The role of text pre-processing in opinion mining on a social media language dataset

Fernando Leandro dos Santos
CIC-UnB
University of Brasília
Brasília, Brazil
Email: fernandoleandro1991@gmail..com

Marcelo Ladeira
CIC-UnB
University of Brasília
Brasília, Brazil
Email: mladeira@unb.br

***Abstract*—This work describes an opinion mining application over a dataset extracted from the web and composed of reviews with several Internet slangs, abbreviations and typo errors. Opinion mining is a study field that tries to identify and classify subjectivity, such as opinions, emotions or sentiments in natural language. In this research, 759.176 Portuguese reviews were extracted from the app store Google Play. Due to the large amount of reviews, large-scale processing techniques were needed, involving powerful frameworks such as Hadoop and Mahout. Based on tests conducted it was concluded that pre-processing has an insignificant role in opinion mining task for the specific domain of reviews of mobile apps. The work also contributed to the creation of a corpus consisting of 759 thousand reviews and a dictionary of slangs and abbreviations commonly used in the Internet.**

Keywords*—*opinion mining, text mininig, sentiment analysis, text pre-processing, large-scale data processing.

I. Introduction

Considering the web as a source of information and communication for a significant amount of people, it is natural to think of ways to identify the general opinion expressed in social medias when talking about a product, a company, a service or maybe a political candidate. Each opinion expressed in the web can be positive (favorable), negative (unfavorable) or maybe a varying degree between these two extremes.

This paper describes the development and results of an opinion mining application based on a particular dataset. Opinion mining, or sentiment analysis, is the study field that tries to identify and extract subjective information from non-structured data [12]. The web has many comments, reviews and texts that reflect people's opinions.Thus, one can think of a way to label these opinions in categories such as "positive", "negative" or "neutral", and then, automatically identify each review as belonging to one of these categories.

In this research, large amounts of reviews about Android apps were analyzed. In total, 759.176 Brazilian Portuguese reviews were extracted from the Google Play app store. However, these reviews have some particularities. Since they were extracted from an online platform where any user is able to post, the occurrence of misspellings, slangs and Internet abbreviations are quite frequent, and thus, the analysis become more complex when compared to a well-written text.

Machine learning algorithms are used to train and classify reviews in this work. In this approach, documents are converted to term-document matrices. At this point, pre-processing and filters are applied, words are corrected, abbreviations are expanded, meaningless words (stopwords) are removed and so on. Furthermore, new terms (n-grams) can be created.

In order to enhance performance of the review's classification, different pre-processing stages were applied to the dataset. The main goal of this work is to conclude whether this pre-processing is effective or not. To achieve this overall goal, the project has also two secondary points: veryifying the effectivenes of text pre-processing in opinion mining task with the form of Brazilian Portuguese language typically used in social media and contribute to create a corpus into this domain. Simple data mining tools were unable to handle the complete dataset with the hardware we had available, forcing us to use more robust and complex data mining tools. Thus, machine learning models were created using distributed processing in a cluster of computers, organized with Hadoop.

II. Previous work

The website *Sentiment Analysis with Python NLTK Text Classification*[1] allows users to write a review over a movie and this review can be classified on the fly according to its sentiment. The classifier is based on a corpus provided by Bo Pang and Lilliam Lee [12]. The corpus is composed of movie reviews from IMDb[2]. Naive Bayes, Maximum Entropy and Support Vector Machine (SVM) algorithms were applied and compared. The reviews are written in well-formed English and can be large, having more than one paragraph and many sentences, which is an important difference from the corpus we analyze in this work. Our corpus is composed of smaller and informal reviews.

