Polarity Classification of Traffic Related Tweets

Clarissa Castellã Xavier[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

ccxavier@inf.ufrgs.br

Abstract. *In this paper we present a study about polarity classification of tweets in the traffic domain. Specifically, we use the data in Portuguese language from an account maintained by a traffic management agency. We evaluate the performance of three learning methods: SVM (Support Vector Machine), Naive Bayes and Maximum Entropy. We also explore how the use of balanced vs. unbalanced corpus affects the models behavior. The results show that, in this context, a ML classifier obtains better results than the reported in the literature. In our experiments, SVM trained with a balanced corpus outperforms all tested models, achieving 99% of Accuracy, Average Recall and Average Precision.*

1. Introduction

As the number of vehicles circulating in the cities expands, traffic problems increases. Traffic jams are becoming a common problem in megacities of several developing countries. In these countries, normally, the traffic monitoring infrastructure is poor and the authorities do not have the resources to collect data [Aching. et al. 2014]. Therefore, mechanisms that help the management of urban traffic have become essential. One option is to build tools that get data from social networks and generate traffic databases for use of urban planners.

For such purpose, one of the possibilities would be to obtain data from microblogs, as Twitter, and feed datasets with information that can be used to develop better transportation services and traffic policies. In this context, it is important to distinguish among these data, if the information is negative, positive or neutral. In this work we address the problem of how to classify the polarity of tweets related to urban traffic.

The main task of a sentence level sentiment analysis is to find the polarity of the sentence. Lalrempuii and Mittal [Lalrempuii and Mittal 2016] explain that "*for example, a positive sentiment is assigned when the tweet indicates happiness, excitation or sympathy and a negative sentiment may relate to anger and sadness*" and that "*when there are no emotions implied, the text may be neutral*".

However, this definition doesn't fit for traffic behavior classification. In this case, sentences that report a favorable situation, such as free traffic, are considered positive. Sentences that report an unfavorable situation, such as a traffic jam, are considered negative. Sentences that report an event with unknown consequences, such a new traffic light, are considered neutral. For this reason, Cao *et al.* [Cao et al. 2014] propose Traffic Sentiment Analysis (TSA) as a subfield of sentiment analysis, which concerns about the issues of traffic in particular, arguing that TSA should treat the traffic problems in a new angle.

The work reported in literature [Cao et al. 2014, Karthik 2017] uses rule-based methods to perform TSA. In this work we investigate if Machine Learning (ML) techniques are suitable for this task. We use and compare the following classification models[1]: Maximum Entropy, Naive Bayes and SVM. We don't use recursive neural networks methods, as Deep Learning, in view of the fact that they require a very large amount of training data for being efficient [LeCun et al. 2015]. In the current context of our research, we do not have enough data to benefiting from the advantages of this approach.

The corpus used in the models evaluation is composed by Tweets written in Portuguese language. We didn't find in literature any studies reporting polarity classification of traffic related texts from social media in Portuguese, which adds importance to this work.

The paper is arranged as follows. Section 2 summarizes the research that based this work. The learning approach and the strategy to evaluate its performance are described at Section 3. Section 4 presents how the performance analysis of the classification models was conducted and its results. Finally, Section 5 provides concluding remarks.

2. Related Research

In this section we discuss related work in three directions. First, TSA implementations. Next, the use of ML to extract information from traffic-related posts generated by traffic authorities. Finally, the use of ML classifiers to classify Tweets from any domain.

Sentiment Analysis is a well-known Natural Language Processing (NLP) task. Given a text, the aim is to get the information sentiment polarity. In this research field, TSA is designed to specifically classify traffic-related information. TSA was first proposed by Cao *et al.* [Cao et al. 2014]. This paper presents a system that uses a rule-based approach to classify Twitter-like posts in Chinese as positive or negative. The authors work with two experimental data sets containing 547 positive and 5937 negative messages on topic 1 (yellow light rule) and 516 positive and 7418 negative messages on topic 2 (fuel price). Besides having a dataset with only traffic related posts, the data came from all kind of account, which increases the diversity of writing styles in relation to the corpus used in our work. In contrast, they limited the subject of the posts in the experiments, which makes the corpus more cohesive. They report the following results: for topic 1, 30.52% of Recall, 84.64% of Precision for negative and 98.31% of Recall, 82.25% of Precision for positive and 82.45% of overall Accuracy. For topic 2, 25.86% of Recall and 90.5% of Precision for negative and 99.2% of Recall and 81.95% of Precision for positive and 82.51% of overall Accuracy.

