chosen according to an average of the Specialists' suggestion and similar existing words in AFINN list. The most mentioned words by Specialists were adjectives, verbs and some negative words.

Five-hundred texts extracted from Twitter were studied. Some words were also added to the dictionary. The most mentioned by tweets were slangs and some negative words. For other contexts, it will be studied if only five-hundred texts extracted from social networks are necessary or if more texts have to be extracted to make a good classification of polarity.

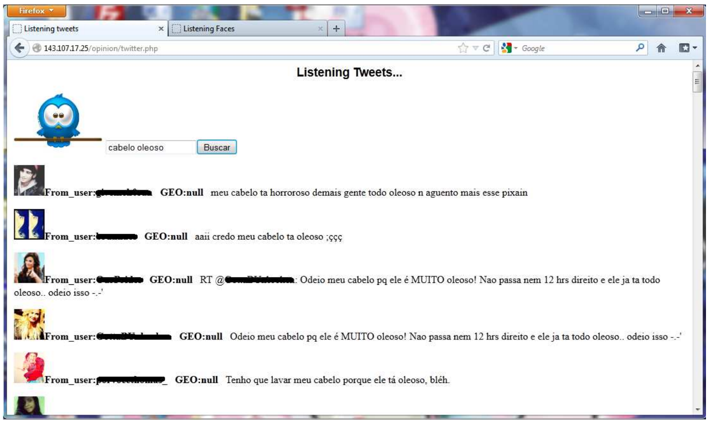
The Sentimeter-Br dictionary contains 2596 words among which 700 words are tenses, 1600 are adjectives (positive and negative adjectives), 130 are slangs, 116 are emotions and 50 are negatives words (e.g., not, never).

The texts from Twitter that helped to build the dictionary were not used as a test. Three other Specialists validated the dictionary in order not to influence the results. Two thousand more tweets were captured to be classified by the Sentimeter-Br and to have their polarity represented in a numeric value.

The SentiMeter-Br architecture is formed by a script (tweet-polarity.py) in python language to calculate the sentiment strength. The script runs and presents the sentiment strength value as shown in Fig. 1. The texts are extracted from Twitter, by the script, using the Twitter Search API (Application Programming Interface). Data is extracted in JSON (JavaScript Object Notation) [16] format.

It is possible to see the tweets collected by means of a friendly framework through a browser, as is seen in Fig. 2. The results of sentiment strength can be seen in Fig. 1.

The messages crossed the Portuguese dictionary (PT-Br),



Listening Tweets...

cabelo oleoso Buscar

From_user:[redacted] GEO:null meu cabelo ta horroroso demais gente todo oleoso n aguento mais esse pixain

From_user:[redacted] GEO:null aaii credo meu cabelo ta oleoso ;ççç

From_user:[redacted] GEO:null RT @[redacted]: Odeio meu cabelo pq ele é MUITO oleoso! Nao passa nem 12 hrs direito e ele ja ta todo oleoso.. odeio isso -.-'

From_user:[redacted] GEO:null Odeio meu cabelo pq ele é MUITO oleoso! Nao passa nem 12 hrs direito e ele ja ta todo oleoso.. odeio isso -.-'

From_user:[redacted]_ GEO:null Tenho que lavar meu cabelo porque ele tá oleoso, bléh.

Fig. 2. Friendly framework of collected tweets.

in which each word has a scale from -1 to -5 for negative sentiments and from $+1$ to $+5$ for positive sentiments. It includes emotions with value -1 or $+1$, slang and strong obscene words with values $+5$ or -5. There are separate files for slangs, negative words, negative adjectives, positive adjectives, emotions and tenses (past tense is in a separate file from present tense) in order to facilitate the application of some exceptions, such as the negative rule and the tense rule.

The general sentiment is calculated, $sentiment_{strenght}$, which is the sum of the words divided by the square of the total number of words that are in the PT-Br dictionary, as shown in line 14, in the pseudocode in Table I. The words that are not in the dictionary are considered stopwords, such as words: de (from), para (to), ela (she), among others.

A test was performed with use of unigrams (one word), bigrams (two words) and some trigrams in the dictionary.

The negative words in the tweets were analyzed. If a negative word (contained in NEG-FILE) is followed by a negative adjective word (contained in NEG-ADJ-FILE), as in the example: *not bad*, the word *not* has value $= -1$, the word *bad* has value -3, the result could be $-4/\sqrt[2]{2} = -2,83$, as shown in line 14 of Table I.

