International Patent Classifier

Machine Learning Models To Classify the Section of a Patent

By Mohammad Faizan

First of all thank you very much for giving me this opportunity of learning and implementing such an interesting and practical task.
It really helps me a lot to understand the different aspects of Machine Learning and Natural Language Processing.
I really enjoyed while doing this wonderful task.

What ,Why,How,Challenges

1) First i divide the whole xml file into smaller files (data.py)

(as the xml file was very big and it was very difficult to read it as my Laptop was not responsive enough for such large file so i read it for about 1000 lines and i came across that for each patent there is a XML script within the file so I applied DIVIDE AND CONQUER strategy and made seperate files for each script of each individual patent in this way the whole dataset become manageable)

2) and then I extract the necessary data from the xml files into dictionary(extraction.py)

(i read the each file one by and extracted the necessary data from each file and put the textual data of abstract, title,description and claims along with its corresponding section into dictionary as dictionary is quite faster and easy to handle when there is task of hashing)

3) from that dictionary of the dataset i made the tsv file for ML models to train with(extraction.py)

(then i move the data from dictionary to the tsv file as it is easy to rad from the tsv file and it was taking

too long to read data from xml files and since i need that data in future so time of extracting data from XML file would be saved)

4)then after importing necessary classes ,read the file1.tsv's file's data into variable called data

5) after that I pre processed or cleaned the data by removing the punctuations ,unnecessary symbols and stopwords and then i convert the words into there root form (Using WordNetLemmatizer)

(The cleaning of dataset is very necessary step in NLP as there are symbols and words which are unnecessary for the training of the model and apart from increasing the overhead they do not serve any purpose and for Lemmatizing i use WordNetLemmatizer as it converts the words into there root form by considering it textual context though it is slower than other stemmers like PorterStemmer ,etc but it is quite accurate)

6)then after cleaning the dataset i split the dataset into training and testing datasets.

7)After all the above tasks the final task was to choose the most efficient and appropriate classifier to train the dataset with.

So to choose the appropriate classifier i need to study a lot and i came to a conclusion that without trying the Algorithm i can not decide which one is better.

But before that i used Pipeline module of python because for the machine learning algorithms(or models) it is necessary to fetch them digital data as they don't understand the textual data, and for this purpose i have to use word to vector transformers like CountVectorizer(convert the words into their frequency count[bag of words] or vectorizer) and TF-IDF vectorizer(convert the documents into the TF-IDF vector considering the context and weightage of a particular word in the sentence)

Term Frequency(TF)=(# of repetition of words in a sentence)/(# of words in a sentence)

Inverse Document Frequency(IDF)=log((# of sentences)/(# of sentences containing that word))

Then,I applied the following Classifier Algorithms

1) Naive Bayes(MultinomialNB)

Advantages: This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.

Disadvantages: Naive Bayes is is known to be a bad estimator.

In this classifier first i use both transformers(CountVectorizer and TF-IDF vectorizer) but it gave very poor accuracy of 0.56 then i remove the TF-IDF vectorizer then it gave accuracy of 0.71 because Multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts.

Since there are so many parameters ,attributes and hyperparameters for a particular classifier.

So to find which combination is most appropriate to get best out of the classifier was very difficult and hectic

Then i came across the GridSearchCV for choosing the most appropriate combination out of the given combinations. Though it solved my issue of selecting a combination and train-test the model on it but it was taking to long to get the desired result.

So i search more and came to know about RandomizedSearchCV and it was quite faster than the GridSearchCV but the problem was that it was not accurate all the time so you have to run it again and again to get the best accuracy respectively.

After training and testing the model on MultinomialNB() i used other classifiers,

2) Stochastic Gradient Descent Classifier(SGDClassifier)

Advantages: Efficiency and ease of implementation.

Disadvantages: Requires a number of hyper-parameters and it is sensitive to feature scaling.

I train this model on multiple combinations of hyperparameters and accuracy was ranging from 0.78 to 0.83.

Then i used GridSearchCV and RandomizedSearchCV to get the combination of most accurate hyperpara-

meters .

Then i did the same for other classifiers:

NOTE: I did not use GridSearchCV and RandomizedSearchCV for every classifier because the approach is same for all the classifiers.

3)Logistic Classifier

Advantages: Logistic regression is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable.

Disadvantages: Works only when the predicted variable is binary, assumes all predictors are independent of each other and assumes data is free of missing values.

This model gave me the same accuracy range as that of SGDClassifier.

This might be because SGDClassifier itself works Linear SVM (when loss='hinge') and Logistic Classifier when (log='log').

4)K Neighbors Classifier

Advantages: This algorithm is simple to implement, robust to noisy training data, and effective if training data is large.

Disadvantages: Need to determine the value of K and the computation cost is high as it needs to compute the distance of each instance to all the training samples.

5)Decision Tree Classifier

Advantages: Decision Tree is simple to understand and visualise, requires little data preparation, and can handle both numerical and categorical data.

Disadvantages: Decision tree can create complex trees that do not generalise well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

6)Random Forest Classifier

Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.

7)Support Vector Machine

Advantages: Effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient.

Disadvantages: The algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

The results of all the classifiers in different cases is given in the execution log folder and classifier.ipynb.

Suggestions and Improvements

1) Since i used multiple classifiers which results in better accuracy and computation as there was comparative study among them.
2) For improvements and save some of my time i used GridSearchCV and RandomizedSearchCV which i think definitely help me to improve my results and accuracy.
3) Since i faced a lot of problem while i was performing GridSearchCV as it was taking too long to train and test despite using all four cores of i5 8th Gen ,so this problem can be handled by using fast GPUs and TPUs to enhance parallelism and concurrency.Which ultimately results in fast computation.
4) We can filter the dataset more efficiently so that there will be more accurate results.

And there will be more improvements and methods to enhance the accuracy of models

Which i believe you all people will teach me .

Well of course if I get selected and if you think i am capable of that.