Automatic Sentiment Analysis of Twitter Messages

Ana C. E. S. Lima
Natural Computing Laboratory
Mackenzie Presbyterian University
São Paulo, Brazil
aceslima@gmail.com

Leandro N. de Castro
Natural Computing Laboratory
Mackenzie Presbyterian University
São Paulo, Brazil
lnunes@mackenzie.br

Abstract— **Twitter® is a microblogging service usually used as an instant communication platform. The capacity to provide information in real time has stimulated many companies to use this service to understand their consumers. In this direction, TV stations have adopted Twitter for shortening the distance between them and their viewers, and use such information as a feedback mechanism for their shows. The sentiment analysis task can be used as one such feedback mechanism. This task corresponds to classifying a text according to the sentiment that the writer intended to transmit. A classifier usually requires a pre-classified data sample to determine the class of new data. Typically, the sample is pre-classified manually, making the process time consuming and reducing its real time applicability for big data. This paper proposes an automatic sentiment classifier for Twitter messages, and uses TV shows from Brazilian stations for benchmarking. The automatic sentiment analysis reduces human intervention and, thus, the complexity and cost of the whole process. To assess the performance of the proposed system *tweets* related to a Brazilian TV show were captured in a 24h interval and fed into the system. The proposed technique achieved an average accuracy of 90%.**

Keywords - Component; Twitter; Sentiment Analysis; Text Mining; Big Data.

I. Introduction

Twitter is a popular microblogging service, founded in 2006, designed for simplified (short messages) communication. The simplified communication and connection accelerates the process of message update. There are about 250 billion messages posted daily in Twitter [1], characterizing this service as an important repository for data analysis, and useful content source for marketers, psychologists and other social interested in the extraction and mining of opinions, views, moods and attitudes [2].

Many companies see this service as an important place to monitor and promote their brands. The monitoring of messages helps companies understand a specific environment and its constant changes, with one basic principle: if something is said in the social media, then it can be qualified and quantified [3]. Quantitative measures can, for instance, provide information about the impact, evolution, dispersion, frequency and sentiment aggregated to the text. On the other hand, qualitative measures are made by teams of analysts according to the goals, needs and characteristics of the project [4].

Similarly to companies and brands, TV stations have realized that Twitter has a corpus of messages that can provide real time feedback about their shows. According to some recent researches [5], 50.6% of Internet users surf the web while watching TV. Thus, by monitoring such messages, they can detect certain aspects on the preferences of their viewers and improve their shows, elaborating more specific marketing campaigns according their audience, and making their TV shows more interactive [6].

Depending on the type of data analysis technique used to explore the social media data captured from Twitter, it is possible to find texts related to one another, subjects discussed in conjunction, and to assign labels to texts based on their characteristics [7]. When the classification task aims to assign a label according to the sentiment expressed in a text, it receives a special name, called *sentiment analysis* [8].

Sentiment analysis corresponds to the determination of the sentiment that the writer intended to transmit in his/her message. This sentiment normally represents the text polarity, that is, whether the message has a positive, negative or neutral sentiment. For companies, brands, famous people and others this measure helps to observe the public and market opinion about itself [9].

In practical terms, the classification task requires a pre-classified database sample, called *training set*, which is either used to generate a classifier (classification model) or to compare with new unlabelled data to be classified. This is important because the classifier accuracy is highly dependent upon such training data. When the application involves social media data, however, this pre-classification is made mostly manually, making the process very time-consuming, subjective and reducing its real-time big data applicability.

An alternative to reduce the operational cost of manually generating training data and making it independent of human intervention is to automate this process. In [10] it was proposed the automatic generation of training data by taking into account the sentiment available in texts containing *emoticons*. Emoticons are graphic representations formed by punctuation marks and letters representing facial expressions, generally included as a means to express the mood of a person. Also, explicitly representing the mood in a message, with words such as, *good*, *excellent*, *worst*, and *terrible*, may improve the sentiment analysis [11]. The proposal of this paper is to use (isolated and in combination) these information, sentiment and explicit declaration of mood, to generate the training set for a sentiment analyzer.

978-1-4673-4794-5/12/$31.00 ©2012 IEEE

The context that we used to assess our proposal is related to TV shows of one Brazilian station. The experiments to be performed will consider three different approaches: *emoticon-based approach*, *word-based approach* and *hybrid approach*.