Another work about TSA is presented by Karthik [Karthik 2017]. Besides containing few informations, we believe it is important to present this work here, since it is the unique work about TSA we found in the literature, besides Cao *et al.* [Cao et al. 2014]. The proposed approach works by computing 5 input parameters (mean, maximum value, minimum value, standard deviation and variance) for each extracted word and classifying it as positive or negative. Unfortunately, the article do not inform how the tests were performed and what corpus was used on it. The reported results are 94% of Accuracy and 34% of Recall.

Although not performing TSA, Albuquerque *et al.* [Albuquerque et al. 2016] evaluate ML techniques for interpreting Tweets in Portuguese language. Their data classifi-

[1] The terms *classifier* and *classification model* indicate the same concept in this paper.

cation is based on learning techniques and they analyze traffic-related posts generated by traffic authorities and news agencies, as we do, reinforcing the feasibility of getting data about urban traffic from official Twitter accounts. They propose a domain ontology that models traffic-situation as events and an automatic tweet interpretation tool. They perform two tasks using ML: entity extraction and relation extraction. The first was implemented using a SVM classifier and second one using a ML algorithm whose name was not explicitly stated. They evaluate the results using a corpus of 690 manually annotated Tweets. The relation extraction of 200 Tweets achieved a mean 73% Accuracy. Using a 10-fold cross-validation, the entity extraction task achieved accuracy higher than 70% in the majority of cases.

Also regarding the use of ML classifiers to perform polarity classification, SemEval-2017 Subtask A [Rosenthal et al. 2017] proposes that given a tweet, decide whether it expresses positive, negative or neutral sentiment. As it performs the classification of Tweets using ML methods, it is very important for us to understand how the participating studies worked. However, they work with very large corpus. SemEval-2017 dataset had 50,333 training and 12,284 test multiple topics Tweets. For English the best ranking teams achieved an Average Recall of 68.1%. The top teams used deep learning and three of the top-10 scoring teams used SVM classifiers.

Besides having very clear purposes and being useful for several traffic-related systems, TSA is a field that has not being much explored in literature. For this reason, we believe that a work like ours, that proposes the use of a different approach for doing it, in a language that has not yet been used for this task, is very relevant.

3. The Approach

In this section we justify the use of learning models to perform the classification of traffic-related Tweets and the strategy to evaluate their performance.

3.1. Learning Approach

Our goal is to perform polarity classification of Tweets created by traffic agencies. Both rule and leaning-based approaches can be used for this task.

Rule-based approaches are context [Cao et al. 2014] and language dependent. They are also independent of clauses sizes and since the syntax of a language does not change, the process and word choice basically remain unchanged. In that way, the rules are relatively static.

In other hand, ML requires large amount of training data and it is often more computationally expensive in terms of CPU processing, memory requirements, and training/classification time. However, according [Gilbert and Hutto 2014] *"because manually creating and validating a comprehensive sentiment lexicon is labor and time intensive, much work has explored automated means of identifying sentiment relevant features in text*. Several practices incorporate ML approaches to classify text polarity.

We decided to explore learning-based methods for classifying the polarity of the posts. In this way, we will evaluate the following classifier: Maximum Entropy, Naive Bayes and SVM.

3.1.1. Maximum Entropy:

According [Ratnaparkhi 1997] the principle of maximum entropy states that "*the correct distribution is that which maximizes entropy or uncertain subject to the constraints which represent "evidence", i.e., the facts known to the experimenter*". This principle is often invoked for model specification, assuming that the observed data itself is the testable information.

