NLP

Research paper analysis :

A Comprehensive Analysis of Preprocessing for Word Representation Learning in Affective Tasks

Master 2 - Data Science

Hugo Rialan

Novembre 2021

Version 1.0

Link to research paper : https://aclanthology.org/2020.acl-main.514/

Linked notebook of this analysis : https://notebooks.hugorialan.site/NLP_M2DataScience/research_paper_analysis.html

Introduction

As part of the NLP course linked to the master 2 Data Science, we had the opportunity to choose among research articles to analyse one of them. I decided to choose the article : *A Comprehensive Analysis of Preprocessing for Word Representation Learning in Affective Tasks*. This article was written by Nastaran Babanejad, Ameeta Agrawal, Aijun An and Manos Papagelis in 2020. I chose it because it shows that, despite the complexity of NLP algorithms, some simple techniques can improve performance. So I was able, at my own scale, to try these techniques on a python notebook whose link is available on the front page. I also liked this article because it refers to many techniques we have seen in class : preprocessing techniques, Word Embedding, C-Bow, Skip-Gram, BERT, Glove, Fast Text and others. This article comes with the fact that there is not a lot of existing analysis on preprocessing in word vector based models applied to affective tasks. It therefore tries to answer the question of the usefulness and the possible gain in performance thanks to the application of preprocessing techniques on models using word vectors representations. First we will discuss the problem that this article tries to answer. Then, we will analyse the advantages and issues of the proposed method. Finally, I will talk about my personal take on interest on the article.

1 Preprocessing for Word Representation Learning in Affective Tasks

This analysis allows empirical testing of preprocessing techniques and measuring their performance on affective tasks. First of all, we have to ask ourselves what exactly are affective tasks. We can say from the article that it is the meeting of sentiment analysis, emotion classification and sarcasm detection. Sentiment analysis could be for example to rate a movie review between 1 and 5 stars. Emotion classification could be to give the emotion of a client textual opinion. Sarcasm detection is the most difficult task. The field of affective tasks is very used nowadays with the abundance of data as it is reminded in the paper.

The research paper wants to analyse the efficiency of preprocessing techniques applied to word vector models. They want to prove this empirically, so the texts chosen are very diverse. We will summarize the way in which they carried out this study.

Preprocessing techniques They used very simple techniques, achievable by everyone in a few lines of python code. This is also what makes the article so powerful. There were a few techniques I had never heard of. Spell checking is used to check the spelling of words. This may seem smart if you look at customer reviews where people write quickly and there is a greater chance of mistakes. I found the "negation handling" technique very clever, replacing not happy by sad for example.

This technique was found to be very effective in the analysis results. Other techniques are more basic like 'stop words', 'stemming' and others.

Training corpora These corpora were used to train the neural networks that allowed them to obtain the word embeddings. They had to be substantial enough and above all diversified so as not to be specific to particular areas. The choices were therefore not made at random. Wikipedia allows them to have a bank of texts on truly diversified subjects, since we can find everything on this encyclopaedic site. Similarly, news articles can be about anything and everything.

Word embedding models The article is specific to models based on word vectors, or word embedding. There are several ways to obtain these word vectors representations. They used the different ways we saw in class, its to say C-BOW, Skip Gram and Bert. Again, this article is based on experimentation. It is therefore necessary to try to be as exhaustive as possible to obtain interesting results. The fact that they used three different ways of having the word representations reinforces the power of the results they obtained.

Evaluation dataset To be as exhaustive as possible, they used 9 datasets. Each one deals with a particular task in the field of the affective task. There are three for sentiment analysis, three for emotion classification and finally three more for sarcasm detection.

Setup For each evaluation dataset, they converted the words by vectors thanks to the words embedding trained previously with Wikipedia and News. Afterwards, They implemented a recurrent neural network composed of LSTMs, which is perfect for processing sequential data like text. Depending on the problem, They used as output of the network a sigmoid and a BCE loss for binary classification and a softmax with a CE loss for multi class classification.What is important is when they applied the preprocessing techniques. Either they applied them on the word embedding training corpus (PRE), or they applied them on the evaluation datasets (POST), or both (PRE + POST).