The paper is organized as follows. Section 2 provides a brief overview of text mining, and Section 3 describes the proposed model for sentiment analysis. Section 4 presents the obtained results and discussions and the paper is concluded in Section 5.

II. TEXT MINING

Text mining corresponds to a set of techniques used to extract patterns or trends implicit in textual databases [12]. Similarly to data mining that searches for patterns in numerical data, text mining is involved in the search for patterns in texts. However, this apparent similarity hides a major difference [13]: while data mining deals with structured data in formalized databases, text mining works with semi or unstructured data; that is, databases that have a heterogeneous or no structure at all, for instance texts, images and voice signals [14].

The text mining process can be divided into four steps [15]: text collection; preprocessing; analysis and validation (Fig. 1).

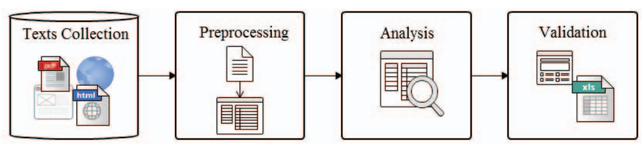

Figure 1. The Text Mining Process (Adapted from [15]).

The first step (collection) consists of capturing texts in their original source, for instance, the Twitter. The preprocessing step is used not only to transform the unstructured data into structured data, but also to prepare it for analysis. In the analysis step, several types of algorithms can be applied to extract relevant knowledge and patterns from the structured data. In the validation step the results presented are assessed and interpreted [16].

A. *Text Collections*

Capturing the texts is the first step of the process and aims at generating the data, also known as *Corpus* or *Corpora*, to be analyzed [7]. This database can be static or dynamic. A static database remains the same throughout the process, whilst a dynamic database can be updated at each instant of time. The update is done by new content addition, removal or the updating of available contents [17]. Generally, texts can be collected from three types of environments: file folders, databases and the Web. Whatever the chosen environment, a *crawler* is responsible for the text collection [17].

B. *Preprocessing*

The preprocessing step starts the text preparation into a more structured representation. This is the most computationally expensive phase, because it requires the processing of unstructured data [18]. This step can be divided into five substeps: 1) tokenization; 2) stopwords removal; 3) stemming; 4) document representation; and 5) feature selection. The end result is a data matrix in which each row represents a text (or document) and each column a term (word) or *token*.

Tokenization is used to identify all words in a given text. These words, called *tokens* or terms, are the basic units of the documents. Formally, there is a set $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_N\}$ of N documents and a dictionary $T = \{t_1, t_2,...,t_c\}$ of c terms [19].

The stopwords removal is used to eliminate those words that occur too frequently, such as articles, prepositions, conjunctions and adverbs. These words are called *stopwords* and depend on the language of the text in question [15].

Stemming makes a linguistic standardization in the token, in which variant forms of this token are reduced to a common form, called *stem*. This process makes the reduction to the root by removing derivatives, genre and plural. This is usually applied to words that have the same meaning, allowing a significant reduction in the dimensionality of the feature vector [15].

After all the previous steps have been implemented, the set of documents that was initially unstructured becomes closer to being structured. What still needs to be done is to choose a means to represent the membership of tokens to documents. The most common model of representation is called *bag-of-words*, which uses the words of the document as features. Thus, the dimension of the feature space is equal to the number of different words in all documents [19]. The membership of a word to a document can be measured, for instance, by its existence or not in a document, by its frequency in the document, or by its frequency in the document taking into account its frequency in the other documents.

The final step in text preprocessing is feature selection. It aims at finding a reduced set of attributes that provides a suitable representation of the database given a certain analysis to be performed. Despite the reduced length of Twitter messages, their informality, the abundant use of slangs, ironies and language mixtures, make the analysis of tweets a substantial challenge [20].

C. *Analysis*

The analysis step is usually considered the core of text mining, because this is when some type of useful, nontrivial knowledge is extracted from the text [19]. The analysis can be classified into two categories: descriptive and predictive. Descriptive analysis characterizes the general properties of the data by means of a characterization and/or discrimination. Characterization promotes a data summarization, whilst discrimination provides descriptive comparisons among collections of the database. The predictive analysis makes inferences about the database in order to make predictions [20].

D. Validation

In order to validate the analyzes performed, it is necessary to employ quantitative and qualitative measures. After such validation it may be necessary to return to one or more of the previous step so as to perform modifications and try alternatives [18].

