Sentiment Analysis of Tweets in Brazilian Portuguese with Convolutional Neural Networks

Juan Manuel Adán Coello, Armando Dalla Costa Neto

Brazil

Abstract

Sentiment analysis of texts posted on Twitter is a natural language processing task whose importance has grown along with the increase in the number of users of the platform and the interest of organizations on the opinions of their employees, customers and users.Although Brazil is the sixth country in the world with most active users of Tweeter and Portuguese is the seventh most spoken language in the world, with 221 million speakers (200 million of them living in Brazil), the number of articles that discuss sentiment analysis approaches for Brazilian Portuguese is a small fraction of those that focus on the English language. On the other hand, few works use deep learning for this task when compared with other machine learning and lexical based methods. In this context, the work described in this article addresses the problem using Convolutional Neural Networks (CNN). The paper presents the results of an experimental evaluation that shows that a CNN with a relatively simple architecture can perform much better than a previous approach that uses ensembles of other machine learning classifiers combined with text preprocessing heuristics.

Keywords: opinion mining; deep learning; Brazilian Portuguese.

1. Introduction

Despite having to face new competitors, Twitter is still one of the most important social media platforms in the world, being in February 2019 the third in market share in the world and the fourth in Brazil. Tens of millions of Brazilians use Twitter to consume and comment on information, being a valuable source for governments and companies to know the habits and opinions of their citizens, users and customers.

Sentiment analysis is one of most demanded tasks when it comes to analyze the manifestations of people in social networks. The task usually corresponds to determine if a message expresses a positive or negative feeling about some object of interest.

Most of the papers that deal with sentiment analysis focus on the English language with a relatively small presence of works directed to the Portuguese language, and in particular to Brazilian Portuguese. An evidence of this is that a search conducted in Google Scholar on 3/29/2019 with the keywords *"sentiment analysis" Twitter* returns 40,800 results, while the search for *"sentiment analysis" "Brazilian Portuguese" Twitter* returns only 298 results. The disproportion is very great if we consider that English is the third language of the world when considering first-language speakers, with approximately 379 million speakers,

but Portuguese is the 7th with 221 million speakers, around 200 million living in Brazil, and the second most spoken romance language in the word after Spanish (2nd in the world with 460 million speakers)[1].

1.1 Convolutional Neural Networks in Text Processing

For more than a decade, Natural Language Processing (NLP) had as its main protagonists machine learning approaches that use linear models such as Support Vector Machines (SVM) and Logistic regression (LR), trained from sparse vectors of characteristics. In recent years, there is a process of replacing these linear models with neural network models trained with dense vectors.

Indeed, neural networks have reemerged as powerful machine learning models, generating state-of-the-art results in fields such as image recognition and speech processing. Recently, neural network models have also been applied to natural language text processing tasks, with very promising results. The nonlinearity of the network, as well as the possibility of easily using vector representations of pretrained words, usually lead to a highly accurate classification.

Networks with convolution and pooling layers have been very useful in tasks where it is expected to find strong local clues about belonging to a class, but these clues can appear in different places of the input. It is desired to learn what sequences of words are good indicators of the considered topic, without necessarily being relevant where they appear in the text. Convolution and pooling layers allow the model to learn to find these local indicators regardless of their position.

A convolutional layer is, in essence, a feature extraction architecture. It does not constitute a useful network alone, it must be integrated into a larger network, and be trained to work together with it to produce the end result. The responsibility of the convolution layer is to extract significant substructures that are useful for the considered prediction task.

Kim [1] observed that despite the need to adjust a large number of hyperparameters, a CNN with a single convolution layer can perform remarkably well, and confirms evidence already established in the literature that vector pretraining is an important element in deep learning for NLP applications.

In the standard approach to using CNN for NLP, sentences are mapped to word vectors that are presented to the model as an input matrix. Convolutions are then applied to input words using distinct sizes of kernels, for example covering 2 or 3 words at a time. The resulting feature map is then processed by a max pooling layer to condense or summarize the extracted features. In short, the architecture of a CNN is therefore composed of three key elements:

- Word Embedding: a distributed representation in which words with similar meanings (based on their use) also have a similar representation.

[1] https://www.ethnologue.com/statistics/size

- A Convolution Layer: a feature extraction model that learns to extract salient features of represented documents using word vectoring.
- A Fully Connected Model that interprets the characteristics extracted in terms of a predictive output.

A recent study involving problems of binary classification of texts has identified some useful elements that can be used as a starting point for the configuration of a CNN for text classification. The general findings are as follows [2]:

- Vectorization models lead to higher performance than using one-hot encoding (vector representation where all vector elements are 0, except one, which has 1 as the value)
- Core size is important and should be adjusted for each problem.
- The number of feature maps is also important and should be adjusted.
- 1-max pooling generally performs better than other types of pooling.
- Dropout has little effect on model performance.

The study also provides some specific heuristics, among them:

- Make a grid search considering several sizes of cores in the range 1-10 in order to find the optimal configuration for the problem.
- Search for the best number of filters, between 100 and 600, and explore dropout rates between 0.0 and 0.5 as part of the same search.
- Explore the use of linear activation functions such as relu and tanh.