3.1.2. Naive Bayes:

It is a probability model that assumes independence among the input features, based on the application of the Bayes' theorem. It assumes that the presence of a particular feature is unrelated to the appearance of any other feature [John and Langley 1995].

3.1.3. SVM:

Support Vector Machine is a supervised classification algorithm that has been extensively and successfully used for text classification task [Pawar and Gawande 2012]. Given a set of training examples, each marked as belonging to one category, a SVM training algorithm builds a model that assigns new examples to a category. A SVM model is a representation of the examples as points in a hyper-plane. The best hyper-plane for a new sample is the one with the maximum margin from the training samples and is computed based on the support vectors [D'Andrea et al. 2015].

3.2. Strategy

The imbalance between classes is considered an obstacle for classifiers in imbalanced domains. However, it has also been observed that in some domains, the use of imbalanced training sets generated good results [Batista et al. 2004]. For this reason we decided to test the models using both balanced and unbalanced data sets, in order to identify which is the most appropriate approach within our problem.

In the ideal situation we would have enough data to train and validate the models and have separate data for assessing the quality of each classifier [Krstajic et al. 2014]. However, that is not our case. In case of lack of relevant problem-specific knowledge, a cross-validation approach may be used to select a classification method empirically [Schaffer 1993].

Cross-Validation is a method of evaluating and comparing learning algorithms by dividing data into two segments: one used to train and the other to validate the model. In typical cross-validation, the training and validation sets must cross-over in successive rounds such that each data point has a chance of being validated [Refaeilzadeh et al. 2009]. In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. In this work we perform a 10-fold cross-validation, like [Albuquerque et al. 2016]. The 1/4 - 3/4 proportion of the corpus is maintained on each fold, as well as the ratio for each polarity.

4. Performance Analysis and Results

In this section we present how we conducted the performance analysis of the classification models and the obtained results. The work was conducted in three main steps: corpus elaboration, tests execution and results evaluation. Each step is detailed bellow.

4.1. Corpus

The corpus used in this study is composed by Tweets written in Portuguese language from @EPTC_POA [2] Twitter account. The account is maintained by the public agency with the purpose of informing citizens about the city traffic.

We extracted posts generated from October 25, 2017 to January 23, 2018.

The Tweets were manually classified as positive, negative or neutral. We labeled as positive, posts that report a good situation or the solution of a problem, as negative, posts that report a problem and as neutral, posts not related to traffic or general situations. Table 1 shows one example for each classification.

Table 1. Example of negative, positive and neutral @EPTC_POA tweets.

Classification	Example	Translation
Positive	Trânsito é tranquilo também no túnel da Conceição nos dois sentidos da via.	Transit is also smooth in the Conceição tunnel in both directions.
Negative	Neste momento, bem complicado o acesso a Rodoviária no Largo Vespasiano Julio Veppo, pelo Túnel da Conceição.	At the moment, the access to the Bus Station at Vespasiano Julio Veppo Square is very complicated by the Conceiçção Tunnel.
Neutral	ATENÇÃO: Linhas de ônibus terão alterações a partir deste sábado, 27.	ATTENTION: Bus lines will change from this Saturday, 27.

On total, 3950 Tweets have been extracted and classified. Of these, 752 were classified as positive, 1756 as negative and 1442 as neutral.

4.1.1. Corpus Balancing

As previously mentioned, the classifiers are tested using the unbalanced corpus and a balanced version of the same corpus. This balancing is performed using a random oversampling technique. In random oversampling, minority class examples, Tweets in our case, are randomly duplicated to balance the datasets. Thus, in the balanced set, all test corpora have the same number of sentences: 1756.

[2] EPTC is the Empresa Pública de Transporte e Circulação (*Public Transport and Circulation Company*) from Porto Alegre city.

4.2. Implementation

The posts extractor and classifiers were implemented in Python [3] programming language. The Twitter posts extractor was implemented using Twitter API [4]. The corpus balancing was implemented with Scikit-learn [5]. For Maximum Entropy, Naive Bayes and SVM models we used NLTK [6] library.

