SEPLN TASS-2017 Task 1 and Task 2: Aspect-based sentiment analysis with Convolutional Neural Networks for spanish in short texts

Joao V. Pinheiro R. & Leticia L. Pinto A.

Catholic University San Pablo

Abstract. This paper describes a model of deep learning to aspect based sentiment analysis as part of SEPLN TASS-2017 Task 1 and Task 2. In this project we use a convolutional neural network(CNN) for both tasks where we use three channels for each input. For Task 1 we use pre-trained models to represent our input like three channels of embeddings, which later will be proccess by CNN. For Task 2, based on the strategy for TASK 1 we additionaly to determine the sentment towards an aspect, we concatenate an aspect vector with every word embedding and apply a convolution over it.

1 Introduction

With the advance of technology and the easiest access to the Internet in the world, an increasing number of people express their opinions online. One common social network is Twitter in which are based on our experiments, their twetts. Sentimental analysis allows us to derive these tweets in superficial opinions related to their overall polarity. Aspect-based sentiment analysis allows us to go deeper and determine sentiment towards such aspects of an entity.
In this paper, we introduce the CNN used in the SEPLN TASS-2017 for TASK 1 and TASK 2 using their dataset for training and test.

2 Related works

Aspect-based sentiment analysis is traditionally split into an aspect extraction and a sentiment analysis subtask. In this case we focus in the sentiment analysis. Past deep learning-based approaches have focused mostly on the sentiment analysis subtask: Tang et al. (2015) use a target-dependent LSTM to determine sentiment towards a target word, while Nguyen and Shirai (2015) use a recursive neural network that leverages both constituency as well as dependency trees. Previous approaches in both cases like [1], which introduce their own deep learning, using a model architecture as an extension of the CNN structure used by Collobert et al. (2011). In our case we use the best part of each one to develop our model.

3 Main concepts

When we think in Convolutional Neural Network(CNN) we usually think in computer vision, because the biggest development is that line from Facebook's automated photo tagging to self-driving cars. Most recently it's started to apply in Natural Language Processing (NLP). The main idea is understand what is a convolution. One easiest way to understand a convolution is by thinking of it as a sliding window function applied to a matrix. CNNs are basically just several layers of convolutions with nonlinear activation functions like ReLU or tanh applied to the results. In a traditional feedforward neural network we connect each input neuron to each output neuron in the next layer. That's also called a fully connected layer, or affine layer.

4 Model

We use an architecture of convolutional neural network proposed by Collobert et al. (2011) which new features achieving good results for others authors like (Kim, 2014) like being described in the Fig 1 and (Ruder , 2016) , the model take like input a text which is padding with zeros to obtain a length of n words o cutting if excess n. To obtain n value it takes the most frequently words inside the dataset of tweets. We represent the tweet like a concatenation of word embbedings x donde $x_i \epsilon R^k$ and it is a vector of the i-th word in the tweet, also it was used three channels of embeddings (Word2Vec, FastText y GloVE) like input for the CNN following the same strategy of concatenation for each channel. For each iteration we apply one convolutional layer which slides by one filter of defined size of two over the input of embedding; We decided to use that length of filter because of the short media length of words by tweet. As activation, we opted for a nonlinear function ReLU. Then the model take the feature more relevant taking the maximum value of the layer output of convolution, this phase is performed simultaneously for each input channel and later one dense layer with activation function ReLU take the concatenation of maximum values of the map of features produced by filters in each channel. Finally one softmax layer classifies like a distribution of probability over all classes of the output.

5 Methodology

5.1 Preprocessing

Initially we convert all the tweets to lowercase and we get rid of symbols and special characters. Subsequently, a vocabulary was defined, which was defined by extracting the words from pre-trained vectors and searching coincidences in the whole corpus of the dataset, we only keep the words of the vocabulary.

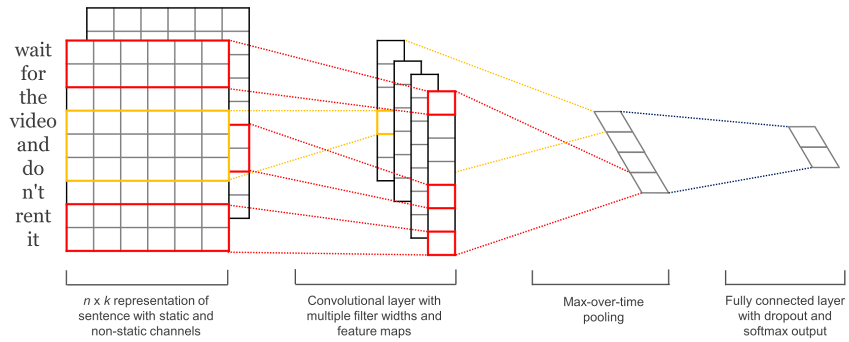


Fig. 1: Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification

5.2 Hyperparameters

For both tasks we use the next Hyperparameters, mini-bach zise of 32, maximum sentence lengh of 'n' tokens, word embedding size of 300, and 100 as filter map. We use filter length of 2. The model was trained with 5 seasons using minibach SGD and optimizer Adam to stop the calculation if necessary.

