Sentiment Analysis in Microblogging: A Practical Implementation

Mauro Cohen, Pablo Damiani, Sebastian Durandeu, Renzo Navas, Hernán Merlino, Enrique Fernández

Departamento de Computación, Facultad de Ingeniería, Universidad De Buenos Aires, Paseo Colón 850, Buenos Aires, Argentina
{litodam, maurocohen, renzoe, sebastiandurandeu}@gmail.com, hmerlino@fi.uba.ar, enfernan@itba.edu.ar

Abstract. This paper presents a system that can take short messages relevant to a particular topic from a microblogging service such as Twitter or Facebook, analyze the messages for the sentiments they carry on, and classify them. In particular, the system addresses this problem by retrieving raw data from Twitter - one of the most popular microblogging platforms - pre-processing on that raw data, and finally analyzing it using machine learning techniques to classify them by sentiment as either positive or negative.

Keywords: Microblogging, sentiment analysis, sentiment classification, opinion mining, information retrieval, text mining, social web, twitter, python, nltk

1 Introduction

In the past few years, social networks have increased their popularity to become the mainstream platforms of the Internet world. An average Internet user nowadays spend more time on social networks than on search engines and e-mail [1]. Among the different social networks types, microblogging networks have gained a strong importance in recent years [2].

Templeton [3] defines microblogging as a small-scale form of blogging made up from short, succinct messages, used by both consumers and businesses to share news, post status updates and carry on conversations. Millions of Internet users use microblogging to talk about their daily activities and to seek or share information. These published messages might also include real-time opinions and feelings on certain topics, for example likes or dislikes statements.

There are different microblogging platforms available today: Twitter [4], Jaiku [5], Tumblr [6] to name just a subset. Among them, Twitter [4] has become the prevalent platform. Since Twitter's inception in 2006, it has grown at an unprecedented rate. In just four years, the service has grown to approximately 20 million unique visitors each month with users sending short 140-character messages (known as "tweets") approximately 40 million times a day [7]. Twitter is a public data source that has been proven a valuable source of information [8].

As Twitter gains popularity, it becomes more useful to analyze trends and sentiment of its users towards various topics. Determining the general attitude of users towards a product or service, for example, can help a business measure overall consumer attitudes, overall satisfaction and feeling about the brand. It can also provide a warning when there is a sudden change in sentiment [9]. As a whole, applying automated tools that attempt to classify tweets into either positive, negative or neutral categories automatically could be quite useful for companies and marketers.

In this context, with the population of social networks and microblogging, new research fields on sentiment analysis - also known as sentiment extraction or opinion mining - have grown considerably and gained special attention lately.

Sentiment analysis, grounded on machine learning techniques, is the task of identifying positive and negative opinions, emotions, and evaluations [10]. It aims to identify the sentiment or feeling in the users to something such as a product, company, place, person and others based on the content published in the web. In the view of the requester, it is possible to obtain a summary about what people are feeling about a topic, without the need of finding and reading all opinions and other news related to it.

Sentiment analysis is mainly a text categorization problem which desires to detect favorable and unfavorable opinions related to a specific topic. Its main challenge is to identify how the sentiments are expressed in text and whether they point a positive opinion or a negative one [3].

This paper presents the base implementation for a sentiment analysis tool that uses machine learning techniques as a Naïve Bayes classifier, to classify tweets by sentiment as positive or negative. The tool was implemented using the IronPython [11] language and an open source language processing library called NLTK [12]. NLTK - Natural Language Toolkit - is a suite of program modules, data sets, tutorials and exercises, covering symbolic and statistical natural language processing. The toolkit is written in Python and distributed under the GPL open source license.

The remainder of the paper is organized as follows: in section 2, a brief description of the state of the art for the main topics covered by this paper. In section 3, the specific problem tackled in this paper. In section 4, the proposed solution, describing in detail the methods and techniques employed. Finally, in section 6, some future directions and areas of analysis for further work.

2 State of the Art

2.1 Social Networks

A Social Network is an abstraction used to describe a social structure as a network, based on the mathematical concepts of graph theory. A network is composed of nodes which will represent the individuals of the network and connections between these nodes, representing the relationships between those individuals (e.g.: friendship, common interests, business links, etc.).

In recent years the term has become of common use, especially with the arousal of one social networking service in particular, with 500+ million users-nodes [13],

Facebook [14]. The latter is the best-known example of an online social networking service which his purpose is to reflect and even create a social network using internet as the place for interaction. This is of tremendous importance for the purpose of this paper because online social networks explicit and persist the topology and interactions of the network.

