SENTIMENT ANALYSIS USING ADVANCED TECHNIQUES FOR SPANISH IN SHORT TEXTS

A PREPRINT

José Chávez
Universidad Católica San Pablo
Arequipa, Perú
jose.chavez.alvarez@ucsp.edu.pe

Daniel Palomino
Universidad Católica San Pablo
Arequipa, Perú
daniel.palomino.paucar@ucsp.edu.pe

March 13, 2019

ABSTRACT

This paper presents three approaches for classifying the sentiment of tweets for the Peruvian Spanish variant in the Sepln TASS 2017 and 2018 challenge, including traditional techniques like word embedding with CNN as well as advances techniques such as Smooth Inverse Frequency (SIF) and Universal Language Model Fine-tuning (UMLF).

***K*eywords** SIF · UMLF · Sepln · TASS 2017 · TASS 2018

1 Introduction

Sentiment analysis is one of the most important tasks related to subjectivity analysis within Natural Language Processing. The sentiment analysis of tweets is especially interesting due to the large volume of information generated every day, the subjective nature of most messages, and the easy access to this material for analysis and processing. The existence of specific tasks related to this field, for several years now, shows the interest of the NLP community in working on this subject.

In this paper, we compare different advances techniques for Sentiment Analysis such as Word Embedding with CNN, Smooth Inverse Frequency (SIF) and Universal Language Model Fine-tuning (UMLF) in Spanish Tweets using data samples of SEPLN TASS 2017 and 2018.

2 Corpus and Dataset

In order to use a well known and complete corpus in spanish for build word embedding, we use the Spanish Billion Word Corpus with a size of 1.4 billion words. The algorithm used is Word2Vec with Skipgram by GenSim (For details on parameters please refer to the SBWCE page) which were computed by Cristian Cardellino with the parameters:

Dimensions	Vectors
300	1000653

Table 1: Word Embedding Parameters

The datasets used for training, development and testing are downloaded from the official page TASS 2017 and TASS 2018.

All the corpora are annotated with 4 different levels of opinion intensity (P, N, NEU, NONE). In order to get a good performance on training procedure we equate the amount of data of each label.

3 Models

We implement three different different architectures:

3.1 Convolutional Neural Network with Word Embedding

Our method uses a CNN based on [3] that concatenate *bi-grams*, *tri-grams* and *four-grams* in order to have a good representation of a sentence. All sequences of words on each tweet having a fixed size padding with zeros. Finally we have a two dense layers with a *softmax* activation of four outputs(P, N, NEU, NONE).