Kim explored a CNN architecture for a variety of sentence classification tasks. Many researchers quickly adapted its simple yet effective network. After trained, the convolutional kernels became n-gram feature detectors for the target task.

Overall, CNNs are extremely effective in mining semantic clues in contextual windows. However, they are very data heavy models. They include a large number of trainable parameters which require huge training data. This poses a problem when scarcity of data arises. Another persistent issue with CNNs is their inability to model long-distance contextual information and preserving sequential order in their representations

2. Datasets for Brazilian Portuguese Sentiment Analysis

One of the major difficulties for comparing approachs for sentiment analysis of tweets in Brazilian Portuguese is the lack of standard annotated datasets. For the accomplishment of the present work we have considered some of the few recent corpus of tweets in Brazilian Portuguese cited in the literature, including the PELESent [3], Tweet-SentBR [4] and BRTweetSentCorpus [5].

PELESent is a large collection of labeled tweets built using distant supervision. Following the approach of [6], 41 million of Tweets were collected and their polarity determined using a lists of emojis and emoticons according to the sentiment they conveyed. If a tweet contains both positive and negative elements, it is discarded since it is likely to be ambiguous. The final corpus comprises 554,623 positive and 425,444 negative tweets.

TweetSentBR has 15,000 manually annotated tweets on the domain of TV shows, labeled as positive (44%), neutral (26%) and negative (29%), and includes tweets related to nine programs from three major TV channels in Brazil, selected based on their popularity and presesnce in social media.

BRTweetSentCorpus is a dataset with 12076 labeled tweets in Brazilian Portuguese, collected over a six months period, from a range of different topics, including brands, social networks, telecommunication companies, companies with active marketing campaigns, sports, regions, video-games, movies, books, food, government and events. Tweets were manually classified as positive, negative, ambiguous and non-opinionated. Tweets are ambiguous when they have more than one polarity, such as "I love tulips but I hate roses" and are non-opinionated when they express facts, such as in "The president arrives today". Out of the 12076 tweets, 5034 were classified as non-opinionated, 582 as ambiguous, 3280 as negative and 3180 as positive.

PELESent's approach to label data is fast, but is unable to deal with several language phenomena as irony and sarcasm, that can lead to incorrect labeling. Indeed some research, as [7], has found that emoticons are able to reverse the polarity of the true sentiment values of sentences. For our experiments we choose to use BRTweetSentCorpus as it covers a broader set of topics than TweetSentBR.

3 Convolutional Neural Net Architecture

To evaluate deep neural networks in the task of sentiment analysis of Brazilian Portuguese tweets, we experimented with CNNs using the Keras API running on top of TensorFlow.

As the networks' inputs have to be numerical, before feeding each tweet into the net it is tokenized and mapped to a vector of integers corresponding to the index of the represented word on a previously built dictionary. This vector is then used as the input to the first level of the net, an embedding layer that maps each integer representing a word into a vector. The objective is to represent semantically close words by nearby vectors in the multidimensional space created as part of the embedding process.

The embedding layer is followed by a dropout layer that randomly sets a fraction of the input units to zero at each actualization during the training phase, aiming at reducing the complexity of the model and prevent over fitting.

The output of the dropout layer is fed into a 1D convolution layer (CONV1D) that inserts a convolution kernel to perform a unidimensional convolution with its input (to deal with adjacent words) and produce a tensor of outputs. The main parameters of this layer are the number of filters and the kernel size. The number of filters defines the dimensionality of the output space, i.e., the number of output filters in the convolution. The kernel size corresponds to the size of the 1D convolution window.

The output of the 1D convolution layer feeds a pooling layer (GlobalMaxPooling1D) that reduces the dimensionality of the output from the convolution layer. It also helps to control overfitting without losing the features extracted by the precedent layers. Jacovi et al. [8] showed that maxpooling essentially separates "between features that are relevant to the final classification and features that are not".

A dense layer comes next (Dense). This is just a regular layer of neurons that represent a matrix vector multiplication. The values in the matrix are the trainable parameters, which are updated during backpropagation.

A new Dropout layer in inserted after the dense layer and its output feeds an activation layer (Activation) with a relu function, whose output is then projected onto a single unit output dense layer, and squashed to the interval 0..1 by a sigmoid activation function.

4 Experimental Evaluation

This section will describe a set of tests performed to improve network performance. The tests seek to adjust the following parameters: number of training epochs (default= 10), size of the tokenization dictionary (default= 5000 words), input length (number of words fed into the net to represent each tweet; Default= 140) number of convolution layer filters (default = 250) and size of convolution filters (ie, the amount of elements on which filters act; default= 3).

In each test, only the parameter being adjusted has its value changed, the others keep the initial default values. At the end of the test, the modified parameter returns to its initial value for the adjustment of the other parameters. It is observed the influence of the considered parameter on classification accuracy (the fraction of classifications that are correct), network loss and training time. Since we are working with a perfectly balanced dataset, the metric accuracy (percentage of total items classified correctly) seems to be the most adequate to evaluate the models. The experiments used a subset of BRTweetSentCorpus with 3180 positive and 3180 negative tweets, as done by [5], in order to have a baseline to compare our approach.[2]

It was used the binary cross-entropy loss function and the adam optimization algorithm to update network weights.