Other examples of social networking services are: Twitter [4], LinkedIn [15], Hi5 [16], Orkut [17].

2.2 Microblogging and Twitter

A Web log or *blog,* is a site where an individual writes entries (*blogs*) which are visible to anyone who has access to the site, such entries can express thoughts, comments about certain topics, objective information, etc.

Microblogging is a special form of a blog, where the content of the blog is limited in length. Microblogging has become one of the major forms of communication worldwide.

Twitter is the largest microblogging service with 200 million users [18]. Messages, called *tweets*, are limited to 140 characters. This motivates special means of communicating, such as shortening words, extensive use of *emoticons*, and the use of informal language expressions. Examples of twitter messages are shown in Table 1.

The amount of public information present on twitter makes it a unique data source, with the challenge of overcoming the particular language used on it.

Table 1. Examples of Twitter messages.

Jonnyamazing @MikeCjourno #River went down, it was a football nightmare. Like Liverpool going down... suportrs devastated in BA. They take is seriously!
OipiNk RT @BeatlesLane: "Happiness is just how you feel when you don't feel miserable." ~ John Lennon ... - #Beatles
YoliBiebsLove2 RT @Paula_Shawty: Help, we need somebody.. Who help us to put #SomedayInSpain in globals TT :(
iPolhy I'm in blue moon :O! ..... wow I lost my mind ha ha ha xD

2.3 Text Mining and Sentiment Analysis

Marti Hearst [19] gives a clear definition of Text Mining: "*Text Mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources.*"
When the unknown information is subjective information from the writer, determining the topic is itself another aim of text mining. In general, when dealing with the writer opinion, the field of Sentiment Analysis comes into play.

Automated Sentiment Analysis uses merely computer resources to identify this subjective information within the content through the use of Natural Language Processing such as Text Mining. Human and hybrid methods also exist, but this paper emphasizes on automated sentiment analysis.

The basic approach for Sentiment Analysis is determining the *polarity* of a text, the polarity could be: positive, negative or neutral. More fine-grained taxonomies exist with his associated complexity.

2.4 Naïve Bayes Classifier

The Naive Bayes model or Naive Bayes classifier [20] is a simple probabilistic classifier based on the Bayes' theorem with strong independence assumptions. This probability model assumes that the presence or absence of a particular feature of a class is unrelated to presence or absence of any other feature [3]. For further reading on the topic refer to [21].

2.5 IronPython and NLTK

Python [22] is an interpreted, interactive, object-oriented, high-level programming language. IronPython [11] is an open-source implementation of the Python programming language which is tightly integrated with the .NET Framework [23].

Natural Language Toolkit [12] (NLTK) is a suite of open source Python libraries for symbolic and statistical natural language processing (NLP). NLTK defines an infrastructure that can be used to build NLP programs in Python [24]. In particular, for building an Automated Sentiment Analysis application, NLTK provides a set of trainable classifiers [25] including a Naïve Bayes Classifier [20] [26].

3 Problem Definition

The purpose of this paper is to present a system that can accurately classify data coming from microblogging platforms such as Twitter in either positive or negative messages. In general, this type of sentiment analysis can be useful for consumers who are trying to research a product or service, or marketers researching public opinion of their company, among other usages.

While sentiment analysis has garnered great interest recently due to its difficulty as well as potential benefits in trending analysis [27][28][29][30]; sentiment analysis for microblogging messages is a more difficult task due to the nature and inherent limitations of the source data. Among those limitations, worth mentioning:

a) twitter messages are limited to 140 characters per message which motivates users to use creative shortening techniques to express a message through less characters;

b) microblogging messages tend to be more informal in language resulting in a broader, slightly modified and sometimes completely different vocabulary;

c) twitter messages very short in characters length are very common; classification of very short messages is a difficult one;

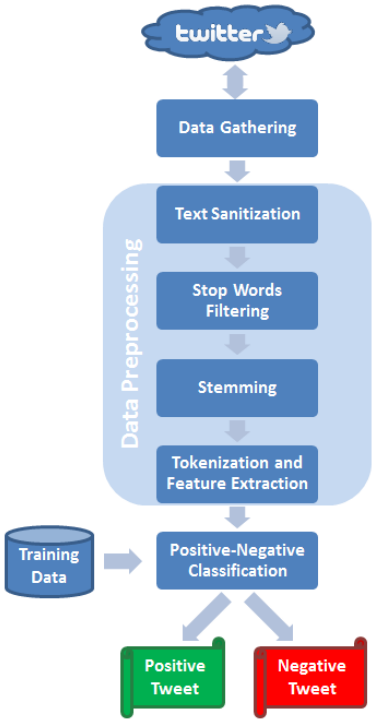
d) ambiguity in microblogging messages; e) emoticons and special characters used within each particular microblogging platform (@RT for instance is specific to Twitter).