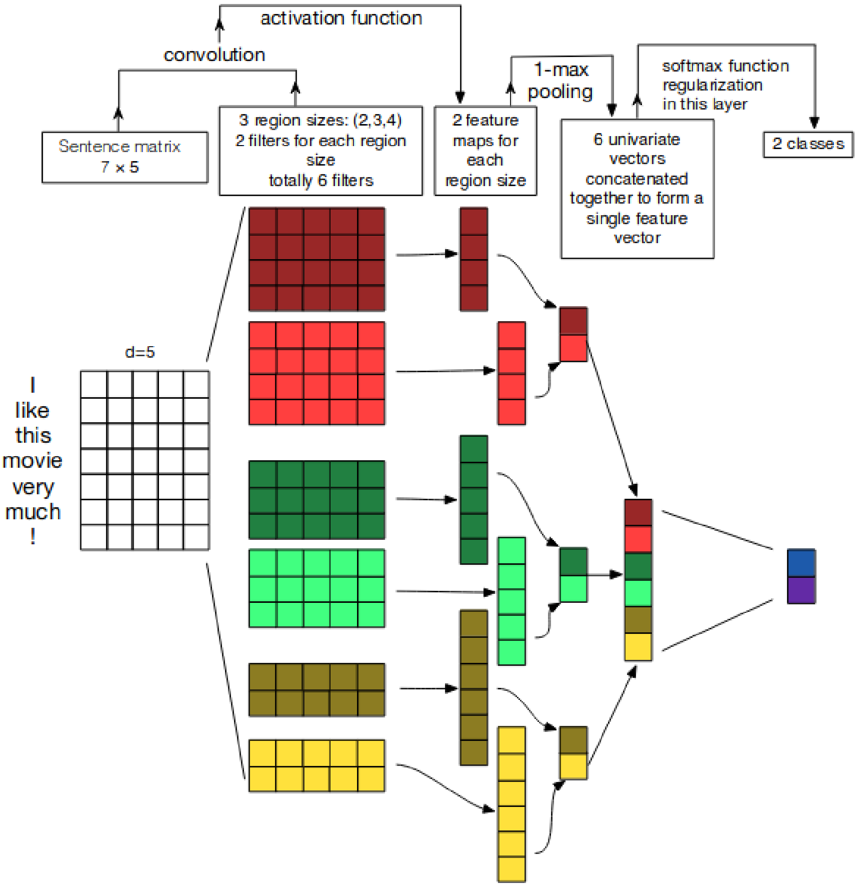

activation function
convolution
1-max pooling
softmax function regularization in this layer
Sentence matrix 7 × 5
3 region sizes: (2,3,4) 2 filters for each region size totally 6 filters
2 feature maps for each region size
6 univariate vectors concatenated together to form a single feature vector
2 classes
d=5
I
like
this
movie
very
much
!

Figure 1: CNN Architecture model for *"A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification"* [3].

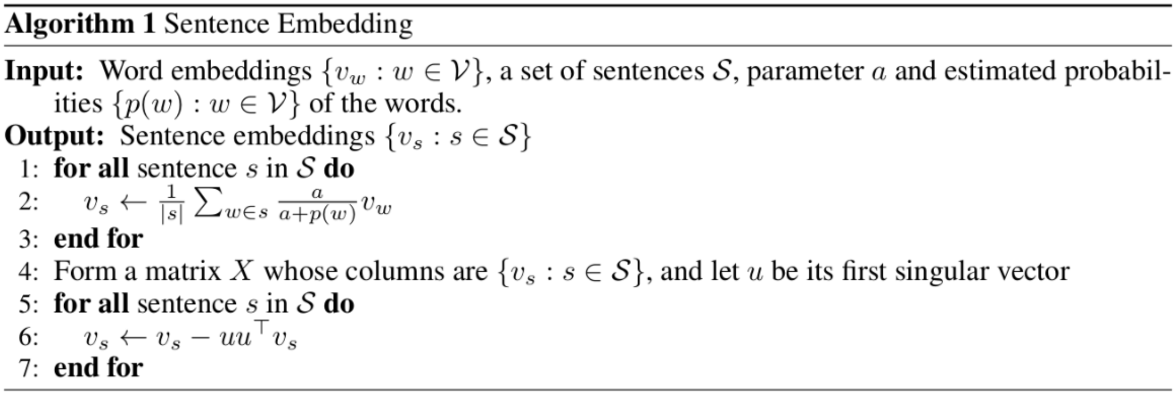
The **Run ID** on TASS 2018 for this model is *cnn-btf*-2019_03_13-04_12_55 (cnn-btf-run1).

3.2 Smooth Inverse Frequency

Here we have a new sentence embedding method that is embarrassingly simple: just compute the weighted average of the word vectors in the sentence and then remove the projections of the average vectors on their first singular vector (c̈ommon component removal)̈. Here the weight of a word w is $a/(a + p(w))$ with a being a parameter and $p(w)$ the (estimated) word frequency; we call this smooth inverse frequency (SIF). This method achieves significantly better performance than the unweighted average on a variety of textual similarity tasks, and on most of these tasks even beats some sophisticated supervised methods tested in (Wieting et al., 2016), including some RNN and LSTM models. The method is well-suited for domain adaptation settings, i.e., word vectors trained on various kinds of corpora are used for computing the sentence embeddings in different testbeds.

It is also fairly robust to the weighting scheme: using the word frequencies estimated from different corpora does not harm the performance; a wide range of the parameters a can achieve close-to-best results, and an even wider range can achieve significant improvement over unweighted average.

See Figure 2, for more detail about the algorithm of the SIF model:



```
**Algorithm 1** Sentence Embedding
**Input:** Word embeddings $\{v_w : w \in \mathcal{V}\}$, a set of sentences $\mathcal{S}$, parameter $a$ and estimated probabilities $\{p(w) : w \in \mathcal{V}\}$ of the words.
**Output:** Sentence embeddings $\{v_s : s \in \mathcal{S}\}$
**for all** sentence $s$ in $\mathcal{S}$ **do**
    $v_s \leftarrow \frac{1}{|s|} \sum_{w \in s} \frac{a}{a+p(w)} v_w$
**end for**
Form a matrix $X$ whose columns are $\{v_s : s \in \mathcal{S}\}$, and let $u$ be its first singular vector
**for all** sentence $s$ in $\mathcal{S}$ **do**
    $v_s \leftarrow v_s - uu^\top v_s$
**end for**
```