III. AUTOMATIC SENTIMENT ANALYSIS BASED ON EMOTICONS

Before describing the automatic sentiment analysis method this section brings an overview of sentiment analysis and the proposed method. It then follows with a discussion of the three approaches proposed: emoticon-based, word-based, and hybrid.

A. Sentiment Analysis

It has been long since people seek and observe the opinions of others to direct their behaviors, such as purchases or considerations about a determined subject [18]. The work in [8] argues that most people use a web search about a product or service so as to drive a purchase decision about it.

Sentiment analysis aims at understanding how a reader can interpret the emotion within a text and then transpose it to an algorithm that can perform this task automatically. For example, given a review about a product, the sentiment analysis system determines whether the sentiment expressed in the review has a positive or negative connotation.

One challenging aspect in sentiment analysis is that a feeling can be expressed over anything. Therefore, it is important to identify the object (or subject) that you desire to know the opinion about. To illustrate, consider the following message: "*although the screen is small, the television is amazing*". If the goal is to understand what was the sentiment of the writer about the screen, the object to be analyzed is the screen and the sentiment is negative. By contrast, if the goal is to understand the sentiment about the television as a whole, the object to be analyzed is the television and the sentiment is positive.

Therefore, sentiment analysis aims to understand the opinions expressed on text and classify them into a finite set of categories. Typically, these categories represent the text polarity, which can be *positive*, *negative* and, eventually, *neutral* [21]. This classification problem is defined as follows: a given document $\mathbf{d}_i \in \mathbf{D}$, $\forall i$, is associated with a class belonging to the set $C = \{c_1, c_2, ..., c_k\}$, also called, label or categories. Through a learning method or learning algorithm the classifier learns a function γ that maps each document into one of the classes [22].

The sentiment analysis of Twitter data requires a greater effort from the classifier because tweets are short, have an informal nature and a large variety of subjects, leading to a higher degree of ambiguity. Furthermore, the sentiment is not always obvious in the texts, being ambiguous even for a human reader; and a considerable fraction of *tweets* do not have sentiments, just like those linked to news or announcements.

B. Overview of the Automatic Sentiment Classification

This paper proposes three approaches for the automatic sentiment classification: an *emoticon-based approach*, a *word-based approach*, and a *hybrid approach*. The emoticon-based approach uses the sentiment incorporated in the emoticons as a criterion to automatically classify the messages. The word-based approach uses words that express sentiment as criteria for the automatic classification. The hybrid approach is a combination of the two previous methods.

In order to introduce the proposed system, let ε be the set of emoticons and ω the set of words associated with a given sentiment. Now, let us define s_ε as the support of set ε, and s_ω the support of set ω; where s_j is the percentage of documents that contain at least one term from the set j (ε or ω). The proposed system is composed of three modules (Fig. 2):

1) *Support Counting Module* (SCM): this module is responsible for checking the percentage of *tweets* that contain at least one emoticon from the set ε or word from the set ω. The premise here is that either the emoticons or the sentiment-based words must have a minimal coverage (support), min_*s*, of the set of documents in order to serve as labels to train a classifier. In the experiments to be performed here it will be assumed $\text{min\_s}_\varepsilon = \text{min\_s}_\omega = 5\%$;
2) *Database Selection Module* (DSM): the dataset has to be divided into two sets: training and testing. The training set contains the tweets that will be classified automatically and the test set contains the other tweets;
3) *Classification Module* (CM): this module is responsible for classifying the tweets whose labels are unknown. Among the many classification algorithms available in the literature, we chose to use the naïve Bayes for two main reasons: first, it is a sort of lazy classifier, in the sense that no training is required, only the storage of pre-classified samples; second, because it has been broadly used to classify text due to its nature of dealing with each feature independently [23].

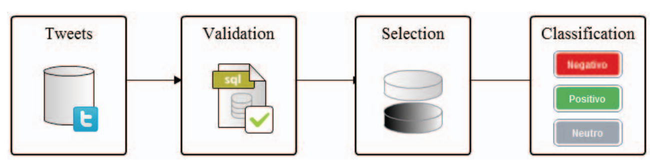

Figure 2. Proposed Classifier Architecture.

C. Emoticon-based Approach

In this approach the criterion to select a tweet for automatic classification is the presence of at least one emoticon. Based on the emoticon used in the message it is possible to infer the sentiment of the text. Table 1 shows a sample of emoticons used in this work.

TABLE I. SAMPLE OF EMOTICONS USED.