The solution proposed on this paper relies on intense pre-processing techniques, some of them with awareness of the specific microblogging platform limitations, in

order to prepare the input before classification with the goal of producing more accurate results.

4 Solution

Figure 2. Solution flow.



The proposed solution, as shown in Figure 1, is based on 3 main stages: Data Gathering, Data Preprocessing, and finally the Positive-Negative Classification. During the Data Gathering, the Twitter messages to be classified are retrieved from its source. Then the Preprocessing stage takes place; this stage is composed by a set of internal steps where the Twitter messages are decomposed into data that is prepared to be analyzed by the classifier. Finally, the Classification stage, which is implemented through a Naïve Bayes classifier that takes some training data as input during its initialization, analyzes the preprocessed twitter messages and returns the twitter messages classified by sentiment as either positive or negative.

4.1 Data Gathering

The Twitter public data API was used to collect a set of tweets for a particular current topic. Over 1500 tweets were gathered and manually classified using two categories: positive or negative. The final purpose of this data is to serve as a training corpus for the classifier.

4.2 Data Preprocessing

After the data has been collected and manually classified, it is passed through a series of preprocessing steps that will be described in this section. After this process finishes, each message is decomposed into a set of features, which in the model used are represented mostly by words that can be taken as input for the classifier.

4.2.1 Text Sanitization

This step executes a set of text processing techniques that sanitize and normalize the messages. Because of the informal and non-grammatical nature of the language used in tweets these steps gain significant importance. These techniques have been proven to improve the quality of the features extracted from the messages and therein improve the performance of the classifier used afterwards [29] [30] [31].

a) Detecting capitalized words. The use of all capital letters in a word is a common method for indicating powerful emotions, and therefore can relate to the message sentiment. Series of capitalized words were identified, adding a special keyword to the message, before removing casing [30].

b) Punctuation. Irrelevant punctuations were removed from the messages for consistency. However, in microblogging it is common to use excessive punctuation in order to convey emotions. So the series of exclamation marks or combinations of exclamation and question marks were replaced with a keyword before removing all punctuation.

c) Lower casing. Because of the erratic casing often found in messages, the messages were turned to lower casing. This is an important step for improving consistency.

d) Replacing emoticons. Many microblogging messages make use of emoticons in order to transmit emotion, making them very useful for sentiment analysis. A range of about 30 emoticons, were replaced with a matching keyword. In addition, variations of laughter such as "haha" or "ahahaha" were all replaced with a particular keyword.

e) Replacing URLs. Many microblogging messages contain URLs in order to share more content than can be given in the limited messages. Since URLs are unique, they were removed from the messages to avoid including them as possible features.

h) Replacing platform-specific characters. Twitter messages use the ' @' character in front of a username to address other users inside the platform. As done before, these pointers were replaced with a special keyword.

i) Removing repeated characters. For emphasizing messages, some words might include repeated characters. These occurrences were compressed to their original form by using regular expressions techniques [32].

4.2.2 Stop Words Filtering

It is also useful to ignore very common words of the messages that do not provide any useful information in the classification. In text mining, these words all called stop words and mainly consists of pronoun, articles, and prepositions and so on. For this step, the English stop word included in NTLK corpus was used.

4.2.3 Stemming

Stemming is the process for removing and replacing common suffixes of English words. This implementation uses the Porter Stemming Algorithm, included in NLTK. The purpose of this step is to reduce the size of the feature set presented to the classifier.

4.2.4 Tokenization and Feature Extraction

The last pre-processing step consists of breaking up the messages into tokens that can serve as input for the classifier. Following usual text mining techniques, performed the tokenization at word level, using a tokenizer included in NTLK, the TreebankWordTokenizer [32].

Then, a Bag of words model was used to transform the list into a feature set that is consumable by the classifier. This simplifying model takes individual words as features, assuming their conditional independence and equality [3]. So the messages are represented by an unordered collection of words, disregarding grammar and even word order. Each feature represents the existence of one word.