The training time has also been recorded, since in some cases the accuracy may have a slight increase over the course of the test, but the time it took to achieve a slight increase may not be worth it. The training times presented refer to experiments performed on an Intel i7 2.00Ghz CPU.

[2] Loss is the quantitative measure of deviation or difference between the predicted output and the actual output in anticipation. A loss value shows how well or poorly a certain model behaves after each iteration of optimization. Ideally, it is expected the reduction of loss after each iteration.

In the following graphs, the solid lines refer to the accuracy and loss values during training and the dotted lines to the values obtained during network validation.

4.1 Number of Training Epochs

This test consisted of training and validating the network using 5-fold cross validation, with the default values of the parameters previously presented. The graphs presented in figures 1 and 2 show that after the second epoch the network starts to memorize the training data (overfitting), reducing its accuracy and increasing the loss for new data. The execution time, not shown here, presents little variation as expected since there was no change of the network parameters during the test.

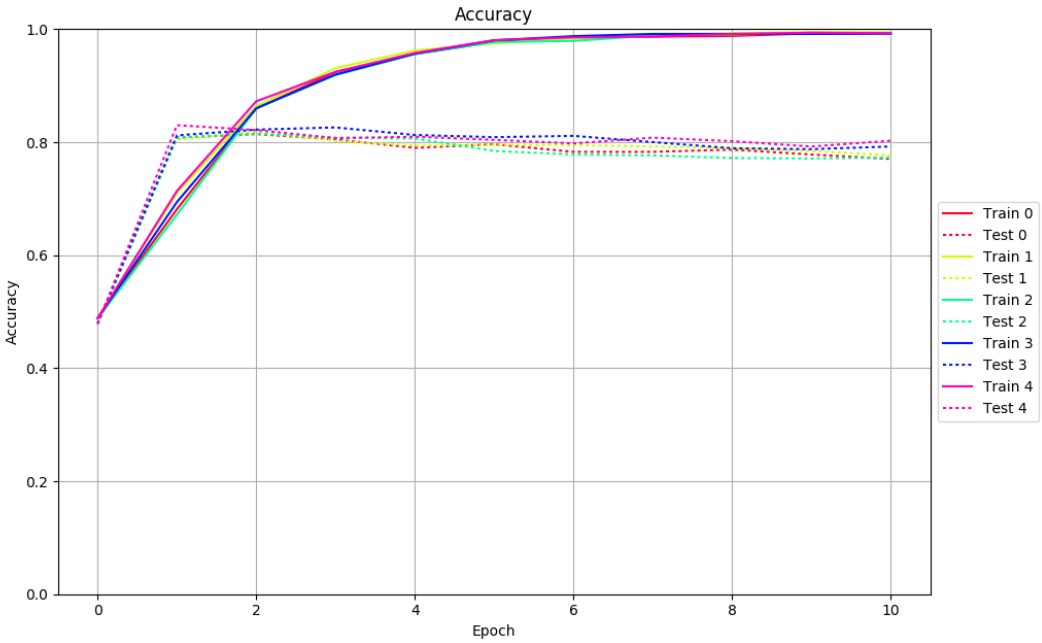

Figure 1: Accuracy vs. Epoch.