Emoticon	**Feeling**	**Sentiment**
:) :-)	Happy	Positive
:(:-(	Sad	Negative
:D :-D	Very Happy!	Positive
D: D=	Very Sad	Negative
_ *.* *-*	Fascinated	Positive
D:< D: D8	Horror, disgust, sadness	Negative
xD XD	Laughing, big grin	Positive
:| =| :-|	Straight face no expression	Neutral

The support counting module checks whether the text corpus satisfies the min_s_{ε} = 5% threshold suggested. In such case, the database selection module divides the corpus in two sets, that with emoticons and that without it.

D. *Word-based Approach*

In the word-based approach the criterion for selecting a *tweet* to automatic classification is the presence of words that express sentiment, such as, good, bad, excellent or terrible. From these words it is possible to infer the sentiment present in the text. These words are used to determine the tweet sentiment and must be created for each sentiment (positive and negative) according to the application.

The support counting module checks whether the text corpus satisfies the min_s_{ω} = 5% threshold suggested. In such case, the database selection module divides the corpus in two sets, that with sentiment-based words and that without it.

E. *Hybrid Approach*

In the hybrid approach emoticons and words are both used as classification criteria. Those tweets that have a word from ω or an emoticon from ε are automatically classified according to the sentiment of the word or emoticon found in the message

Again, the support counting module checks whether the text corpus satisfies the min_s_j = 5% (j = ε, ω) threshold suggested. In such case, the database selection module divides the corpus in two sets, that with emoticons/words and that without it.

IV. PERFORMANCE EVALUATION

This section starts with a brief overview of the Twitter microblogging system. Afterwards, it presents the materials and methods used for assessing the performance of the proposed word sense disambiguation system. Results are then presented and discussed.

A. *A Briefing on the Twitter*

Microblogging are web services designed for simplified and instant communication. This implies in shorter messages when compared to those contained in blogs, and of the presentation in reverse chronological order, such as the blogs. In these services, the process of information update is faster due to the simplified communication and short texts [24].

Within Twitter, the focus is the information, not the user interaction. Therefore, the reciprocity between two followers is not required; the first user can follow the second without being followed by the second [25].

Tweets have a maximum length of 140 characters, and can be used more informally, with slangs and special characters. Therefore, the automatic analysis of Twitter messages has a different difficulty level than that of more formal texts with longer character limits [25].

The monitoring of posted messages is possible through the *Application Programming Interface* (API) of Twitter that provides several methods for data retrieval and access to user information. Internally, the Twitter API is divided into Search API and Stream API. The Search API provides access to a limited set of recent tweets. The Stream API allows access to the flow of messages in real time.

There are libraries [26] available in several programming languages to facilitate access to the Twitter API. Among these, the Twitter4J [27], which is a library for Java applications developed by Yusuke Yamamoto, was particularly used in the development of this research.

B. *Materials and Methods*

The Twitter4J library was used to capture tweets and a search script was written in JAVA to make queries in real time from the 6th to the 7th July 2012, totaling 24 hours of tweets captured. The queried term was about the program "*Agora é Tarde*", which means "Now it is too late" in Portuguese, and 5,349 tweets were collected.

For prediction problems, the results are normally presented as a confusion matrix [28], with a row and column for each class. The confusion matrix contains information about the correct and predicted classifications done by a classifier. The performance of such systems is evaluated using the data in the confusion matrix [29]. Table II shows an example of matrix for a two-class classifier. The correct class is placed in the rows and the predicted class in the columns. TP is the number of correct predictions to the Positive class, TN is the number of correct predictions to the Negative class, FP is the number of Negative class objects predicted as Positive, and FN is the number of Positive class objects predicted as Negative.

TABLE II. CONFUSION MATRIX.

		Predicted	
		Positive	Negative
Correct	Positive	TP	FN
	Negative	FP	TN

The performance measures used to evaluate the classifiers were: Precision (Eq. (1)), Recall (Eq. (2)), and F-measure (Eq. (3)). These measures are used to evaluate how satisfactory are the answers retrieved by an information retrieval system and, thus, suit the purposes of this research. Precision corresponds to the proportion of the predicted positive cases that were correctly classified, and Recall is the proportion of positive cases that were correctly identified. F-measure is the harmonic mean between precision and recall [23].

$$Pr = \frac{TP}{TP+FP} \quad (1)$$

$$Re = \frac{TP}{TP+FN} \quad (2)$$

$$F = \frac{2}{1/Pr + 1/Re} \quad (3)$$

To measure the overall classifier performance the Accuracy of the classifier was calculated (Eq. (4)). It represents the success rate of the classification algorithm and corresponds to the number of correct classifications divided by the number of documents [20].

$$Acc = \frac{TP+TN}{TP+FN+FP+TN} \quad (4)$$

In addition to the accuracy, the *false positive rate* – FPR (Eq. (5)) corresponds to the rate of incorrect classifications made by the algorithm [20].

$$FPR = \frac{FP}{FP+TN} \quad (5)$$

C. Results and Discussion

This section presents the results obtained by each approach. The training set was formed using a stratified approach, balancing the number of positive and negative tweets for training.