Farley Fernandes [5] developed a similar work, using restaurant's reviews in Brazilian Portuguese language extracted from Twitter. A linguistic approach was applied, trying to identify polarities by analyzing syntactic characteristics of words as well as natural language processing techniques. It combines adjectives and adverbs with negations or adversative sentences in order to analyze the overall polarity. To identify syntactic attributes of words, a POS tagger is used and its performance is highly correlated with the formality of the language. Continuing the research, Nelson Silva [2] introduced a lexical approach to the same system. He used a list of

[1]http://text-processing.com/demo/sentiment
[2]http://www.imdb.com

978-1-4799-5618-0/14 $31.00 © 2014 IEEE
DOI 10.1109/BRACIS.2014.20

words previously classified according to its polarity, called *SentiWordNet*[3]. With this corpus, he was able to suppress the manual effort of classifying adjectives and adverbs and maintain the same performance.

The role of pre-processing and normalization of reviews were also studied by Duran et. al. [4]. Over a dataset of product reviews in Portuguese, they observed the improvement in precision of a POS tagger after applying each of the following steps: case correction, punctuation correction, spelling and Internet slang correction. Indeed, in a POS tagging task, case correction has an important role when dealing with named entity recognition. While they tackled the problem of POS tagging, in this work, we use similar pre-processing techniques and verify their performance in machine learning methods.

The studies [15][3][7] explore sentiment analysis using machine learning methods, such as classifiers, within reviews extracted from Twitter. Naive Bayes, SVM, Logistic Regression and decision trees are frequently used as classifiers. Also, different tests with n-grams are performed in order to fit better models. In most cases, better results are encountered with uni-grams or bi-grams [12][7]. Some studies [7][15] also explored *emoticons* present in *tweets* to improve classifier's performance. It is worthwhile to say that these studies are not focused in pre-processing, but on comparing different machine learning approaches.

There are still few studies regarding sentiment analysis involving large amounts of text. In [9], around 300.000 *tweets* were analyzed in order to develop a real time sentiment classification tool. They used *emoticons* to pre-define the polarity of the *tweets* and train them, considering that happy *emoticons* usually appear in positive sentences and sad *emoticons* tend to appear in sad sentences. They used Apache Hadoop and Apache Mahout for the task. In [10], Lin and Kolcz used Hadoop and Apache Pig to classify even larger datasets. After tests performed with 1, 10 and 100 million of *tweets*, it was concluded that after a while, the improvement of performance by increasing the amount of training samples was very low. For instance, the classifier performance increased less than 0.5% when using 100 million of tweets instead of 10.

III. Experiment

The study case conducted in this research involves two phases. First, a small amount of the total dataset was analyzed using Weka[4]. In this phase, 10.000 reviews were manually labeled to be used for training and testing classification algorithms. Some initial conclusions were made, and in the following moment, the whole dataset was involved to confirm the results encountered intially. In this second phase, a cluster of computers was created using Hadoop and the framework Mahout was used to create classification models using the map-reduce paradigm[13].

A. Dataset selection

Every review available at Google Play app store has a rating based on stars together with its text. This rating indicates how much the user enjoyed the app and can vary from 1 to 5 stars. First, it was needed to verify whether this stars rating is compatible with a manual classification made by a human. A stratified sample of 10.000 reviews were read one by one, and classified into "positive", "negative" and "neutral". This manual labeling work was done by one of the authors and another researcher, enabling them to decide later how to distribute stars rating into classes. Using the Student's t-test as a hypothesis test, we concluded that the stars rating is similar and equivalent to a manual classification, turning possible, thus, to use the stars rating to train and classify the entire dataset. This first phase of the analysis was done according to the dataset described in Table I.

TABLE I. Dataset used for the initial analysis

Corpus	Reviews	Classification
Corpus A	Initial 10.000	Manually
Corpus B	Corpus A	Stars rating
Corpus C	Another 10.000	Stars rating

The corpora A and B were used for validating the stars rating. A new corpus C, extracted from the entire corpus, was used to verify how the classifiation model, induced with corpus B, would play with these new data.

To achieve better classification results, in both analyses, balancing techniques such as resample and undersample were tested. Thus, in the first phase, corpora A and B, which were used for training, had their minority classes resampled, leaving corpus C intact, since it was used for testing. Table II shows the class distributions of these corpora.

