Software Engineering Practice

KF5012

Sentiment analysis project

Team name :’Allo ‘Allo World

Project statement:

Our proposal is to build a text classification solution. This will be a supervised learning and classification problem. We will base the initial implementation around movie descriptions and train a model to output the movie genre. The ambition is to improve on an initial proof of concept. We will expand the known genres and dataset as well as looking to improve the accuracy by using different classification algorithms. We will also look to use stop words to improve the accuracy. Due to losing a team member we will not be doing the GUI side mission so we will look to document iteration via improved code, data manipulation and output with accuracy measures.

As a team, we discussed several ideas for the project and reached a consensus that text classification was an interesting topic for all of us, as it is used across such a broad range of applications and for many different purposes.

Our project will be focused on text classification, although sentiment analysis was a close second in our discussions. Text Classification is something which has a range of uses, such as making navigating to desired outcomes streamlined and seamless.

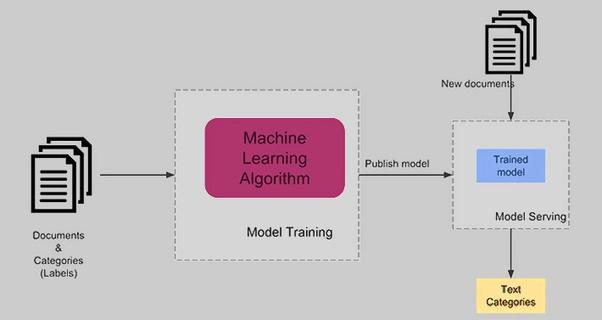
We have aimed to achieve a piece of usable software which is simple to learn to use, powerful and shall be based upon supervised learning and classification. The objective is to first gather data which contains movie descriptions, either from IMDb or Rotten tomatoes and achieve increasing levels of accuracy pertaining to the genre of the movie. Our team had thoughts that the solution could be scalable, and would not be limited to movies but indeed series, theatre and even TV programmes.

Our team did consider doing a sentiment analysis project, as ring fencing and adequately dealing with sexism, racism, extremism or categorisation of emails into legitimate or spam and generating business leads are fascinating, but we knew that there would be ethical considerations and we did not want to attempt to tread those waters.

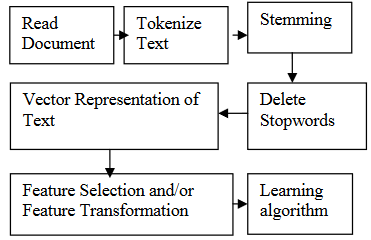
**Why the problem is important & justification**

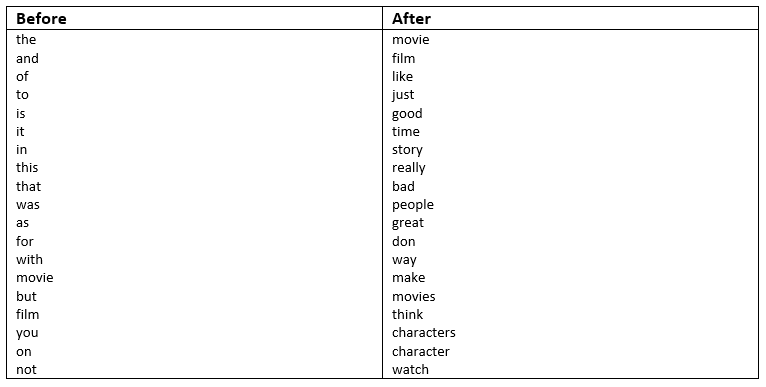
Within the use cases of text classification, overall, there are many advantages and benefits. Data is growing exponentially all the time, being able to produce tangible relevant data in a timely way is pivotal.

Text classification is important from an efficiency perspective on entertainment platforms, for example, if you had to manually read descriptions of films in order to find their genre and were only interested in comedy films, you might read many descriptions before you found one relevant to your wants. Being able to categorise movies by genre is what our team found an interesting problem and the justification for providing a solution was to provide a great user experience, giving clarity, so you are able to quickly arrive at what you are looking for depending on your preferences.

A simple model of our solution:

Another way to visualise text classification shown in this flowchart



Removing stop words will narrow and make the data more usable.

“An ancillary feature engineering choice is the representation of the feature value. Often a Boolean indicator of whether the word occurred in the document is sufficient. Other possibilities include the count of the number of times the word occurred in the document, the frequency of its occurrence normalized by the length of the document, the count normalized by the inverse document frequency of the word.”