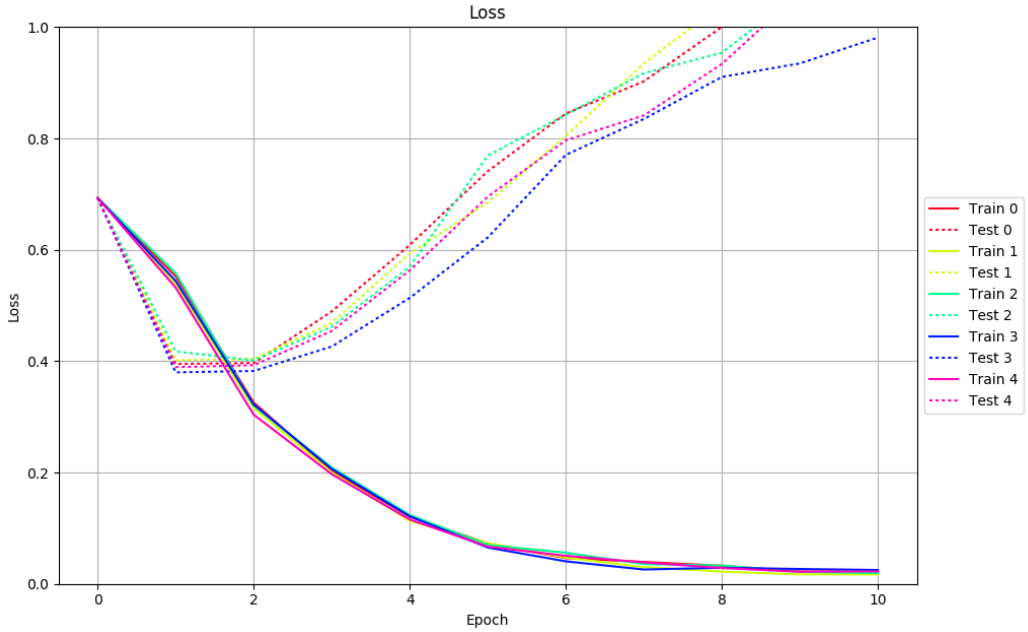

Figure 2: Loss vs. Epoch

4.2 Size of the Tokenization Dictionary

In this test, we analyzed the impact of tokenization dictionary size on network performance. Figure 3 shows that with values above 5000 words (Test 4) there is a marginal increase in accuracy, however, as shown in Figure 4, from this point on loss increases.

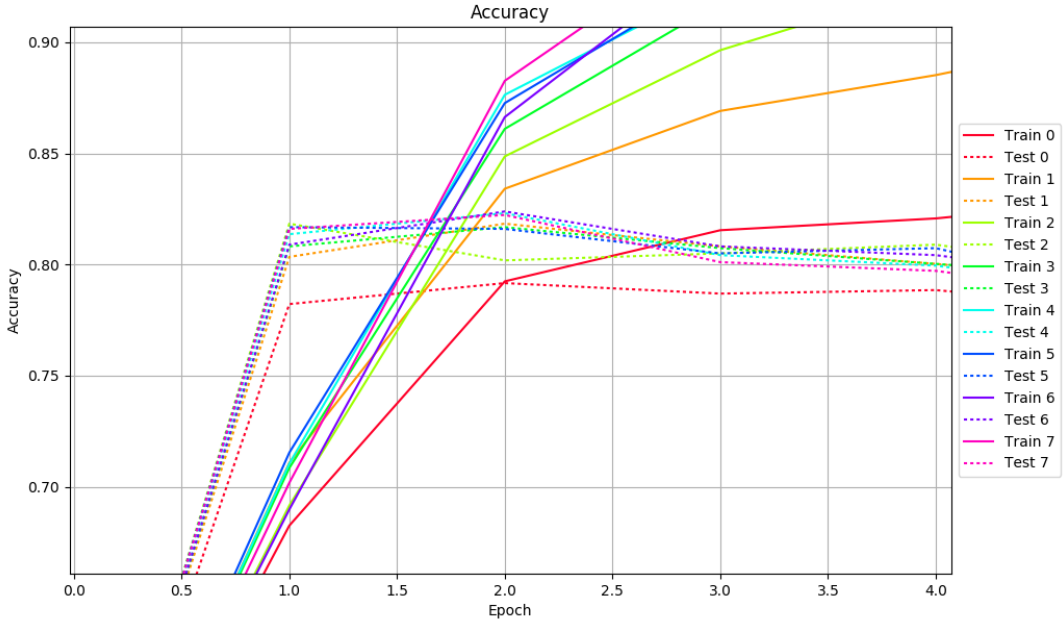

Figure 3: Accuracy vs. Tokenization Dictionary Size.

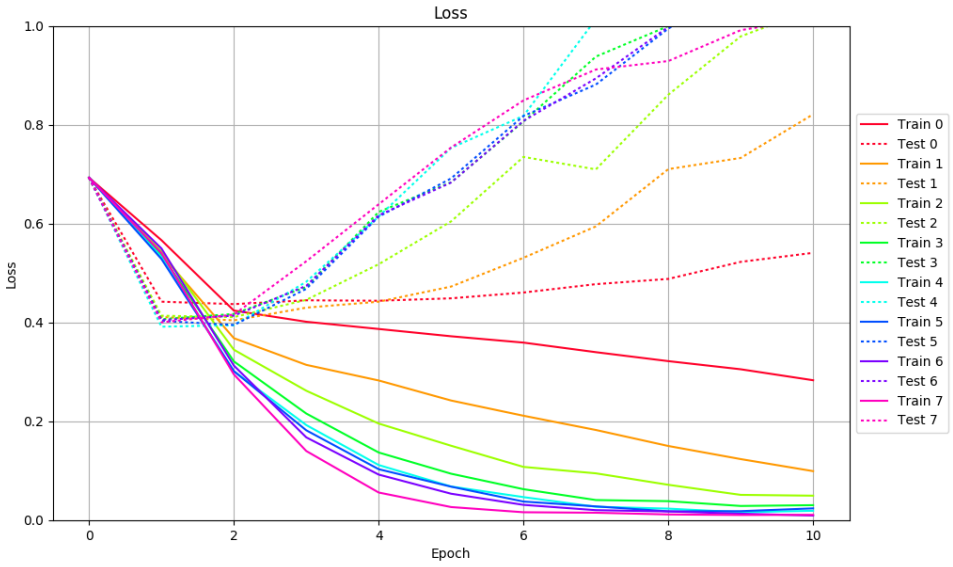

Figure 4: Loss vs. Tokenization Dictionary Size.

