CS 735/835 Information Retrieval Project - Evaluation

Shayan Amani
(993550898, sa1149)

Soheil Gharatappeh
(933639024, sg1147)

1 Introduction

The problem to be solved is called *Affect in Tweets*, and it is defined as how we can determine the user's emotions according to their tweets to know what sort of feelings they have, and to what degree they have those feelings. Four emotions are studied here are semantically distinct emotions; *anger, fear, joy, sadness.* Our task is; given a tweet, and an emotion X, we have to determine the intensity or degree of emotion X felt by the speaker. The data we have in our output is a value between 0 and 1, in which the maximum value of 1 stands for feeling the maximum amount of emotion X, and the minimum score of 0 stands for feeling the least amount of emotion X. This can be interpreted as, when the user is having a maximal/minimal mental state toward/away from emotion X.

In this project, for sake of simplicity, we only consider one emotion; joy, mainly because we all are looking for it in this day and age. Hopefully, this research will help us to know the underlying meaning of the language we use everyday.

2 Motivation

Tweeter is one of the most famous social app that gives this opportunity to its users to communicate their ideas via text-based messages. These feature has made tweeter a unique platform, for doing all sorts of language/text-based analysis. In order to understand how important is this research, just think about that the president of the most powerful country in the world, uses tweeter on daily basis to convey his messages and ideas to the world. It is a crucial piece of information for political strategist, to have an understanding of the true feeling of the user about a particular issue, to come up with ideas in their negotiations to solve, for example, international disputes, or make deals between countries. The outcome of our analysis is not limited to only the world of politics. Knowing the users' emotion (at the moment of writing the tweet, or about a specific topic in general), can be useful for businesses and marketing strategists to better sell their products. By knowing about emotions of people towards something in a specific area, we can come up with a better business plan that works best for that society. We can adjust advertisements produced for them to their tastes, and use elements that can better convince them to buy that product.

3 Method

To solve the problem of finding users emotion given a tweet using conventional IR methods, there are three possible approaches; using a *Vector Space Models*, and *Language Models*, or using *Regression* methods. Regardless of the method we are going to implement, we need a document that somehow plays a reference role for happiness. If we look at this as the similarity perspective, the document shows the direction of *pure* happiness in our space. So, our job is to find the similarity of tweets to the reference document, to decide whether the document points to the same direction as the reference or not. In section 3.1, we discuss how we obtained our happiness reference document.

3.1 Happiness Reference

To understand how much a tweet is happy, we need to have a good reference document. If we, somehow, can generate this document, we can make comparison between the words in the given tweet, and our reference and say how much they look similar. We can make this document by adding "happiness" synonyms to our document. So, we searched the keyword "happy" in the website http://www.thesaurus.com, and add all of relative synonyms to our database. Next, we opened each of the synonyms, and added them to our database (duplicate word have not been removed on purpose).

To boost up our database, we then used another helper database. This document is taken from the competition Multilingual Emoji Prediction [1]. In the provided database, they associate each tweet to an emoji. So, we took tweets that are associated with happy faces emojis, namingly; smile, tears of joy, wink, and etc. and add them to our database. The idea of improving our database using other tweets is mainly based on the fact that the language used in tweets are different than what we can find in dictionaries. So, if we only use dictionaries to obtain words related to happiness, we cannot guarantee that we have a reasonable data base that contains vocabularies used by people in their social life that indicates their happiness.

To enrich the database even further, twitter API was employed. This API lets users to search for an specific keyword in the last available tweets. The powerful interface of twitter API also gives users this option to choose the language of retrieved tweets. So, the retrieved tweets can be easily added to the database, with no further filtering.

Here, it is assumed that if a tweet contains some happy words such as; "haha", "Lol", ":)", the user was happy when generating the tweet. So, all the words in the tweet can be conceived as happy words, and added to the database (which is a naive assumption).

3.2 Vector Space Model and Language Model

In the previous section, we discussed how we can obtain a document that shows the direction of happiness in space. The task here is to calculate the similarity between the happiness document, and a given tweet. We use different methods such as; lnc.ltn, bnn.bnn, anc.apc in vector space models to get Tf-idf scores, which shows how similar tweets are to the happiness document. In addition, we applied some language model methods such as; *Laplace, Jellinek, and dirichlet* Smoothing.