TABLE II. Class distributions on corpora A, B and C

Corpus	Negative	Neutral	Positive
Corpus A	16.31%	21.93%	61.76%
Corpus B	16.31%	21.93%	61.76%
Corpus C	12.89%	6.14%	80.97%

To the final analysis, the whole dataset composed of 759.176 reviews was used. Due to its own nature, the reviews are not well balanced among its classes. Table III shows how stars are distributed in the entire corpus.

TABLE III. Stars distributions among all reviews

(%)	1*	2*	3*	4*	5*
Reviews	16.01%	3.88%	8.30%	13.43%	58.38%

After reading 10.000 reviews, it was decided to split stars ratings into classes. Reviews with 1 or 2 stars were set to "negative", reviews with 3 stars were set to "neutral" and reviews with 4 or 5 stars were set to "positive".

B. Pre-processing

Three steps of pre-processing were sequentially applied to the corpora. Thus, four different sets of corpora were created where each level has an extra pre-processing step. The first one is the data without any pre-processing and the following three are formed according to the following tasks, respectively:

- *Terms standardization*: eliminate accentuation, punctuation, special characters and numbers. All letters are converted into lowercase letters.

[3]http://sentiwordnet.isti.cnr.it

[4]http://www.cs.waikato.ac.nz/ml/weka

- *Spell check*: eliminate spelling errors in the text to be analyzed, using dictionaries.
- *Stemming*: eliminate variations from words, such as plural forms, gerunds or temporal suffixes, in order to reduce each word to its stem.

Thus, we have our dataset divided in the following stages:

- Stage 1: reviews without any pre-processing
- Stage 2: reviews with terms standardization
- Stage 3: reviews with terms standardization and spell check
- Stage 4: reviews with terms standardization, spell check and stemming

Stopwords removal is applied in all stages in order to reduce the dimensionality of the model.

Due to the large occurrence of slangs and internet abbreviations, normal spelling check tools, such as Aspell[5], have had a bad performance in the spell check task. To solve this issue, a dictionary of 2000 terms [6], including slangs and abbreviations, was created.

Based on a dictionary and association rules for Brazilian Portuguese available in Aspell, it was possible to generate a list of all words of this language. This list was compared to each word present in the dataset, and thus, the "incorrect" words, such as slangs and abbreviations, became visible. After filtering and ordering these terms by frequency, we ended up with a list of the most common slangs and abbreviations present in this specific domain of reviews. The first 2000 terms of this list were selected, and its respective correction was manually added, forming a dictionary for the specific internet terms domain.

With the direct use of the Aspell checker, bad results were found. Since the domain of the dataset are related to games and cellphones, many English words such as "android", "tablet" and "app" are considerably frequent in the reviews. Thus, the Brazilian Portuguese checker was treating these words as being wrong and converting it to other Portuguese words.

In order to perform the stemming, it was used the PTStemmer[7].

C. Knowledge extraction: Datasets A, B and C

The knowledge extraction is made in two steps. First, the pre-processed text is converted into a term-document matrix and then, classification algorithms are trained and tested with this matrix.

During the generation of the term-document matrix in the first part of the analysis, the following techniques were considered:

- A TF-IDF [14] model was used for dimensionality reduction
- Portuguese stopwords were removed
- Only terms with a minimum frequency of 5 were considered
- Uni-grams and bi-grams were considered

Undersampling and oversampling [1] techniques were applied to the training dataset, since the class distribution of the dataset is not balanced.

Once the term-document matrix was constructed, the next step was to apply classification algorithms to it. Before that, an InfoGainAttributeEvaluator filter of attribute selection was applied to the matrix in order to select the best terms [6].

Neural Network, Bayesian network, Naive Bayes and SVM classifiers were tested in this phase. All these tests were conducted following a 5-fold cross validation. In the end, the best results were achieved with SVM classifiers.

D. Knowledge extraction: Complete dataset

In order to analyze the complete dataset, a cluster of 6 computers[8] was configured with Hadoop. Since Mahout and Hadoop follow a key-value pair data representation, the reviews are converted into sequential files where the polarity of the review is a key and the review itself is the value.