4.3 Input Length

Taking into account that Twitter texts are short, the size of the character string that represent tweets starts at 5 and is incremented by 5 each running. In can be observed in Figures 5 and 6 that up to "Test 4"

(sequence size equal to 25) the accuracy increases and the loss decreases. After this point, there is a small increase in accuracy, but also in loss.

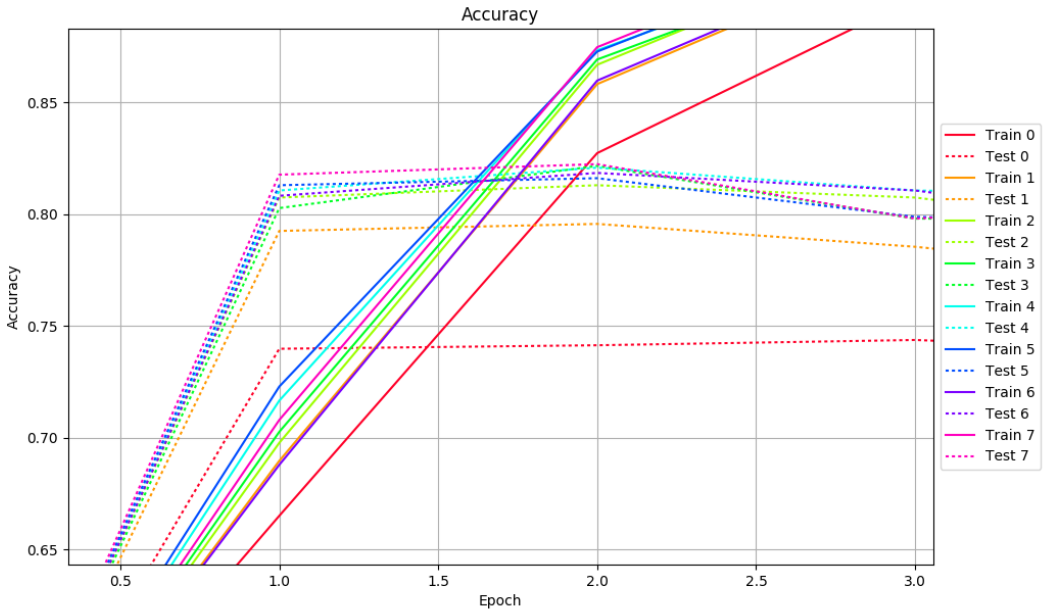

Figure 5: Accuracy vs. Input Length.

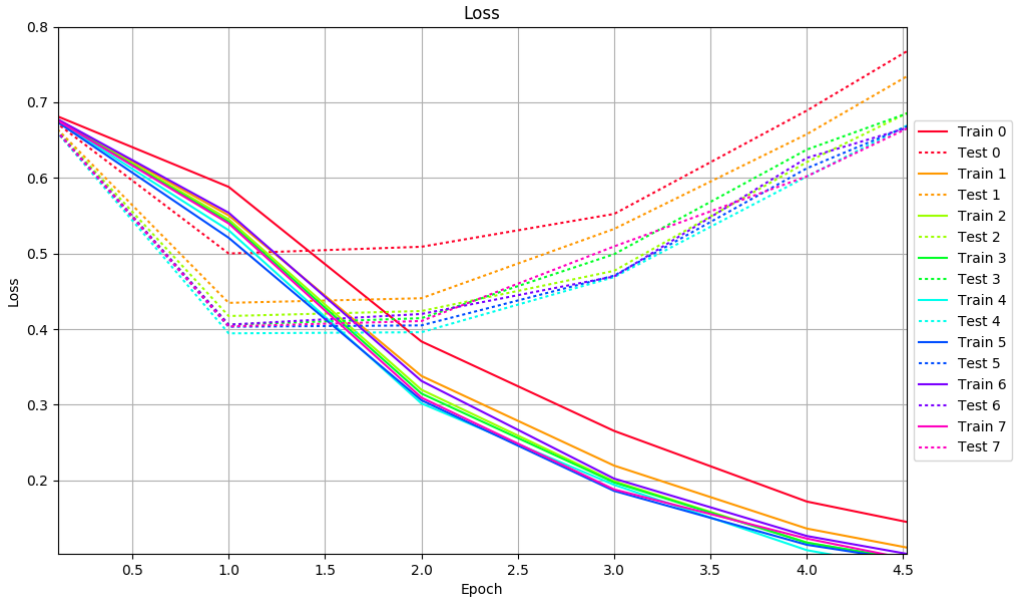

Figure 6: Loss vs. Input Length.

4.4 Number of Convolution Filters

The number of filters was modified at each run, starting at 50 and increased by 50. The results obtained are shown in Figures 7 and 8 and the times required for each training step are recorded in Table 1. The figures and table suggest that it is not interesting a large increase in the number of filters, due to its the impact on training time with no equivalent increase in network performance; 250 seems to be a good value.