TF-IDF is short for **t**erm **f**requency–**i**nverse **d**ocument **f**requency and It reflects how important a word is to a document in a collection. TF-IDF values increase in relation to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

There are several ways a supervised learning approach can be utilised. The first method is called ranking classification, which assigns multiple categories to a given input and hard categorisation is assigning just a single category. It is apparent that creating “features” (strings containing more than one word) can appear thousands of times, should the dataset be large enough, there are reduction methods which can be considered such as selecting a subset of the data or transforming some original features into revised versions.

With a choice of algorithms and approaches to achieve this solution, I present an exploration of existing methods, solutions and related work.

“A document is a sequence of words. So each document is usually represented by an array of words. The set of all the words of a training set is called vocabulary, or feature set. So a document can be presented by a binary vector, assigning the value 1 if the document contains the feature-word or 0 if the word does not appear in the document.”

Highlighting the importance of removing stop words, which helps to produce more accurate results from the early stages of the development cycle, consolidation of a dataset to use only data which is deemed relevant to the classification is sensible and is known as feature selection and something which this project shall adopt. Not only is feature selection a valuable exercise in terms of clarity and reduces overfitting, it also has a much smaller footprint on computational resources.

Stemming, as detailed in an earlier diagram relates to the cleaning of the dataset, to remove null values, spelling mistakes and other non-expected characters. There is some debate as to how effective stemming is, some researchers state that it does increase the performance of the classifiers and some give the notion that aggressive stemming could alter things in a way which isn’t conducive with the desired outcome.

“there are some doubts on the actual importance of aggressive stemming, such as performed by the Porter Stemmer”

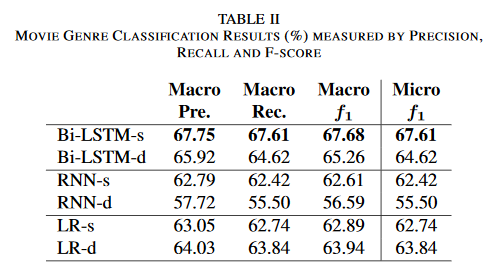
Work which relates to our project has been carried out at the middle east technical university, where they used plot summaries to classify movie genres. The methodology which was employed was a bi-directional LSTM, which is a type of recurrent neural network. As this project used plot summaries as their data, they managed to attain granularity through breaking the data into individual sentences, and then considering the probability of the genre. Work has also been carried out when genre classifications can be done by analysis of visual and audio features, even abstract methods such as using the colours of movie posters have been carried out.

“We first converted all texts in the plots to lowercase. Next, we eliminated all punctuation marks except the ones that separate the sentences. Additionally, we eliminated the stop-words. We also divided plot summaries into sentences for the sentence-level classification task.”

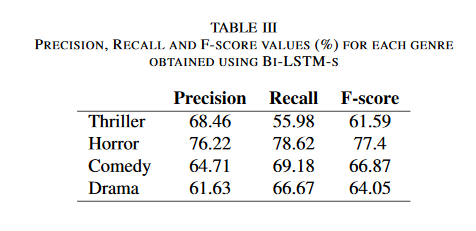
This shows how the data is manufactured into a more usable format and is something which our team will also follow, also, this was deemed as a “sentence level” approach and gave the opportunity to contrast that performance with the text of the entire plot description, which was called the “document approach”. The results were that at baseline the Bi-LSTM was more accurate than a logistic regression model and a traditional RNN.

“Bi-LSTM-s, significantly outperforms the other methods in terms of both macro f-score and micro f-score which are 67.78% and 67.61%, respectively. We also observe that sentence-level approach importantly boosts the performance when the recurrent neural networks are used for the classification. Bi-LSTM-s and RNN-s perform superior than the document level settings of the same networks. On the other hand, document-level approach gives slightly better performance compared to sentence-level approach when LR is used for training.”

The ambition of their project was to increase how much data their model could accept and also see if they could attribute multiple genres to the movies.



“We also share the values of precision, recall and f-score of Bi-LSTM-s method for each genre in Table III. According to the results, the proposed method performs better while estimating the genre of Horror. On the other hand, the lowest performance is obtained while predicting the genre of Thriller”



Shown here is that there are some adjustments that could be made in order to increase accuracy when identifying Thriller movies, this could be achieved by further developing features common in Thriller plots and analysing why the results for identifying Horror movies was so high.

We have given weight to this being a valuable project to pursue, as our aim was always to attempt to make the solution scalable and able to do classification across a range of different mediums and to potentially provide multi-genre classifications, NetFlix for example has over 75,000 categories.

References

Movie genre classification from plot summaries using bidirectional lstm **–** Pinar Karagoz et al 31 Jan.-2 Feb. 2018.

WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4, August 2005, pp. 966-974V.Tampakas et al.