Course Title: ARTIFICIAL INTELLIGENCE LABORATORY

Course Code: CSE 418

TEXT CLASSIFICATION

Supervised By

Supta Richard Philip

Lecturer, Department of Computer Science & Engineering

City University, Dhaka, Bangladesh

Submitted By

Md. Shalman Shah – 153402312

Nibir Setu – 153402330

Farhad Hossain – 153402324

Md. Masud Mia – 153402302

13th March, 2019

Letter of Acceptance

This project worked entitled “TEXT CLASSIFICATION” submitted by Nibir Setu, Farhad Hossain, Md. Masud Mia, Md. Shalman Shah, to the department of Computer Science and Engineering, City University, Dhaka, Bangladesh is accepted by the department in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering.

Project Supervisor

Supta Richard Philip

Lecturer

Department of Computer Science & Engineering

City University, Dhaka, Bangladesh

ACKNOWLEDGEMENT

I would like to thank The Almighty God that gave great health upon us in the struggle to accomplish our goals in this project and we greatly thank our supervisor Supta Richard Philip, Lecturer Department of Computer Science and Engineering, City University. She guided us Encouraged us, for her help, advice and every correction of the way. We are also grateful to all our teachers.

Contents

1	ABSTRACT	2
2	INTRODUCTION	2
3	NATURAL LANGUAGE PROCESSING (NLP)	3
4	STANDARD NLP WORKFLOW	3
5	TEXT CLASSIFICATION	4
6	WHAT CAN WE DO WITH TEXT CLASSIFICATION	5
7	TEXT CLASSIFICATION PIPELINE	6
8	SETTING UP THE ENVIRONMENT	7
9	TRAINING SETS	8
10	FUTURE SCOPE	9
11	CONCLUSION	9
12	REFERRENCE	10

ABSTRACT

Text classification is the process of classifying documents into predefined categories based on their content. It is the automated assignment of natural language texts to predefined categories. Text classification is the primary requirement of text retrieval systems, which retrieve texts in response to a user query, and text understanding systems, which transform text in some way such as producing summaries, answering questions or extracting data.

INTRODUCTION

There are numerous text documents available in electronic form. More and more are becoming available every day. Such documents represent a massive amount of information that is easily accessible [1]. Seeking value in this huge collection, organization requires much work to organize documents, but this can be automated through data mining-an artificial intelligence technique. The accuracy and understanding of such systems greatly influence their usefulness. The task of data mining is to automatically classify documents into predefined classes based on their content. Many algorithms have been developed to deal with automatic **text classification.**

The most common techniques used for this purpose including naïve Bayes classifier, association rule mining [2], genetic algorithm, decision tree etc. The discovery of these relationships among huge amounts of transaction records can help in many decision making process. On the other hand, the naïve Bayes classifier uses the maximum a posteriori estimation for learning a classifier.

It assumes that the occurrence of each word in a document is conditionally independent of all other words in that document given its class. Although the naïve Bayes works well in many studies [3] it requires a large number of training documents for training accurately. Genetic algorithm starts with an initial population which is created consisting of randomly generated rules. Each rule can be represented by a string of bits. Typically, the fitness of a rule is assessed by its classification accuracy on a set of training examples.

NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing (NLP) is all about leveraging tools, techniques and algorithms to process and understand natural language-based data, which is usually unstructured like text, speech and so on.

Natural Language Processing (NLP) is a wide area of research where the worlds of artificial intelligence, computer science, and linguistics collide. It includes a bevy of interesting topics with cool real-world applications, like named entity recognition, machine translation or machine question answering. Each of these topics has its own way of dealing with textual data. But before diving into the deep end and looking at these more complex applications, we need to wade in the shallow end and understand how simpler tasks such as text classification are performed.