4.2.1. Classifiers Training Features

During training, each input value must be converted to a feature set. These feature sets capture the basic information about each input that should be used in the classification. Pairs of feature sets and labels are fed into the machine learning algorithm to generate a model [Bird et al. 2009].

Machine learning classifiers typically require the text input to be represented as a fixed-length vector. Perhaps the most common fixed-length vector representation for texts is the bag-of-words due to its simplicity, efficiency and often surprising accuracy [Le and Mikolov 2014]. In this model, the text is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.

Therefore, the experiments reported in this article use bag-of-words model as feature set.

4.3. Results and Evaluation

To evaluate the results we employed the most frequently used statical metrics: Accuracy, Precision, Recall and F-score. We also use Average Recall and Average Precision (*AvgRec, AvgPrec*) calculating the Recall and Precision averaged across the positive, negative and neutral classes [Rosenthal et al. 2017].

As stated at Section 3.2, our strategy is to evaluate the classification models using 10-fold cross-validation. We executed this evaluation using a balanced and an unbalanced corpora. The results are summarized at Table 2. The first column informs the tested corpus (unbalanced / balanced). Column 2 shows the classification method. Column 3 presents what metric corresponds to columns 4, 5 and 6 values, which relate to negative, positive and neutral classification, respectively. Column 7 shows Accuracy, column 8 Average Recall and column 9 Average Precision.

All classifiers present an improvement in its results working with the balanced corpus. This means that, in the context we are working on, this is the best strategy for training.

SVM is the classifier with the best results. It got an Average Recall of 99% in the test with the balanced and unbalanced corpus, higher than the one obtained by the system with the best result at Sem-Eval 2017 [Rosenthal et al. 2017] (68.1%). We assume that this difference of results happens due to the fact that we work in a defined domain and only with texts generated by the same account, that does not usually use informal language.

[3] *https://www.python.org/*

[4] *https://developer.twitter.com/en/docs/api-reference-index*

[5] http://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html

[6] *http://www.nltk.org/*

Table 2. Results for each classification method. First, all results for unbalanced corpus, followed by all results for balanced corpus. Data ordered by Average Recall (higher is better).

Corpus	**Results**							
Unbalanced	**Method**	**Metric**	**Neg**	**Pos**	**Neu**	**Accuracy**	**AvgRec**	**AvgPrec**
	SVM	Precision	0.99	0.99	0.98	0.99	0.99	0.99
		Recall	0.98	0.99	0.99			
		F1	0.98	0.99	0.99			
	Naive Bayes	Precision	0.93	0.91	0.91	0.91	0.90	0.91
		Recall	0.83	0.95	0.93			
		F1	0.87	0.93	0.92			
	Maximum Entropy	Precision	0.53	0.88	0.97	0.85	0.90	0.79
		Recall	0.97	0.94	0.78			
		F1	0.67	0.90	0.87			
Balanced	**Method**	**Metric**	**Neg**	**Pos**	**Neu**	**Accuracy**	**AvgRec**	**AvgPrec**
	SVM	Precision	0.99	0.99	0.98	0.99	0.99	0.99
		Recall	0.99	0.99	0.99			
		F1	0.99	0.99	0.99			
	Naive Bayes	Precision	0.93	0.92	0.91	0.92	0.92	0.92
		Recall	0.92	0.95	0.89			
		F1	0.93	0.94	0.90			
	Maximum Entropy	Precision	0.91	0.89	0.88	0.90	0.90	0.90
		Recall	0.89	0.94	0.87			
		F1	0.90	0.91	0.88			

SVM also achieved a higher Accuracy than Cao *et al.* [Cao et al. 2014]. However it is not as expressive as Sem-Eval 2017 difference. The radical distinction between Portuguese and Chinese languages may have an impact in the result. But, for us, the important information in this case is that a system based in ML behaved better than a rule-based one.

The SVM performance using the balanced corpus (99% of Accuracy, Average Recall and Average Precision) is very promising. In our conclusion this result shows that this classifier trained with a balanced corpus, is the best way of classifying traffic-related Tweets generated by traffic authorities.