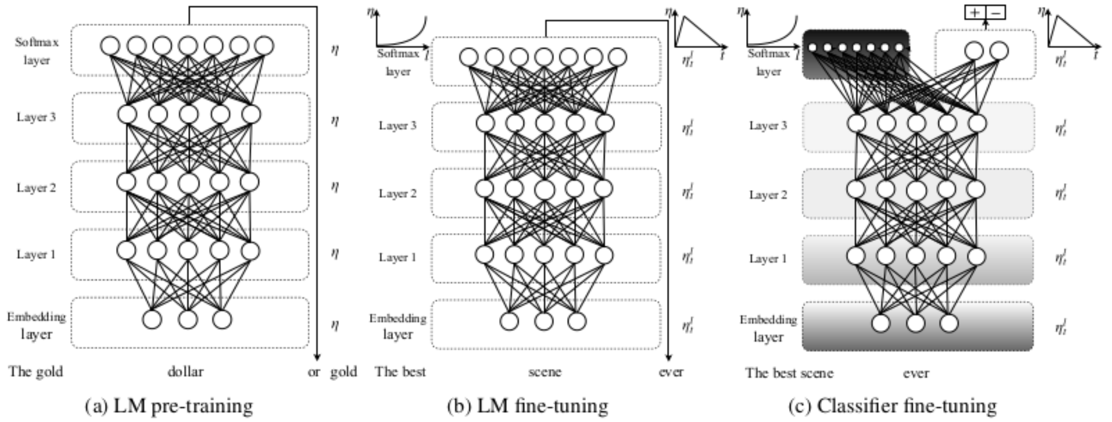
Figure 2: SIF Algorithm

The **Run ID** on TASS 2018 for this model is SIF-MLP-2019_01_30-00_27_52 (UCSP-SI-G1-run3).

3.3 Universal Language Model Fine-tuning

Transfer Learning has had a good performance on small datasets, because It uses a large pre-trained model to then adjust it for another task. ULMFit [2] use new techniques like *Discriminative Fine-Tuning*, *Slanted triangular learning rates* and *Gradual Unfreezing* to avoid forgetting knowledge on Fine-Tuning procedure.

The **Run ID** on TASS 2018 for this model is $md3$-2019_03_12-06_37_15 (m3-btf-run1).



(a) LM pre-training (b) LM fine-tuning (c) Classifier fine-tuning

Figure 3: Three stages of ULMFit.

4 Results

In order to verify our test predictions, we send our results to the TASS 2018 website, but we can't do the same for TASS 2017, so the result behind are for the development dataset.

Table 2: Task 1: International TASS Corpus 2017

TASS 2017 (Development)		
Model	Macro F1	Accuracy
CNN	0.330188	0.77075
SIF	**0.733**	**0.773**
ULMFit	0.2221	0.3837

Table 3: Task 1: InterTASS Monolingual PE 2018

TASS 2018				
		Test		Development
Model	Run	Macro F1	Accuracy	Accuracy
CNN	cnn-btf-2019_03_13-04_12_55	**0.387**	**0.432**	**0.755**
SIF	SIF-MLP-2019_01_30-00_27_52	0.399	0.398	0.602
ULMFit	$md3$-2019_03_12-06_37_15	0.270	0.299	0.309

5 Conclusions

We verify that equating the amount of data of each label we reach better results than with all the dataset for training. We also prove that considering all data of positive and negative labels we reach good results in case of CNN model. ULMFit is the worst model because we have to tuning a lot of parameters .

References

[1] Sanjeev Arora, Yingyu Liang and Tengyu Ma. A Simple But Tough-To-Beat Baseline For Sentence Embeddings. In *International Conference on Learning Representations(ICLR)*, 2017.

[2] Howard, Jeremy and Ruder, Sebastian. Universal Language Model Fine-tuning for Text Classification In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, July 2018.

[3] Ye Zhang and Byron C. Wallace A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification In *CoRR*, 2015.

[4] Eugenio Martínez-Cámara, Manuel Carlos Díaz-Galiano, Miguel A. García-Cumbreras, Manuel García-Vega and Julio Villena-Román Overview of TASS 2017 In *TASS 2017: Workshop on Semantic Analysis at SEPLN, septiembre 2017*, págs. 13-21, 2017

[5] Eugenio Martínez-Cámara, Yudivián Almeida-Cruz. Manuel C. Díaz Galiano, Suilan Estévez-Velarde, Miguel Á. García Cumbreras, Manuel García-Vega, Yoan Gutiérrez, Arturo Montejo Ráez, Andrés Montoyo, Rafael Muñoz, Alejandro Piad-Morffis and Julio Villena-Román Overview of TASS 2018: Opinions, Health and Emotions In *TASS 2018: Workshop on Semantic Analysis at SEPLN, septiembre 2018*, págs. 13-27, 2018