Sentiments with Large Language Models as Multi-Class classifications problem

**Abstract**: Sentiment analysis has been a topic of many researches and interest. In this article, computational steps are shown and worked out to show, how we can have multiple centres in the sentiment classification task. This does not stop the existing works but provides more avenues to open research applications more. Conceptualize a 5-class classification task for sentiments, what it means to look in more than three-dimensional spaces, and what it means to have more insights. This can be taken as an ML exercise, both practical & theoretical, till it’s real applications are out. However, major application is that new words can be assigned sentiment without a supervised approach, but only via Trained Large Language models. The 3-class sentiment orientation is well defined as till now we consider subjectivity in three spheres. But consider the case, of very high positivity, positivity, neutrality, and very negative to negative sentiments. Hence the need to define, more than three typical classes. Here we provide the algorithm for assigning the multi-class sentiments and their degree for any new word so that not much of resources are used.

1. Introduction

The unsupervised sentiment assignment can be thought of as an n-class classification task, here the classification can be supervised or un-supervised in nature. The value of n now restricts to 2 or 3, till now. However, in this article, we present the case wherein more than two or three class problems can be formulated from given resources without the need of extra manual tagging, as was part of tagging into three classes, a huge manual work. This does not say that current things are not required, no, it just says this is also an area of research that can give dividends if analyzed well, and can lead to applications worth solving with the proposed architecture.

2. Sentiment Assignments to New Words

The algorithm starts as follows

• 1. Input the text file, the text file is where the sentiments are to be measured of some text in its relative contexts. It can have one sentence, one word or a page, or a book of text data.

• 2. Define some sentiment centres, typically in sentiment analysis tasks, positive, negative, and neutral classes are defined in supervised and even unsupervised classifications same three class works have been performed. These are well defined as till now we consider subjectivity in three spheres. But consider the case, of very positivity, positivity, neutrality, very negative to negative.

• 3. Here as many sentiment epicentres can be created as is required in an application, the current article lays emphasis will be on 5 classes, and classification.

**The following are taken from my presentation in slides, hence this format**

2.1. Typical sentiment analysis task-3 classes

• The typical sentiment Analysis Task is a three Class Classification problem, that can be either supervised or unsupervised.

The three classes in the typical Sentiment Analysis Tasks are

• 1. Positive Class

• 2. Negative Class

• 3. Neutral Class

2.2 Non-trivial sentiment alignments

Proposed Sentiment Analysis Tasks are non-trivial and are given as follows

• 1. Extreme Positive Class

• 2. Positive Class

• 3. Negative Class

• 4. Extreme Negative Class

• 5. Neutral

2.2. Five class sentiment orientations

Proposed Sentiment Analysis Tasks are

• 1. Extreme Positive Class, happy

• 2. Positive Class, good

• 3. Negative Class, bad

• 4. Extreme Negative Class, sad

• 5. Neutral, neutral

These may not be best assigned 5 classes to be assigned. But a good way to start the research in this area. This is a good way to start this as an AI/ML excersise/research.

2.3. Example

• Consider the word “ Nice “ to be assigned to of the 5 spatially distributed classes. This can be some text fragments as well.

• Let us start with the following 5 words as centers of the distributions:

• 1. happy

• 2. good

• 3.neutral

• 4. bad

• 5. sad

• Let us call them Epicenters, for now, for Sentiment Classification task

2.4. Choose the epicentres of sentiment classification task – 5 class problem now

• 1. happy

• 2. good

• 3.neutral

• 4. bad

• 5. sad

2.5. Inputs used are

• LLM Model, these are pretrained

• The LLM model was computed on pretrained GoogleNews-vectors-negative300 data

3. Define the input to be classified: Here, for example, it be word “nice”

• The words similar to Nice are:

• Good, lovely, neat, fantastic, wonderful, terrific, great, awesome, nicer, decent

• These word can be found with Python code

3.1. Most similar words near epicentre 1

• The words similar to  good are:

• great ,  decent , nice , excellent , fantastic , better , solid , lousy ,

• The words similar to happy are:

• glad , pleased , ecstatic , overjoyed , thrilled , satisfied , proud , delighted ,  excited ,

• These word can be found with Python code

3.2. Most similar words near epicentre 2

• Most similar to bad

• terrible , horrible , Bad , lousy , crummy , horrid , awful , dreadful , horrendous ,

• Most similar to sad,

• saddening , Sad , saddened , heartbreaking , disheartening , saddens\_me , distressing , reminders\_bobbing ,

• These word can be found with Python code

3.3. Epicentres to clusters

Based on these best similar words, we make clusters with these best words

Change of LLM models shall change the output

3.4. Compute similarity of input word to clusters

* Similarity of nice to cluster centre of good, which contains the best words around word good.
* The algorithm outputs, 1 as similarity score, as maximum of all these values
* great 0.64546573
* terrific 0.6552368
* decent 0.5993332
* nice 1.0
* excellent 0.47978145
* fantastic 0.6569241
* better 0.38781166
* solid 0.42754313
* lousy 0.3887929

3.5. Compute similarity of input word to clusters

* Similarity of nice to cluster centre of happy, which contains the best words around word happy.
* The algorithm outputs, 0.5067 as similarity score, as maximum of all these values
* glad 0.5067967
* pleased 0.31258228
* ecstatic 0.3099612
* overjoyed 0.2611948
* thrilled 0.32813224
* satisfied 0.21507472
* proud 0.34908974
* delighted 0.27841687
* excited 0.35698736

3.6. Compute similarity of input word to clusters

* Similarly other assignments to other Epicenters are made.
* And the word is assigned a sentiment score depending on what is its similarity to a sentiment class and how many sentiment classes there are. The word here assigned to class of good.

Conclusion and Future Work

These are exercises and good way to progress in this field. There are lot of future works which shall progress once a while in these technical notes. Right now take it as an AI exercise. A major application is that new words can be assigned sentiment without supervised approach, but only via Trained Large Language models.