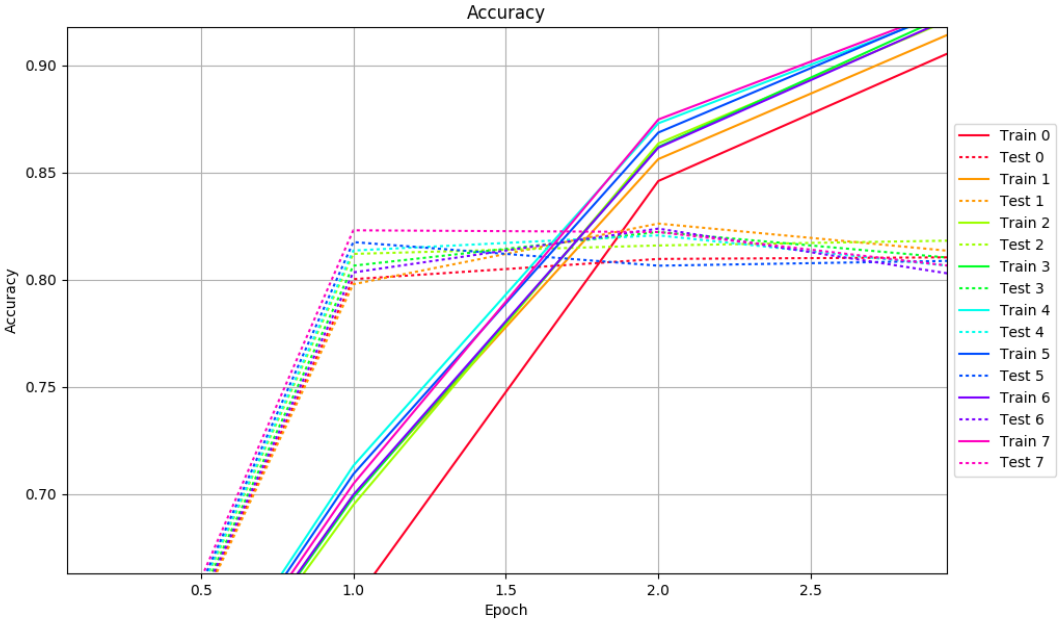

Figure 7: Accuracy vs. Number of Filters.

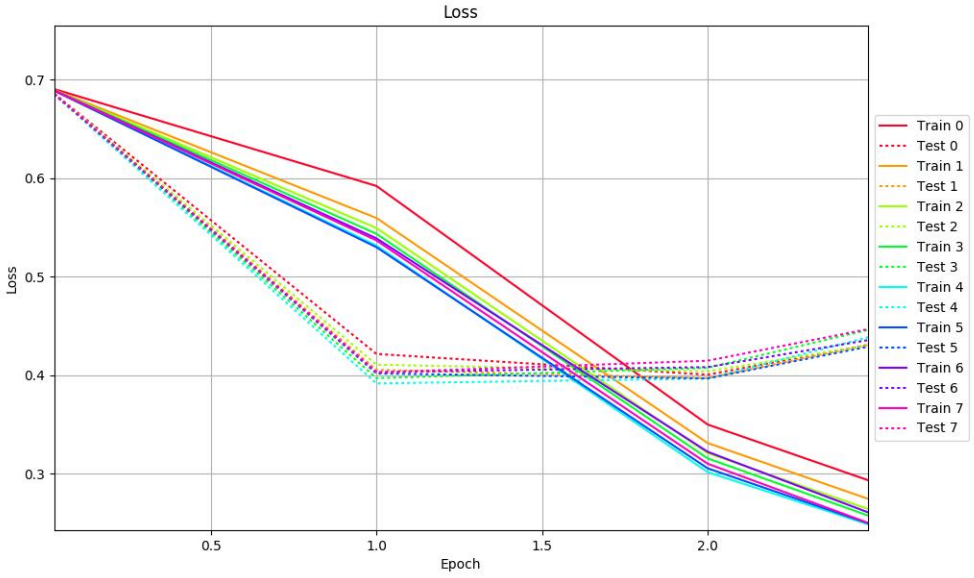

Figure 8: Loss vs. Number of Filters.

Table 1: Training Time vs. Number of Filters

Execution	Filters	Time (s)
0	50	67.16
1	100	107.34
2	150	148.37
3	200	188.14
4	250	232.01
5	300	272.36

6	350	318.8
7	400	439.67

4.5 Size of Convolution Filters

The size of the filters ranged from 1 to 8, in increments of 1. As can be observed in the figures 9 and 10 the best results are verified for filters of sizes 3 and 4.