5.3 Sentiment Polarity

For the sentiment analysis based on aspects we take the tweet as input, using the vectorial representation, along with the vector of the word associated with the aspect. In the case that we count with more than one word for aspect (ENTITY#CATEGORY) The following approach was followed to obtain the aspect vector: We make an split and obtain the two words **ENTITY** and **CATEGORY**, which we averaged in the vector space of embeddings. For the cases which the aspect do not exists inside the vocabulary we choose a random vector. Finally this vector is associated with the aspect and then concatenated with each word of the input, generating a matrix twice the size of the matrix obtained only for the sentence. The rest of the procedure is processed in the same way as in Task 1.

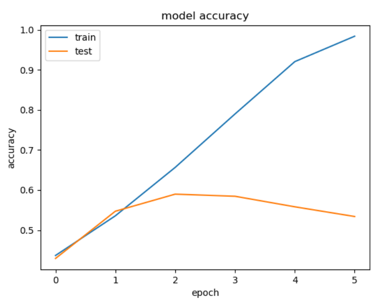
6 Experimental results

After the implementation of the model we evaluated with the same dataset provided by SEPLN TASS-2017 Task 1 and Task 2, for the first TASK we used 1899 samples achieving a maximum accuracy of 59%, while for the second task, 1948 samples were used, achieving a maximum accuracy of 57 %. As can be appreciated in the table 1.

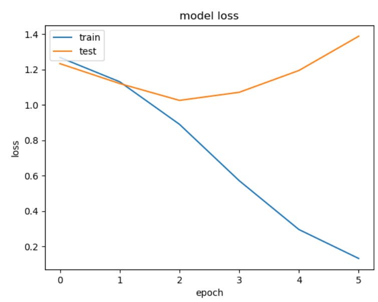
Table 1: Table of experiment results by time

Time	Accuracy Task 1	Accuracy Task 2
Time 1	0.594	0.5513
Time 2	0.5919	0.5693
Time 3	0.5814	0.5601
Time 4	0.5893	0.5712
Time 5	0.5678	0.5509
Time 6	0.5712	0.5623
Time 7	0.5876	0.5412
Time 8	0.5931	0.5614
Time 9	0.572	0.5761
Time 10	0.5861	0.5621

6.1 Task 1



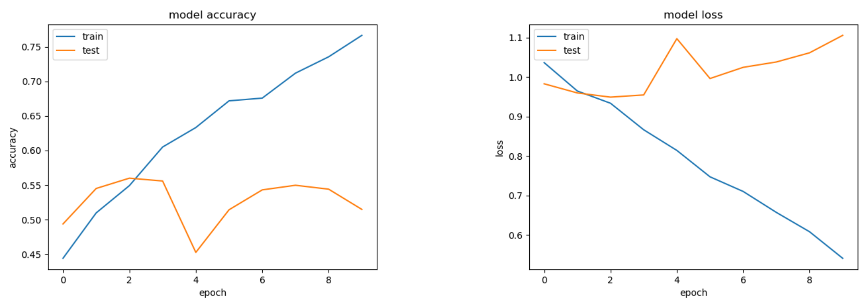
(a) Model Accuracy



(b) Model Loss

Fig. 2: Models for Task 1

6.2 Task 2



(a) Model Accuracy (b) Model Loss

Fig. 3: Models for Task 2

7 Conclusion

In this paper, we present an approach based on deep learning for aspect-based sentimental analysis, that employs a convolutional neural network to make the task 1 and 2 as part of SEPLN TASS-2017. We had demonstrated convinced results for both tasks because both of them approximate to state of art for both of the datasets. We evaluated our model and as a result of the analysis in this case we can conclude that a CNN is adequate to resolve this kind of problems too. For short texts is convenient do not use a many convolutional layers either extensive filters. Regarding the analysis of feelings based on aspects, it has been shown that the approach of covering the problem as a task of classifying multiple classes is effective and inexpensive to process.

References

1. **CNN**
   Kim, Y. (2014) - Convolutional Neural Networks for Sentence Classification
2. **SemEval-2016**
   INSIGHT-1 at SemEval-2016 Task 5: Deep Learning for Multilingual Aspect-based Sentiment Analysis
3. **Spanish Billion Word Corpus and Embeddings**
   Cardellino, C. - 2016 `http://crscardellino.me/SBWCE/`
4. **Target-Dependent Sentiment Classification with Long Short Term Memory**
   Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015.

5. **PhraseRNN :Phrase Recursive Neural Network for Aspect-based Sentiment Analysis**
   Thien Hai Nguyen and Kiyoaki Shirai. 2015.