The reviews are also converted into a term-document matrix as it is converted in the previous phase. Uni-grams and bi-grams are selected, with a minimum term frequency of 5. The key becomes an identifier of the review and the value becomes a list of TF-IDF weights.

Tests conducted with Naive Bayes output good results when applied in the train dataset, however, very disappointing results when applied to a test dataset. The probability $P(c)$ — estimated as the frequency of a class c over all classes — is used in the Naive Bayes algorithm. Once the frequency of "positive" reviews is much higher than the others, the classifier tends to classify more instances as "positive". Undersampling and oversampling techniques were applied in order to solve this problem, without any success, though. In cases where minority classes were oversampled to match to the majority class, the classifier became heavily addicted to minority classes due to repeated occurrences and, thus, also resulted in low performance classifications.

Tests with a logistic regression implementation were also conducted and showed good results. During these tests, the complete dataset was split into three groups: train, validation and test. Train datasets were used to train the algorithm and validation datasets were used to test different parameters. Once the best parameters were defined, the test datasets were used to evaluate the final performance of the model. The train, validation and test datasets were split, respectively, with 60%, 20% and 20% of the 759.176 reviews.

IV. Results

The best performance for the classification was encountered using the SVM in the first phase, where the corpora A, B and C were analyzed, and the logistic regression in the second

[5]http://aspell.net/

[6]Corpus and dictionary available by email

[7]https://code.google.com/p/ptstemmer/

[8]HP Machine Intel i5, 8GB RAM

phase, where the complete corpus was involved. Tables IV and V present the results of the SVM model applied to corpora A and B, respectively. The values are the average after applying a 5-fold cross validation.

TABLE IV. RESULTS OF SVM CLASSIFICATION - *corpus A*

Pre-processing level	Accuracy	F-Measure
Stage 1	81.6482%	0.816
Stage 2	81.9538%	0.818
Stage 3	**82.6644%**	**0.827**
Stage 4	82.0538%	0.819

TABLE V. RESULTS OF SVM CLASSIFICATION - *corpus B*

Pre-processing level	Acurácia	F-Measure
Stage 1	81.2081%	0.779
Stage 2	**81.3232%**	**0.78**
Stage 3	81.2031%	0.78
Stage 4	81.163%	0.775

Contrary to expectations, by analyzing tables IV and V, it was noticed that performance improvement between stages is not very significant. This behavior was not expected, since in each new stage the amount of words is reduced, thereby reducing the dimensionality treated by the algorithm. From stage 1 to 2, for example, terms like "não" and "nao" would be treated as the same, as well as terms "exelente" e "excelente" from stage 2 to 3. It was discussed in [11] that "standard machine learning classification techniques, such as support vector machines (SVMs), can be applied to the entire documents themselves", i.e., without any pre-processing. Emma Haddi et. al. [8] in "The Role of Text Pre-processing in Sentiment Analysis" also obtained poor improvements after applying different types of pre-processing.

A Student's t-test was applied to the F-measures average between corpora A and B. With this test, we could verify that there was no significant difference between manually classifications and classifications made by stars. Therefore, it is verified that the stars ratings are similar to a manual classification and can be used as a valid label for machine learning algorithms. Tests were conducted for the 4 stages and the hypothesis was accepted in all of them. The P-values for these tests are showed in Table VI. A confidence level of 95% and 3 degrees of freedom were used.

TABLE VI. P-VALUE (STUDENT'S T-TEST)

	Stage 1	Stage 2	Stage 3	Stage 4
P-value	0.8102079	0.8108347	0.8175913	0.7871466

The corpus C was created in order to validate the model induced with Corpus B and check whether it is generalizing well for new data. Tests over this corpus were executed and their results are showed in Table VII. The model generalized well when applied to new reviews. However, the same behavior was found: no significant improvement for any of the pre-processing techniques.

The logistic regression classifier output the best results for tests conducted on the entire dataset. Table VIII contains the results of the logistic regression classifier applied to the test set (20% of the entire dataset).