However, the words *not bad* should not be so negative because is similar to the word *adequade* that has value = +1. An exception rule of $sentiment_{strenght}$ is implemented, if there are two negative words together and the final value is less than -1 (line 18 of Table I), the lowest value of negative words (bad = -3) is thus added to the final value, multiplied by -1, line 20 of Table I: $-2.83 + 3 = -0.17$.

When it comes to tenses, there is another exception. If a verb is in the past tense (seen in TENSE-FILE, line 10 of Table I), a value of $+1$ is added to the division part (DIV) because verbs in the past tense are less significant in one sentence than a verb in the present tense, as can be seen below:

- my hair looked (0) good $(+3)$ with the shampoo = $3/\sqrt{3} = +1.73$
- my hair looks (0) good $(+3)$ with the shampoo = $3/\sqrt{2} = +2.12$
- I loved $(+3)$ my hair = $3/\sqrt{2} = +2.12$

TABLE I. SENTIMENT STRENGTH PSEUDOCODE

```
1:  DIV = 0
2:  NEG = 0
3:  for $i = 1$ to $N$ do
4:      read $sentiment(word)$ in ALL-FILES
5:      if (SEARCH $word$ in NEG-FILE) and (SEARCH $nextword$ in NEG-ADJ-FILE) then
6:          $LOWER(sentiment(word), sentiment(nextword))$
            $NEG = NEG + 1$
7:          # NEG-FILE = file with words such as NOT, NEVER
8:          # NEG-ADJ-FILE = file with words such as BAD, UGLY
9:      end if
10:     if SEARCH $word$ in TENSE-FILE then
11:         $DIV = DIV + 1$
12:         # TENSE-FILE = file with LIKED, WAS, WERE
13:     end if
14:     $sentiment_{strenght} = \sum sentiment/\sqrt{len(sentiment + DIV)}$
15:     # $sentiment_{strenght}$: the total of text sentiment value
16:     # $sentiment$: value of words in the PT-Br dictionary
17:     # $len(sentiment)$: the number of words in the text that are in the PT-Br dictionary
18:     if $sentiment_{strenght} < -1$ and $NEG > 0$ then
19:         for $N = 1$ to $NEG$ do
20:             $sentiment_{strenght} = sentiment_{strenght} + (LOWER(sentiment(word), sentiment(nextword))) * -1$
21:         end for
22:     end if
23: end for
```

- I love $(+3)$ my hair = $3/\sqrt{1} = +3$
- it was (0) not (-1) good $(+3)$ = $+2/\sqrt{3} = +1.15$
- it is (0) not (-1) good $(+3)$ = $+2/\sqrt{2} = +1.41$

The sentiment strength was measured throughout the dictionary. In the next section, the classification of positive, negative, neutral or spam was performed by the Weka software to assist in results and Specialists' validation.

IV. MACHINE LEARNING ALGORITHMS

Machine Learning is useful to learn patterns through models and templates already scored. This can be used in sentiment analysis, to discover polarity, for example.

In the Weka software, several machine learning algorithms are already integrated and easy to evaluate.

We used Bayesian networks (Naive Bayes and Bayes Multinomial), Decision trees (C4.5) and the Sequential Minimal Optimization (SMO) algorithm to discover if the texts have a positive value, negative, neutral or spam.

Machine learning was used to evaluate the results already obtained from the PT-Br dictionary.

A. Decision Tree

Decision tree is an algorithm that can be used to give the agent the ability to learn and to make decisions.

A decision tree is a model of knowledge in which each branch linking a child node to a parent node is labeled with an attribute value contained in the parent node.

Learning decision trees are examples of inductive learning; they create a hypothesis based on particular instances that generate general conclusions.

The decision trees take as input a situation described by a set of attributes and return a decision that is the value found for the input value.

B. Bayesian Networks

The Bayesian algorithm rating [17] is based on Bayes' theorem of probability. It is also known as Naive Bayes classifier or only as Bayes algorithm.

The algorithm aims to compute the probability of an unknown sample belonging to each of the possible classes.

This kind of prediction is referred to as statistical classification since it is fully based on probabilities.

Multinomial Naive Bayes is a specialized version of Naive Bayes that is designed more for text documents. Whereas simple Naive Bayes would model a document as the presence and absence of particular words, Multinomial Naive Bayes explicitly models the word counts and adjusts the underlying calculations to deal with them.

The distribution is parameterized by vectors $\theta_y = (\theta_{y1}, \ldots, \theta_{yn})$ for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θ_{yi} is the probability of feature i appearing in a sample belonging to class y.

Parameter θ_y is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting, as in Equation 1.