4.3 Executing the Classification

For performing the classification, the Naïve Bayes classifier implementation of NLTK was used. The classifier was first trained using the gathered, preprocessed and manually classified data. A subset of the data (1/4) was left apart for testing the classifier and evaluating its accuracy. The classifier, after training, reported an accuracy of 0.73.

Once the classifier was trained, it was used for performing manual experiments by evaluating new messages obtained from the web. Find below a table with the results obtained. A quick analysis of the results, exposes some limitations of the classifier as some messages are erroneously classified.

Table 3. Classification examples.

Twitter Message	Classification
@KKirkscey I fell in love with Coke on long days at my 1st Ironman.	Positive
@littlestclouds GO COKE GO! GO COKE GO!	Positive
I dropped a bottle of coke and said "oww"	Negative
Pepsi and coke are disgusting. Yech.	Positive
I Want A Coke So Badly!	Negative
Cherry Coke will never not be delicious.	Positive
Brandy and coke is the best drink.	Positive
Yay my mom brought me coke :DD I love Coke <3. Best soda ever.	Positive
this coke is the best thing right now.	Positive
omg i LOVE coke now :D	Positive
@CocaCola I love coca cola! Can you bring some to me? My name is Caitlyn grande & I love coke!	Negative
@iBabyPeach texting, tweeting, chatting, listening to music, drinking coke ;)	Positive
@RonanMulcaire coke - always coca cola !!!!	Negative
Isn't that like the greatest coke ad, Laura?	Positive
One caffeine free Diet Coke every now and then is okay, right?	Positive
What would make my day right now is chocolate covered cashews and a coke.	Positive

5 Conclusion

Microblogging has become one of the major forms of online communication and Twitter has positioned as the prevalent platform providing this service. The daily growing amount of information contained in microblogging messages makes them an attractive source of data for sentiment analysis, opinion mining and ultimately global trend analysis.

This paper presents a system that uses machine learning techniques for classifying microblogging messages taking into consideration the particular characteristics and limitations proper of these types of messages.

After applying several preprocessing techniques to the messages, the system tries to classify the messages into either positive or negative depending on the sentiment they carry.

Even if we could not reach the accuracy reported by other systems presented in the literature [28] or available commercially [35], the results obtained are satisfactory.

The machine learning techniques based on statistical models used in this paper, seems to be well fitted for sentiment analysis; the accuracy obtained using the naïve bayes classifier has proven to outperform a manual keyword-based model.

However, as exposed when presenting the results, the system has still much room for improvement. Several simplifications were made for creating the feature set and performing the classification. Some limitations include the inability to detect the relationship between words and different meanings of one word, depending on the context.

The evolution of systems like the one proposed in this paper can be of key importance and value for commercial usages such as marketing researches, trend analysis and public opinion about companies branding; or for social matters, like community embracement of public sector initiatives [33].

6 Future Work

For simplicity purposes, the neutral classification category was disregarded. That is those messages that are not strong enough to be classified as positive or negative. In order to make the classification more accurate, this category should be incorporated.

Additionally, other types of classifiers might be used to enhance the effectiveness of the classification. Although differences are not significant, there is evidence [27] [32] that using other classification techniques as Maximum Entropy (ME) or Support Vector Machine (SVM) can improve the accuracy of the classification.

Other possible enhancement on the preprocessing side, is the inclusion on bigrams on the language model. Bigrams are groups of two words that used together can have a different meaning than used separately. For example, the bigram “not good", which is a negative expression, in the basic bag-of-words model used in this paper could be interpreted as positive, since the classifier sees the present of the "good" word as an indicator of a positive sentiment. However, if considering bigrams, the expression can be correctly classified as negative.

From a data gathering standpoint, the system presented focused just on the Twitter platform, it could also be expanded to support messages coming from other popular social networks, like Facebook.

Finally, from a User Interface perspective, the implementation exposed today is entirely focused on the backend processing. This could be complemented with a subscription-based web site that will allow end-users to sign up for an account, setup their interests and present the sentiment analysis results in a way that can be easily consumed, including compelling graphics and trend analysis [26]. Additionally, on this same perspective, it would be useful to add real time processing of messages, that is constantly retrieving new messages posted by users and classify them instantaneously.