After all these set-ups, they collected the data and made comparisons and analyses. The conclusions are very encouraging about the techniques used. The negation processing gave the best results. over all the preprocessing techniques. The best overall performance was achieved by applying all the preprocessing techniques except stop words removal. Finally and interestingly, They had the best results by making PRE -processing, as explained on the paragraph just above.

2 Analysis of the advantages and issues of the proposed method

The main advantage of this method is that it is very simple to set up. In addition to this simplicity it allows to have quite impressive gains in performance. This technique does not revolutionize NLP but is simple to set up and test. It is a strength because it can be applied to already existing algorithms. There are already pre-existing python libraries to do stemming, lemma, or remove stop words. Maybe the most complicated technique to set up is the negation one because you have to use the 'Wordnet' (Or another) API to get access to the antonym of a word. In any case, this advantage of simplicity allows anyone who wants to improve the performance of their algorithm to at least test some of these preprocessing techniques and to see if it works.

The stakes are high because affective tasks algorithms are becoming more and more useful. We can say that the state of the art affective tasks algorithms today are those based on word vectors.

Data boom Thanks to the rise of the Internet, the amount of data that can be analyzed has exploded in recent years. It turns out that a large part of this data are textual. This type of data is difficult to analyze but is nevertheless very important. For example, we could analyze the reviews of a movie by retrieving the tweets that contain the hashtag of this movie and classify them. People are increasingly sharing their opinions on the internet in text form. There is so much data available that now it may be necessary to have an algorithm to aggregate it all.

Use cases boom Companies are also becoming more and more digital. A good way to know what customers think of the company would be to collect reviews on the internet and then analyze them with a NLP algorithm. We can think of other uses, such as the detection of hate messages on social networks.

We can imagine thousands of use cases where the analysis of affective task could be interesting. However, it is not a simple task. This research article therefore proposes a solution to improve the performance of these algorithms. It even shows that they managed to do better than some state-of-the-art algorithms based on word vectors.

3 Personal take on interest

The main limitation of this article is, for me, that it is very empirical. It is indeed difficult to prove mathematically that preprocessing has a significant effect on the performance of an algorithm. To overcome this limitation they have used training texts, validation and algorithms as diverse as possible. But their results, as good as they are, are only valid for the texts they were able to test.

With the diversity of the techniques used, we might think that it would be wise to generalise the results obtained.

This article is very powerful because the stakes are high. We have seen that the use cases of the techniques cited in the article can be found in many problems. The article can therefore be used by a wide range of people and is adaptable to many different uses. However, it is not going to revolutionise the current best performing algorithms. The techniques used allow a maximum of a few percent more accuracy. But, from a personal point of view, I think that even a few percent gains can be very important, especially if they overlap with other gains.

It has been very interesting to experiment the techniques used in this article in Python on an IMDB dataset. I didn't always get the same results as them. But overall, I found that the performance of my neural network was enhanced by the majority of the preprocessing techniques.

I was quite surprised by the results obtained in the article. I knew that some preprocessing techniques could give performance gains but not to this extent. For me it was not intuitive that these techniques were efficient in all cases. Now, I know this and I will be more diligent in pre-process my text before I use it in algorithms.

Finally, you should test the techniques mentioned on your own problem, your own texts and your own models. There may be a better technique for one problem than for another. Maybe the negation technique is not always the best technique, depending on the number of data you have or other factors.

Conclusion

This analysis document is very complete and very interesting to read. To summarize, it proposes the idea that preprocessing is a very important step before using a text in an affective task algorithm. It only applies to models based on word vectors or embeddings. It proposes experiments on the use of preprocessing at the time of embedding training (Pre-processing) as well as at the time of text classification (Post-processing). The results are quite surprising by the performance of the techniques used. Indeed, even a simple preprocessing technique can result in an improvement. It has been shown that the techniques that work best are 'negation' and 'stopwords'. Finally, the best way to get even better results is to apply these techniques during the training phase of the embedding. I tried to reproduce some of these results at my scale in a notebook, it was a very formative exercise. To conclude, NLP is a very important research topic, it is the proof by experience that has been used in this article.