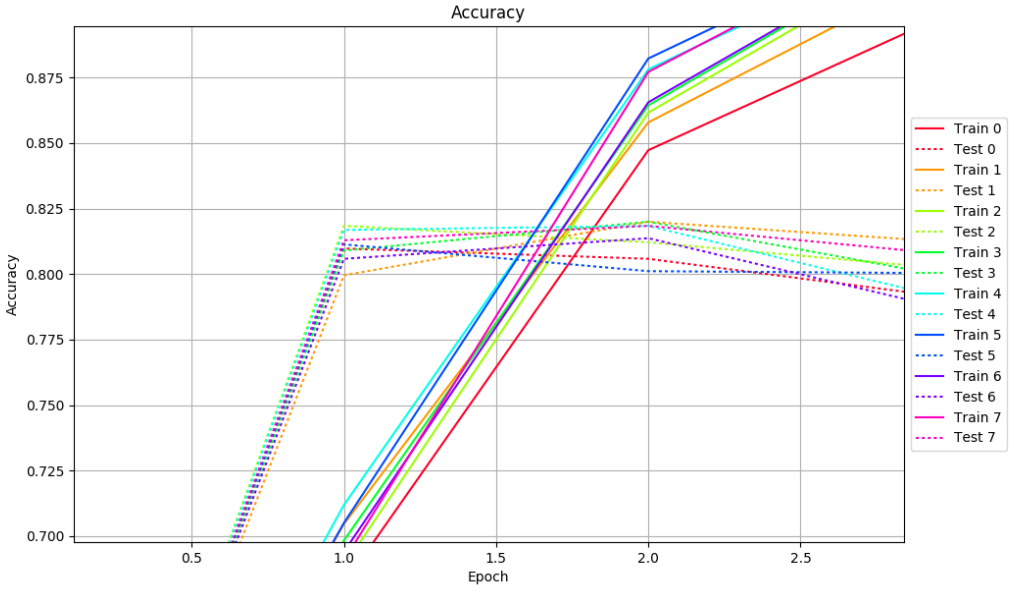

Figure 9: Accuracy vs. Filter Size.

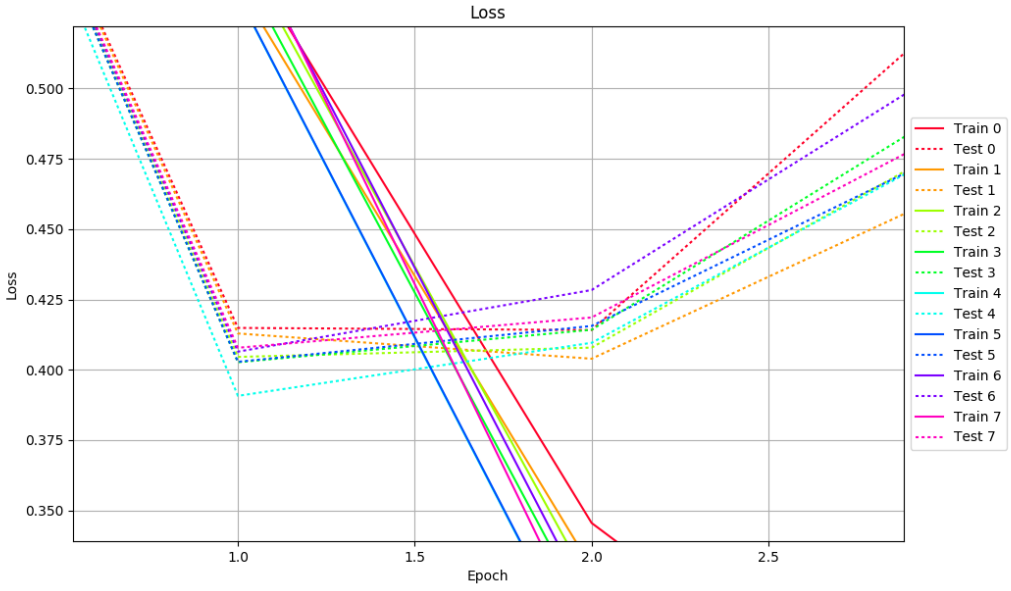

Figure 10: Loss vs. Filter Size.

After performing the tests previously described, the network parameters were set according to Table 2 and three new tests were performed, all with 5-fold cross-validation. The average accuracy on the tests was 82.05% and the mean training time 14.84 seconds. This result can be favorably compared to the 76.14% accuracy obtained in previous work, for the same dataset, using 8 text preprocessing heuristics and an ensemble with three classifiers (Naive Bayes, Logistic Regression and Recursive Neural Tensor Network).

That is, our CNN achieved a precision 7.8% higher, without text preprocessing and using only one classifier.

Table 2: Network Parameters Adjusted.

Parameter	Value
Max_features	5000
Input length	25
Convolution filters	100
Filter size	3
Epochs	2

5 Related work

The task of sentiment analysis in tweets has been addressed since 2013 in the series of evaluations conducted during the International Workshop on Semantic Evaluation (SemEval). The competition focused on the English language until 2017 when the Arabic Language was included [9]. Participated in SemEval2017, 48 teams; 20 of them used deep learning methods and neural networks. SVM was also widely used, mainly in combination with other methods, including neural networks. The team that performed best on the five subtasks of Task 4 (Sentiment Analysis in Twitter) developed a system that combines a Convolutional Neural Network with a Recurrent Network (LSTM) [10]. In Subtask B, which consisted of classifying tweets in POSITIVE or NEGATIVE, depending on the expressed sentiment with respect to a particular topic, this team reached an accuracy of 89.71%.