References

1. Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, Bobby Bhattacharjee. Measurement and analysis of online social networks. IMC '07 Proceedings of the 7th ACM SIGCOMM conference on Internet measurement. Year 2007.
2. Selver Softic , Martin Ebner, Herbert Mühlburger, Thomas Altmann, Behnam Taraghi . @twitter Mining #Microblogs Using #Semantic Technologies.
3. Artifical Inteligence in Motion Blog: Working on Sentiment Analysis on Twitter with Portuguese Language, http://aimotion.blogspot.com/2010/07/working-on-sentiment-analysis-on.html, Year 2010.
4. Templeton, M.: Microblogging defined, http://microblink.com/2008/11/11/microbloggingdefined/, Year 2008.
5. Jaiku - your activity stream, http://www.jaiku.com/, (last accessed: July 2011)
6. Tumblr - the easiest way to blog, http://www.tumblr.com/, (last accessed: July 2011)
7. Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. Summarizing Microblogs Automatically, The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Year 2010.
8. http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html
9. Twitter - follow your interests, http://twitter.com/, (last accessed: July 2011)
10. T. Wilson, J. Wiebe, and P. Homann. Recognizing contextual polarity in phrase-level sentiment analysis. Proceedings of the 2005 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 347–354. Year 2005.
11. IronPython, the Python programming language for the .NET framework, http://ironpython.net/, (last accessed: July 2011)
12. Natural Language Toolkit, http://www.nltk.org/, (last accessed: July 2011)
13. Facebook – 500 million stories, http://blog.facebook.com/blog.php?post=409753352130, (last accessed: July 2011)
14. Facebook helps you connect and share with the people in your life, http://facebook.com/, (last accessed: July 2011)
15. LinkedIn - Over 100 million professionals use LinkedIn to exchange information, ideas and opportunities, http://www.linkedin.com/, (last accessed: July 2011)
16. Hi5 – Social Entertainment, http://hi5.com/, (last accessed: July 2011)
17. Orkut – Social Networking, http://www.orkut.com/, (last accessed: July 2011)
18. Twitter co-founder Jack Dorsey rejoins company, http://www.bbc.co.uk/news/business-12889048, (last accessed: July 2011)
19. Marti Hearst. What Is Text Mining? Year 2003.

20. Fernández, Enrique. Análisis de Clasificadores Bayesianos. Trabajo Final de Especialidad en Ingeniería de Sistemas Expertos. Escuela de Postgrado. Instituto Tecnológico de Buenos Aires. Year 2004.
21. Naive Bayes text classification, http://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html, (last accessed: July 2011)
22. Python Programming Language – Official Website, http://www.python.org/, (last accessed: July 2011)
23. The Microsoft .NET Framework, http://www.microsoft.com/net/, (last accessed: July 2011)
24. Natural Language Processing, http://nltk.googlecode.com/svn/trunk/doc/book/ch00.html, (last accessed: July 2011)
25. NLTK Module Classify, http://nltk.googlecode.com/svn/trunk/doc/api/toc-nltk.classify-module.html, (last accessed: July 2011)
26. NLTK NaiveBayesClassifier type, http://nltk.googlecode.com/svn/trunk/doc/api/nltk.classify.naivebayes.NaiveBayesClassifier-class.html, (last accessed: July 2011)
27. S. R. Das and M. Y. Chen. Yahoo! for Amazon: Sentiment extraction from small talk on the Web, Management Science, vol. 53, pp. Year 2007
28. Alec Go, Lei Huang, Richa Bhayani. Twitter Sentiment Analysis, 2009, Ravi Parikh, Matin Movassate. Sentiment Analysis of User-Generated Twitter Updates using Various Classification Techniques
29. Vipul Pandey, C.V. Krishnakumar Iyer. Sentiment Analysis of Microblogs. Year 2009
30. John S. Lewin, Alexis Pribula. Extracting Emotion From Twitter.
31. Suhaas Prasad. Micro-blogging Sentiment Analysis Using Bayesian Classification Methods.
32. Python Text Processing with NLTK 2.0 Cookbook. Replacing and Correcting Words chapter. Jacob Perkins. Packt Publishing. Year 2010.
33. Merlino, Hernán. Ambiente de Integración de Herramientas para Exploración de Datos Centrados en la Web. Tesis de Magister en Ingeniería del Software. Convenio Universidad Politécnica de Madrid - ITBA. Year 2010.
34. Bo Pang, Lillian Lee, Shivakumar Vaithyanathan. Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79-86. Year 2002.
35. Twitter Sentiment API, http://twittersentiment.appspot.com