1) *Emoticon-based Approach*: in this case, 550 tweets (381 positive and 169 negative) were selected to train the classifier;
2) *Word-based Approach*: 1175 tweets (915 positive and 261 negative) were selected to train the classifier;
3) *Hybrid Approach*: in this case, 1651 tweets (1236 positive and 415 negative) were selected to train the classifier.

Table III summarizes the performance of the Emoticon-based Approach.

TABLE III. EMOTICON-BASED APPROACH PERFORMANCE.

Measure	Result	Measure	Positive	Negative
ACC	0.8956	Pr	0.9897	0.2477
FPR	0.7522	Re	0.9897	0.2477
		F	0.9897	0.2477

Table IV summarizes the performance of the Word-based Approach.

TABLE IV. WORD-BASED APPROACH PERFORMANCE.

Measure	Result	Measure	Positive	Negative
ACC	0.8992	Pr	0.9725	0.6061
FPR	0.6061	Re	0.9736	0.3938
		F	0.9731	0.4774

Table V summarizes the performance of the Hybrid Approach.

TABLE V. HYBRID APPROACH PERFORMANCE.

Measure	Result	Measure	Positive	Negative
ACC	0.9194	Pr	0.9646	0.3923
FPR	0.3923	Re	0.9646	0.6076
		F	0.9646	0.4768

The hit rate for the positive class was 98.97% for the emoticon-based classifier, 97.25% for the word-based approach, and 96.46% for the hybrid approach. By observing the precision for the negative class between the approaches, it was possible to observe significant variations. The word-based approach has achieved the better result, with 60.61%, an increase of 144.69% in relation to the emoticon-based approach and 54.29% relative to the hybrid approach. This variation among the approaches may be explained by taking into account that the emoticons are associated with the sentiment of the user and not necessarily his/her sentiment to the program itself. For example, "*I really wanted to have watched 'Now it is too late'* =\", which is a message that contains a negative emoticon, but transmits a positive sentiment in relation to the program (the viewer desires to watch it).

The false positive rate, that is, the proportion of negative tweets classified as positive, was 75.22% for the emoticon-based approach, 60.61% for the word-based approach and 39.23% for the hybrid approach. In this aspect, the combination of the approaches decreased the FPR, on average, in 73.11%.

The global performance for the emoticon-based approach was 89.56%, for the word-based approach was 89.92%, and 91.94% for the hybrid approach.

V. DISCUSSION

This paper proposed a sentiment analysis system with automatic training based on tweets containing either emoticons or sentiment-based words. These sets were used to categorize the tweets that could not be classified automatically. The technique chosen to classify the unlabelled tweets was the naïve Bayes algorithm. Three different approaches (emoticon-based, word-based, and hybrid) with different criteria for automatic classification were assessed. The results suggested that the combination of techniques provides improved accuracy. One next step of this research is to add the label "Neutral" to the classification.

Acknowledgments

The authors thank Mackenzie University, Mackpesquisa, CNPq, Capes and FAPESP for the financial support.

References

[1] Datasift, “Browse Data Sources - Twitter,” 2012. [Online]. Available: http://datasift.com/source/6/twitter. [Acesso em 29 Abril 2012].

[2] H. Tang, S. Tan e X. Cheng, “A survey on sentiment detection of reviews,” Journal Expert Systems with Applications: An International Journal archive, vol. 36, 2009.

[3] S. Salustiano, “O Profissional Analista,” em Para entender o Monitoramento de Mídias Sociais, 2012, pp. 34-40.

[4] Elife, “Serviços,” Elife, 2012. [Online]. Available: http://elife.com.br/servicos/. [Acesso em 30 May 2012].

[5] Elife, “Estudo Hábitos 2012,” Elife, 2012. [Online]. Available: http://elife.com.br/cadastropapers/?c=estudo-habitos-2012. [Acesso em August 2012].