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad . \tag{1}$$

where:

- $N_{yi} = \sum_{x \in T}$
- $N_y = \sum_{i=1}^{|T|} N_{yi}$ is the total count of all features for class y.

The smoothing prior $\alpha \geq 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations.

Setting $\alpha = 1$ is called Laplace smoothing, while $\alpha < 1$ is called Lidstone smoothing.

C. Sequential Minimal Optimization

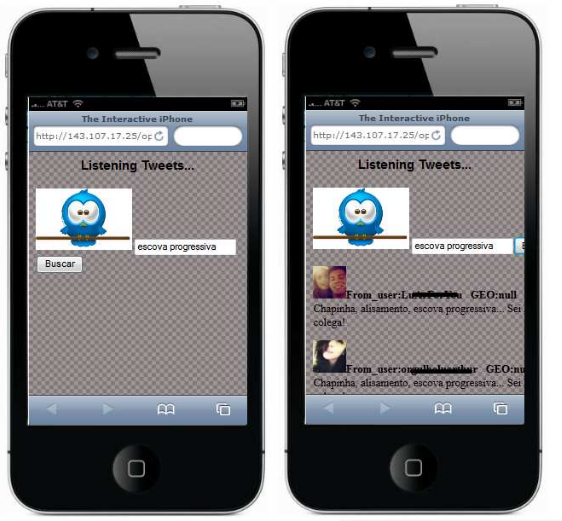
Sequential Minimal Optimization (SMO) is an algorithm described by Platt [18] as using an analytic quadratic programming.

It is an algorithm that solves the Support Vector Machine (SVM) Quadratic Programming (QP) problem without any extra matrix storage and without invoking an iterative numerical routine for each sub-problem.

In [18], the SMO decomposes the overall QP problem into QP sub-problems.

The SMO implements John C. Platt's sequential minimal optimization algorithm for training a support vector classifier using polynomial or RBF kernels.

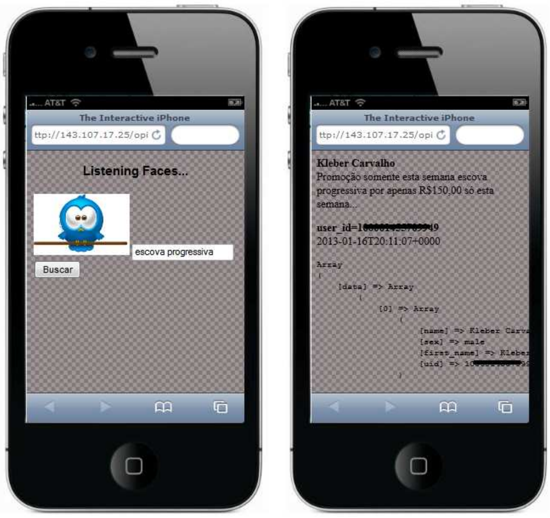
Multi-class problems are solved using pairwise classification.



From_user:Lu[redacted]u GEO:null
Chapinha, alisamento, escova progressiva... Sei colega!

From_user:o[redacted]r GEO:nu
Chapinha, alisamento, escova progressiva... Sei

Fig. 3. Twitter framework.



Kleber Carvalho
Promoção somente esta semana escova progressiva por apenas R$150,00 só esta semana...

user_id=1[redacted]49
2013-01-16T20:11:07+0000

```
Array
(
    [data] => Array
        (
            [0] => Array
                (
                    [name] => Kleber Carva
                    [sex] => male
                    [first_name] => Kleber
                    [uid] => 10[redacted]
                )
```

Fig. 4. Facebook framework.

V. Message Search Framework on Social Web

A friendly message search framework was used to see the texts extracted from Twitter an Facebook. The users can have access to similar preferences and characteristics by using this framework in order to find promotion products in social web.

An interactive iPhone tool [19] was used to emulate an iPhone to test the frameworks. Fig. 3 shows the Twitter framework and Fig. 4 shows the Facebook framework.

The initial configuration in Facebook framework is necessary because it extracts some data that can only be captured by registered users; these data include geographic location and birthdate. The user needs to enter the Facebook search framework and to inform his/her username and password.

In the case of the Twitter framework, no configuration is required. The Twitter framework was built with the PHP programming language version 5.3 and JSON. It is a simple script and does not use an auto login script as Facebook

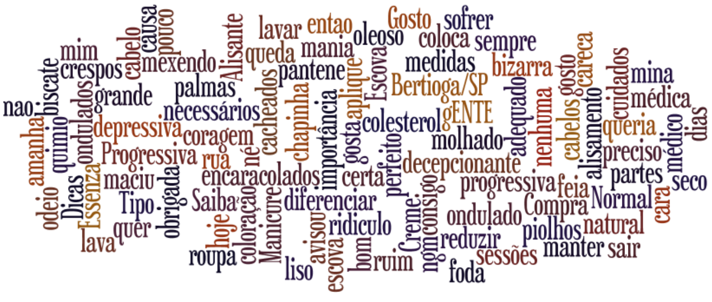


Fig. 5. Word cloud of spam words collected in tweets.

because it captures only tweets and Twitter users' id and name.