As it is noted in the competition website, the reported scores in the training and evaluation database has no meaning by itself. Therefore, it should be seen as a relative value between [0,1], considering the explanation given in 1. And, as we know, none of the above-mentioned method gives us back a value between [0,1]. So, what we need to do is to normalize them into the interval of [0,1], and do the evaluation on them. Score files and the original dataset can be found here. They are formated as:

```
id[tab]tweet[tab]emotion[tab]score
```

3.3 Regression

The second possible approach is to look at this problem as a regression problem. Firstly, it is noteworthy to mention that this is not a normal IR problems. IR problems can be categorized as a classification problem, in which a user issues a query, and the system tries to show some relevant documents to the user. So, at the end, the retrieved documents can be *relevant* or *non-relevant*. Consequently, the ground truth contains information about the fact that the retrieved documents were relevant or not. However, here, the problem is not determine the fact that a tweet is happy or not. But, the problem is to determine to what level is a tweet happy.

This problem can be seen as a regression problem; in which a set of training data is given, and a score (and indicator of the intensity of a emotion) for a new given tweets is asked. This is basically a SVR problem. The first task in a SVR (support vector regression) problem, is to come up with a good feature matrix. It is reasonable to use Tf-idf, and language model data as features. These IR methods reflect a relation between a query and a document. So, they can be a good indication of the fact that whether the query vector was similar to the document or not.

To train a regression model based on SVR, 60% of the data is taken out as the training set. Then, the rest 40% is also divided into two sets; validation set and test set. The validation set is then used for choosing tuning parameter (C), and then the error of fully trained model will be calculated on test set. According to figure 1, C is chosen to be $1.5 * 10^5$.

Implementation of the regression algorithm for this project is done on Python. Sklearn package is particularly used to train a SVR model, based on Radial Basis Functions kernel. The source codes for this part can be found in here.

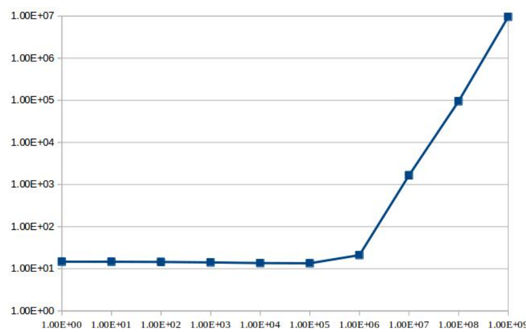

Figure 1: Choosing tuning parameter C

IR method	bnn.bnn	Laplac	Jelinekmercer
Thesauri	0.1905	0.1661	0.1778
Thesauri + "haha"	0.0878	0.0864	0.0893
Thesauri +":)"	0.1089598838	0.0999376552	0.1199213444
Thesauri +"lol"	0.0431435215	0.0500752162	0.0492815959
Thesauri + smily	-0.0238401198	0.1195103257	-0.0238401198

Table 1: Correlations

Apparently, because there is no meaningful correlation between the features and the scores, the trained model could not perform well, and showed a bad result in Pearson correlation test (-0.002).

4 Evaluation

Two main evaluation measures used in the first phase of this study is *Pearson Correlation Coefficient*, and *Spearman*. Pearson Correlation is a measure of the linear correlation between two variables X and Y. It has a value between $+1$ and -1, where 1 is total positive linear correlation, 0 is no linear correlation, and -1 is total negative linear correlation. However, in the second phase, a modified version of Pearson Correlation and Spearman is used, which only considers tweets that obtain scores more than 0.5.

To do evaluation, an implemented version of Pearson and Spearman methods, provided by the competition website is employed. The results for the mentioned IR methods is put in table 1. In this table, the obtained results using different joy database is demonstrated. As it is clear from the table, increasing the size of document using meaningful data, not only does not improve the results, but also causes lower correlations in results.

5 Conclusion

In this research study, two different approaches to Information Retrieval were compared; Vector Space Model and Regression.These two methods have two completely different points of views in looking at a problem, one does not take any sort of feedback from the world, and only works on what we *assume* to be true about IR. However, the other one, only works based on user's feedback.

Despite the first guesses, IR methods worked better here. The main reason for the bad performance of SVR was the poor quality of features. In machine learning approaches, the most important key element of the solution is features. Here, a large majority of potential features were neglected. In here, a long list of good features that can help to solve the problem can be found.

References

[1] Semeval-2018 task 1: Affect in tweets. https://competitions.codalab.org/competitions/17344.

[2] Semeval-2018 task 1: Affect in tweets. https://competitions.codalab.org/competitions/17333#learn_the_details-overview.

[3] Mohammad Salameh Saif M. Mohammad, Felipe Bravo-Marquez and Svetlana Kiritchenko. Semeval-2018 task 1: Affect in tweets. *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018*, 2018.