TABLE VII. RESULTS OF SVM CLASSIFICATION - *corpus C*

Pre-processing level	Accuracy	F-Measure
Stage 1	80.56%	0,819
Stage 2	80.5945%	0,828
Stage 3	**81.0849%**	**0,832**
Stage 4	80.1141%	0,745

TABLE VIII. LOGISTIC REGRESSION RESULTS

Pre-processing level	Accuracy	F-Measure
Stage 1	80.0843%	0,776
Stage 2	82.0331%	0,794
Stage 3	**81.2961%**	**0,796**
Stage 4	81.4463%	0,786

As expected, the tests output the same conclusions from the initial analysis. We observe a gradual and slow improvement of the F-measure after each stage, with an exception for the stage 4. In fact, stemming the words reduced the final performance of the classifier for this dataset.

V. CONCLUSION AND FUTURE WORK

In this work, opinion mining techniques have been applied to a dataset of reviews of applications for mobile devices, which are composed of slangs, abbreviations and jargons from the Internet, in order to classify them regarding their positive, negative or neutral orientation. Pre-processing techniques were gradually applied to these reviews in favor of assessing the impact of such techniques on the performance of classifiers. Opposed to expectations, the proposed pre-processing methods have not resulted in a significant improvement.

Each review analyzed in this research has a star rating between 1 and 5. Using hypothesis tests, we concluded that this rating can indeed represent a positive, negative or neutral polarity. Hence, it is verified that reviews about apps from Google Play are a valid source of pre-labeled data for opinion mining.

A possible reason for the low efficiency found for pre-processing might be the size and the small number of sentences in each review. Thus, experimentations with a different kind of dataset is the type of problem we will be addressing in the future.

REFERENCES

[1] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[2] Nelson G. Rocha da Silva. Bestchoice: Sentiment classification on opinion expression tools (original title in portuguese). Graduation thesis, Universidade Federal de Pernambuco, Recife, 2010.

[3] R. de Groot. Data mining for tweet sentiment classification. Master thesis, Utrecht University, Utrecht, 2012.

[4] Aluisio S.M. Pardo T.A.S. Nunes M.G.V. Duran M.S., Avanço L.V. Some issues on the normalization of the corpus of products reviews in portuguese. In *Proceedings of the 9th the Web Corpus Workshop (WAC-9) EACL*, pages 22–27, 2014.

[5] Fernandes F. A framework for sentiment analysis on social network product reviews (original title in portuguese). Master's thesis, Universidade Federal de Pernambuco, Recife, 2010.

[6] E. Frank, M. A. Hall, G. Holmes, R. Kirkby, B. Pfahringer, and I. H. Witten. *Weka: A machine learning workbench for data mining.*, pages 1305–1314. Springer, Berlin, 2005.

[7] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *Processing*, pages 1–6, 2009.

[8] Emma Haddi, Xiaohui Liu, and Yong Shi. The role of text preprocessing in sentiment analysis. *Procedia Computer Science*, 17:26–32, 2013.

[9] Vinh Ngoc Khuc, Chaitanya Shivade, Rajiv Ramnath, and Jay Ramanathan. Towards building large-scale distributed systems for twitter sentiment analysis. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 459–464, New York, NY, USA, 2012. ACM.

[10] Jimmy Lin and Alek Kolcz. Large-scale machine learning at twitter. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 793–804, New York, NY, USA, 2012. ACM.

[11] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *In Proceedings of the ACL*, pages 271–278, 2004.

[12] Bo Pang, Lillian Lee, and Shivakumar 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*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[13] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, Cambridge, 2012.

[14] Pascal Soucy and Guy W. Mineau. Beyond tfidf weighting for text categorization in the vector space model. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, pages 1130–1135, San Francisco, CA, USA, 2005. Morgan Kaufmann Publishers Inc.

[15] Shruti Wakade, Chandra Shekar, Kathy J Liszka, and Chien-Chung Chan. Text mining for sentiment analysis of twitter data. In *2012 International Conference on Information and Knowledge Engineering*, IKE'12, pages 109–114, 2012.