[6] J. T. V. S. P. S. A. Marasanapalle, “Business Intelligence From Twitter For The Television Media: A Case Study.,” Business Applications of Social Network Analysis BASNA 2010, 2010.

[7] J. R. Carrilho Junior, “Desenvolvimento de uma Metodologia para Mineração de Textos,” Dissetação de Mestrado, PUC-Rio, 2001.

[8] B. Pang e L. L., “Opinion Mining and Sentiment Analysis,” em Foundations and Trends in Information Retrieval, vol. 2, 2008, pp. 1-135.

[9] N. R. S. Filho, “Monitoramento das redes sociais como forma de relacionamento com o consumidor. O que as empresas estão fazendo?,” Gestão Contemporânea., pp. 63-86, 2011.

[10] J. Read, “Using Emoticons to reduce Dependency in Machine Learning Techniques for Sentiment Classification,” Proceedings of the ACL Student Research Workshop, pp. 43-48, 2005.

[11] A. Wolf, “Emotional expression online: Gender differences in emoticon use,” CyberPsychology & Behavior, p. 827–833, 2000.

[12] A. Tan, “Text Mining: The state of the art and the challenges,” In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases, pp. 65-70, 1999.

[13] I. H. Witten, “Text mining.,” em Practical handbook of internet computing, Florida, Chapman & Hall/CRC Press, 2005, pp. 14-1 - 14-22.

[14] R. d. S. Mello, C. F. Dorneles, A. Kade, V. d. P. Braganholo e C. A. Heuser, “Dados Semi-Estruturados,” 2005. [Online]. Available: http://www.cos.ufrj.br/~vanessa/artigos/tutorial.pdf. [Acesso em 23 April 2012].

[15] C. A. Martins, “Uma abordagem para pré-processamento de dados textuais em algoritmos de aprendizado,” Tese de Doutorado, ICMC-USP, São Carlos, 2003.

[16] E. C. N. Barion e D. Lago, “Mineração de Texto,” Revista de Ciências Exatas e Tecnologia, vol. III, pp. 123-140, 2008.

[17] T. Souza e A. Lindgren, “Mineração de Texto,” Revista Eletrônica de Informática da Universidade, vol. 1, 2011.

[18] M. L. Santos, “Protótipo para mineração de opinião em redes sociais: estudo de casos selecionados usando o Twitter.,” Monografia de Graduação, Universidade Federal de Lavras, 2010.

[19] R. Feldman e J. Sanger, The Text Mining Handbook Advanced Approaches in Analysing Unstructured Data, Cambridge, 2007.

[20] J. Han e M. Kamber, Data Mining: Concepts and Techniques, Academic Press, 2001.

[21] S. Batra e D. Rao, “Entity Based Sentiment Analysis on Twitter,” 2010. [Online]. Available: http://nlp.stanford.edu/courses/cs224n/2010/reports/drao-sidbatra.pdf. [Acesso em 15 May 2012].

[22] C. D. Manning, P. Raghavan e H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[23] I. H. Witten, Text mining. In Practical handbook of internet computing, Florida: Chapman & Hall/CRC Press, 2005, pp. 14-1 - 14-22.

[24] A. N. F. Serafimo, C. C. D. D. Cunha e M. P. D. B. Silva, “Redes Sociais e Microblos em Unidades de Informação: explorando o potencial do twitter, do ning e do foursquare como ferramentas para promoção de serviços de informação.,” Anais 33º ENEBD, 2010.

[25] H. Kwak, “What is Twitter, a social network or a news media?,” Proceedings of the 19th international conference on World wide web, pp. 591-600, 2010.

[26] Twitter, “Twitter Libraries,” 6 July 2006. [Online]. Available: https://dev.twitter.com/docs/twitter-libraries. [Acesso em 10 July 2012].

[27] Twitter4J, “Twitter4J - A Java library for the Twitter API,” [Online]. Available: http://twitter4j.org/en/index.html. [Acesso em 20 July 2012].

[28] R. Kohavi e F. Provost, “Glossary of terms,” Machine Learning, vol. 30, pp. 271-274, 1998.

[29] J. Wainer, “Confusion Matrix,” Unicamp, 2 Maio 2008. [Online]. Available: http://www.ic.unicamp.br/~wainer/cursos/1s2012/mc906/Confusion.pdf. [Acesso em 2012 Agosto 13].