The Facebook framework was also built with PHP with an auto login script via a client URL Library (cURL) from PHP.

The script uses the FQL (Facebook SQL query) to capture public posts and data user (name, user-id, genre) by Graph API inside the PHP code.

An auto login script was used with cURL, which is a library that lets one make HTTP requests in PHP. It serves to capture Facebook data, such as geographic location and birthdate, which is not public with the use of FQL.

Users can use these frameworks to access promotions posted on Facebook and Twitter and search others members' opinions about a product or a feeling.

VI. Results and Discussions

To validate the work, the results were compared with the SentiStrength tool. The data were evaluated by three Specialists. Each one analyzed two thousand texts captured from Twitter. One hundred tweets were extracted for each word taken from Google Trends.

Comparing the efficiency of the SentiMeter-Br and the SentiStrength against the Specialists' opinion, a Pearson correlation factor of 0.89 and 0.75 was reached, respectively.

Fig. 5 shows the comparison between the SentiStrength tool, SentiMeter-Br and the Specialists (mean) of the 2000 tweets analyzed. Among the texts collected from Twitter, 67% are negative, 21% are positive, 5% neutral and 7% are spam. The spam texts can be useful for companies to analyze their competitors according to the most often cited keywords in tweets, as can be seen in the word cloud of Fig. 6.

Regarding the tweets extracted, 2000 tweets were analyzed by Specialists, 500 tweets were used to help build the Sentimeter-Br dictionary and another 500 were used for training in Weka to help discover the polarity of the 2000 tweets. The Weka classified the sentences as positive, negative, neutral and spam.

The texts (tweets) analyzed underwent string to vector transformation (filter) in Weka software to be able to analyze the words contained in the texts.

The StringToWordVector takes a string and converts it into a vector consisting of the individual words from that string. This is necessary because the Multinomial Naive Bayes classifier, Decision Tree and SMO do not work directly on text, only with separate words.

In Weka results, as seen in Table II, the use of bigrams and trigrams does not much improve the classification as compared with the use of only unigrams in Table III, but the use of stopwords or non-stopwords improved the correct classification in 18.30% with the SMO algorithm as can be seen in Table II. In all the experiments, 10-fold cross-validation was used to evaluate the classification accuracy, and the results of the F-Measure are shown in Table IV.

Spam and neutral texts have a lower F-Measure because on average (of the searched words in Twitter), only 5% are neutral and 7% are spam. The negative texts had a higher F-measure because most of them were negative words.

F-Measure is used to measure the overall performance, combining precision values and the scope of a model in a single formula. The ideal value for average F is 1.

The results with the SMO algorithm presented the best results. Sentimeter-Br can be associated to the classification by the Weka software with SMO as a way of proving the results.

VII. Conclusions and Future Work

Three Specialists conducted the validation, and the results showed the importance of having a specific dictionary according to a context.

The Pearson correlation factor of 0.89 showed the efficiency of the SentiMeter-Br as compared with the SentiStrength. SentiStrength has limitations for the Portuguese language because it does not use a differential calculation for negative phrases, tenses, and specific idioms.

The best results were obtained with the SMO algorithm with higher F-Measure and Correctly Classified Instances. The results from this algorithm proved the polarity of the tweets analyzed.

As future work, we intend to evaluate the SentiMeter-Br in other contexts (business, education, technology, fashion, health).

TABLE II. Percents of Correctly Classified Instances (CC)/ Percents of Incorrectly Classified Instances (CI) of the algorithms with the use of bigrams and trigrams

	Decision Tree	Naive Bayes	Bayes Multinomial	SMO
stopwords	66.66/33.33	64.95/35.04	65.81/34.18	71.79/28.20
no stopwords	63.24/36.75	66.66/33.33	64.10/35.89	60.68/39.31

TABLE III. Percents of Correctly Classified Instances (CC)/ Percents of Incorrectly Classified Instances (CI) of the algorithms with the use of unigrams

	Decision Tree	Naive Bayes	Bayes Multinomial	SMO
stopwords	66.66/33.33	65.81/34.18	60.68/39.31	70.94/29.05
no stopwords	63.24/36.75	66.66/33.33	64.95/35.04	61.53/38.46

TABLE IV. F-Measure of Rating Methods with bigrams/trigrams and stopwords

Algorithm	Positive	Negative	Neutral	SPAM
Decision Tree	0.74	0.88	0.55	0.65
Naive Bayes	0.75	0.85	0.63	0.65
Naive Bayes Multinomial	0.77	0.85	0.69	0.72
SMO	0.87	0.91	0.75	0.79

A way to capture the geographic location in Twitter framework as in the Facebook framework, will be implemented because the public geocode parameter of Twitter, for security reasons, is not shown, unless the user is logged.