As examples of posts incorrectly classified by SVM classifier as negative, we can cite *"Liberada faixa central da Av. da Legalidade", "Trânsito liberado no sentido capital/interior da Av. da Legalidade e da Democracia. Fluxo ainda apresenta https://t.co/C07Bzr0OHz", "Trânsito totalmente liberado na Av. da Legalidade e da Democracia"*. All cases are about "Av. da Legalidade e da Democracia" or its small form "Av. da Legalidade". This happens because there are many cases of posts classified as negative containing this street. So, the few cases of non-negative posts about this street are erroneously classified.

5. Conclusion

In this work we addressed the problem of polarity classification of Tweets related to urban traffic. In contrast with the work reported in literature [Cao et al. 2014, Karthik 2017] that use rule-based approaches to perform classification, we used Machine Learning (ML) classifiers. We evaluated the performance of SVM, Naive Bayes and Maximum Entropy models and explore how the use of balanced vs. unbalanced corpus affect their behavior.

The SVM classifier trained with a balanced corpus had the best performance (99% of Accuracy, Average Recall and Average Precision). These outcomes are superior to the results reported by [Cao et al. 2014, Karthik 2017], demonstrating that machine learning methods are suitable for the task.

The main contributions of this work are:

- Use of ML techniques for polarity classification in the traffic domain, instead of the rule-based approach.
- Implementation of TSA in Portuguese language.
- Generation of a traffic domain corpus of Tweets in Portuguese, classified as positive, negative or neutral.

This study is part of a larger project that aims to create a structured database of urban traffic information from social media. All source code, results and corpora used in this work are available at `https://github.com/clarissacastella/eniac2018`.

As future work we will enrich the train and test sets with data from other sources as radios blogs and different Twitter accounts. We will also increase the amount of extracted content by performing named entities extraction and context information identification.

References

Aching., J. L., de Oliveira, T. B. F., and Bazzan, A. L. C. (2014). Traffic information extraction from a blogging platform using a bootstrapped named entity recognition approach. In *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2014 IEEE Symposium on*, pages 6–13, Orlando. IEEE.

Albuquerque, F. C., Casanova, M. A., Lopes, H., Redlich, L. R., de Macedo, J. A. F., Lemos, M., de Carvalho, M. T. M., and Renso, C. (2016). A methodology for traffic-related twitter messages interpretation. *Computers in Industry*, 78:57–69.

Batista, G. E. A. P. A., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Cao, J., Zeng, K., Wang, H., Cheng, J., Qiao, F., Wen, D., and Gao, Y. (2014). Web-based traffic sentiment analysis: Methods and applications. *IEEE transactions on Intelligent Transportation systems*, 15(2):844–853.

D'Andrea, E., Ducange, P., Lazzerini, B., and Marcelloni, F. (2015). Real-time detection of traffic from twitter stream analysis. *IEEE transactions on intelligent transportation systems*, 16(4):2269–2283.

Gilbert, C. and Hutto, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*.

John, G. H. and Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc.

Karthik, A. V. (2017). Implementation of fuzzy based traffic sentiment analysys. *International Journal of Advanced Research in Computer Science*, 8(9):851–854.

Krstajic, D., Buturovic, L. J., Leahy, D. E., and Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of cheminformatics*, 6(1):10.

Lalrempuii, C. and Mittal, N. (2016). Sentiment classification of crisis related tweets using segmentation. In *Proceedings of the International Conference on Informatics and Analytics*, page 89. ACM.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436.

Pawar, P. Y. and Gawande, S. (2012). A comparative study on different types of approaches to text categorization. *International Journal of Machine Learning and Computing*, 2(4):423.

Ratnaparkhi, A. (1997). A simple introduction to maximum entropy models for natural language processing. *IRCS Technical Reports Series*, page 81.

Refaeilzadeh, P., Tang, L., and Liu, H. (2009). Cross-validation. In *Encyclopedia of database systems*, pages 532–538. Springer.

Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.

Schaffer, C. (1993). Selecting a classification method by cross-validation. *Machine Learning*, 13(1):135–143.