Instead of building datasets and classifiers for Brazilian Portuguese (BP), some authors translate the text in BP to English and use resources available for that language. Although it is a quick and easy to implement approach, its results are at some distance from the state of the art. That is the case in [11], were tweets and text from other social media are translated to English in order to use available tools. The authors point out that the poor results obtained for the translated text highlight the need for using resources in the target language and social media context.

A study comparing three learning algorithms (Naıve Bayes, SVM and MaxEnt) and three feature selection methods (Chi-Square, CPD and CPPD) for classifying texts related to 2014 elections in Brazil is presented in [12] . The authors analyzed and visualized Twitter messages, ranking the posts relatively to variations in moods within the Brazilian territory.

In [13] it is presented a systematic mapping of approaches for opinion mining of Portuguese online text. The study found that almost 70% of all revised works focus on the Brazilian Portuguese variant. Naïve Bayes and Support Vector Machine were the most used classifiers and SentiLexPT the most used lexical resource.

A language-agnostic translation-free method for Twitter sentiment analysis, which makes use of deep convolutional neural networks with character-level embedding is proposed by [14]. The proposed approach was evaluated in a tweet corpora in four different languages, showing that it outperforms their baselines (including SVM), reaching an accuracy of 0.706 for Portuguese.

A recent work applies a method based on the co-occurrence of terms with a sentiment descriptor vocabulary on a set of Brazilian Portuguese Twitter messages related to health topics(cancer), achieving precision and recall values of 0.68 and 0.67, respectively [15].

6. Final Remarks

In the work described in this paper a Convolutional Neural Network (CNN) was built to identify the sentiment expressed in tweets written in Brazilian Portuguese. After a parameter adjustment phase, the network was evaluated using a dataset composed of 3180 positive and 3180 negative tweets, related to various subjects. An accuracy of 82.05% was obtained, 7.8% higher than that achieved on previous work for the same data set.

The results are good, compared to previous work, but there is still room for performance enhancements, that should be sought in future work by searching for new network architectures and the experimental study of new sets of parameters for the models.

7. References

[1] Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882. (2014).

[2] Zhang, Y., Wallace, B.: A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 253–263 (2017).

[3] Corrêa, E.A., Marinho, V.Q., dos Santos, L.B., Bertaglia, T.F.C., Treviso, M.V., Brum, H.B.: PELESent: Cross-domain polarity classification using distant supervision. In: 2017 Brazilian Conference on Intelligent Systems (BRACIS). pp. 49–54. IEEE (2017).

[4] Brum, H.B., Nunes, M. das G.V.: Building a Sentiment Corpus of Tweets in Brazilian Portuguese. In: Proceedings of the Eleventh International Conference on Language Re-sources and Evaluation (LREC 2018), Miyazaki, Japan (2018).

[5] Gomes, F.B., Adán-Coello, J.M., Kintschner, F.E.: Studying the Effects of Text Preprocessing and Ensemble Methods on Sentiment Analysis of Brazilian Portuguese Tweets. In: International Conference on Statistical Language and Speech Processing. pp. 167–177. Springer (2018).

[6] Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford. 1, (2009).

[7] Teh, P.L., Rayson, P., Pak, I., Piao, S., Yeng, S.M.: Reversing the polarity with emoti-cons. In: International Conference on Applications of Natural Language to Information Systems. pp. 453–458. Springer (2016).

[8] Jacovi, A., Shalom, O.S., Goldberg, Y.: Understanding Convolutional Neural Networks for Text Classification. In: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. pp. 56–65 (2018).

[9] S. Rosenthal, N. Farra, e P. Nakov, "SemEval-2017 task 4: Sentiment analysis in Twitter", in Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017), 2017, p. 502–518.

[10] M. Cliche, "BB_twtr at SemEval-2017 task 4: twitter sentiment analysis with CNNs and LSTMs", arXiv preprint arXiv:1704.06125, 2017.

[11] Cirqueira, D., Jacob, A., Lobato, F., de Santana, A.L., Pinheiro, M.: Performance evaluation of sentiment analysis methods for Brazilian Portuguese. In: International Conference on Business Information Systems. pp. 245–251. Springer (2016).

[12] Prata, D.N., Soares, K.P., Silva, M.A., Trevisan, D.Q., Letouze, P.: Social Data Analysis of Brazilian's Mood from Twitter. International Journal of Social Science and Humanity. 6, 179 (2016).

[13] Souza, E., Vitório, D., Castro, D., Oliveira, A.L., Gusmão, C.: Characterizing Opinion Mining: A Systematic Mapping Study of the Portuguese Language. In: International Conference on Computational Processing of the Portuguese Language. pp. 122–127. Springer (2016).

[14] Wehrmann, J., Becker, W., Cagnini, H.E., Barros, R.C.: A character-based convolutional neural network for language-agnostic Twitter sentiment analysis. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 2384–2391. IEEE (2017).

[15] Araujo, G.D. de, Teixeira, F.O., Mancini, F., Guimarães, M. de P., Pisa, I.T.: Sentiment Analysis of Twitter's Health Messages in Brazilian Portuguese. Journal of Health Informatics. 10, (2018).