In Brazil, Twitter is mostly used by young people. There is difficulty with slangs, repetition of words and, mainly, grammar mistakes. Hence, the studies with Facebook messages will be repeated both in Sentimeter-Br and in Weka.

More texts with spam and neutral classification have to be collected to improve their F-measure.

ACKNOWLEDGMENTS.

The authors thank the University of Sao Paulo for motivating researches in the Computer and Telecommunication Systems area.

REFERENCES

[1] M.M. Bradley, P.J. Lang, Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical Report C-1, The Center for Research in Psychophysiology, University of Florida (1999).

[2] T. Wilson, J. Wiebe, and P. Hoffmann, Recognizing contextual polarity in phrase- level sentiment analysis. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, Association for Computational Linguistics (2005).

[3] M. Thelwall, K. Buckley, G. Paltoglou, and A. Kappas, Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology 61(12) (2010) 2544-2558.

[4] I. A. Braga, Avaliacao da Influencia da Remocao de Stopwords na Abordagem Estatistica de Extracao Automatica de Termos, 7th Brazilian Symposium in Information and Human Language Technology (STIL 2009), So Carlos, SP, Brazil, pp. 18, 2009

[5] W. Chamlertwat, P. Bhattarakosol, and T. Rungkasiri, Discovering Consumer Insight from Twitter via Sentiment Analysis, Journal of Universal Computer Science, vol. 18, no. 8 (2012), 973-992.

[6] Google Trends, http://www.google.com.br/trends, retrieved 18.04.2013.

[7] F. A. Nielsen, A new ANEW: evaluation of a word list for sentiment analysis in microblogs, Proceedings of the ESWC2011 Workshop on Making Sense of Microposts: Big things come in small packages 718 in CEUR Workshop Proceedings 93-98. 2011 May

[8] E. Kouloumpis, T. Wilson, and J. Moore, Twitter Sentiment Analysis: The Good the Bad and the OMG!, In Fifth International AAAI Conference on Weblogs and Social Media (2011).

[9] Weka 3 - Data Mining with Open Source Machine Learning Software,http://www.cs.waikato.ac.nz/ml/weka, retrieved 18.04.2013.

[10] I. Schwab, A. Kobsa, and I. Koychev, Learning User Interests through Positive Examples Using Content Analysis and Collaborative Filtering, Internal Memo, GMD, St. Augustin, 2001.

[11] B. Liu, Opinion Mining and Sentiment Analysis, WEB DATA MINING, Data-Centric Systems and Applications, Part 2, pp. 459-526, 2011.

[12] Bo Pang, L. Lee, Subjectivity Detection and Opinion Identification. Opinion Mining and Sentiment Analysis. Now Publishers Inc, 2008.

[13] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo, Deriving marketing intelligence from online discussion. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD ’05). ACM, pp. 419-428, 2007.

[14] F. Nielsen, AFINN-96, Department of Informatics and Mathematical Modelling, Technical University of Denmark (2010).

[15] L.K. Hansen, A. Arvidsson, F. A. Nielsen, E. Colleoni, and M. Etter, Good friends, bad news affect and virality in Twitter. Accepted for The 2011 International Workshop on Social Computing, Network, and Services (SocialComNet 2011), 2011.

[16] JSON (JavaScript Object Notation), http://www.json.org, retrieved 18.04.2013.

[17] M. C. Cirelo, R. Sharoviski, F. Cozman, Coup Gagliardi, and M. H. Coup Veerle. Aprendizado de semi-supervisionado de classificadores bayesianos utilizando testes de independncia. Encontro Nacional de Inteligencia Artificial, Campinas, 2003. SBC 2003 ENIA Anais, Cincia, Tecnologia e Inovao - atalhos para o futuro. Campinas, SBC, 2003. 6 p.

[18] J. Platt, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, April 1998.

[19] Interactive iPhone. http://interactiveiphone.com, retrieved 18.04.2013.

ISBN: 978-1-61208-279-0