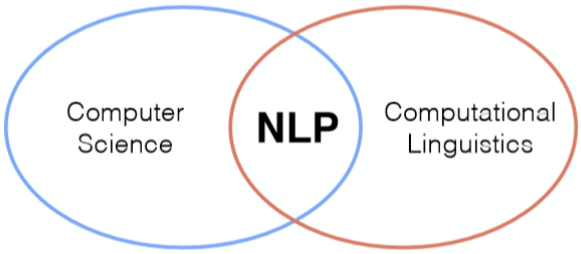
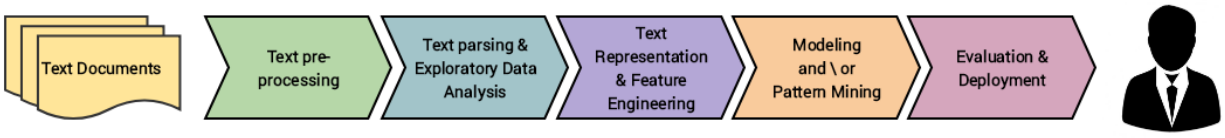


Figure: a simple view of understanding NLP

STANDARD NPL WORKFLOW

Assuming that the aware of the CRISP-DM model, which is typically an industry standard for executing any data science project. Typically, any NLP-based problem can be solved by a methodical workflow that has a sequence of steps. The major steps are depicted in the following figure.



We usually start with a corpus of text documents and follow standard processes of text wrangling and pre-processing, parsing and basic exploratory data analysis. Based on the initial insights, we

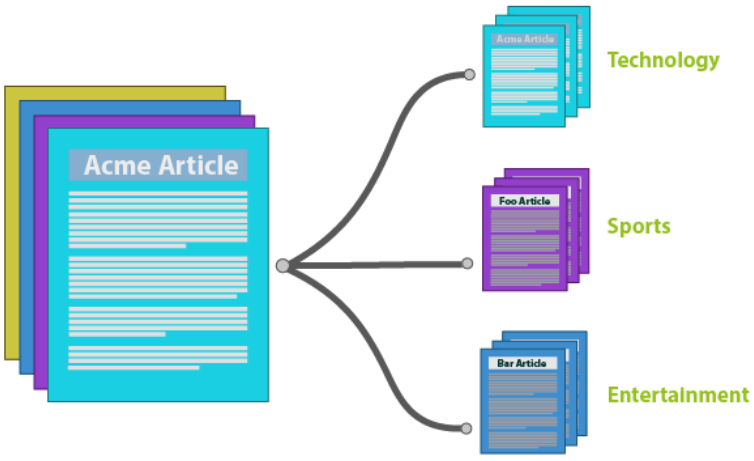
usually represent the text using relevant feature engineering techniques. Depending on the problem at hand, we either focus on building predictive supervised models or unsupervised models, which usually focus more on pattern mining and grouping. Finally, we evaluate the model and the overall success criteria with relevant stakeholders or customers, and deploy the final model for future usage.

TEXT CLASSIFICATION

One of the widely used Natural Language Processing & Supervised Machine Learning (ML) task in different business problems is "Text Classification", it's an example of Supervised Machine Learning task since a labelled dataset containing text documents and their labels is used for training a classifier. The goal of text classification is to automatically classify the text documents into one or more predefined categories.

Some examples of text classification are:

- Understanding audience sentiment from social media
- Detection of spam & non-spam emails
- Auto tagging of customer queries
- Categorization of news articles into predefined topics.



Binary: Only two categories which are mutually exclusive.

- Spam detection, Anomaly detection, Fraud detection.

Multi-class: Multiple categories, mutually exclusive.

- Language detection.

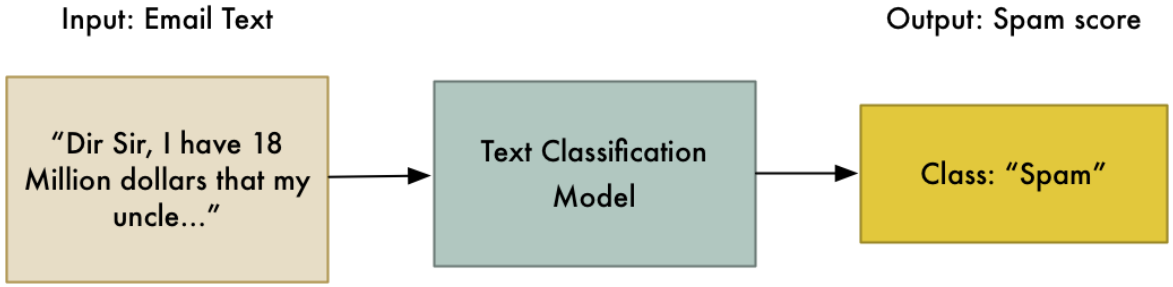
Multi-label: Multiple categories with the possibility of multiple (or none) assignments.

- News Categorization, Marketing profiling.

WHAT CAN WE DO WITH TEXT CLASSIFICATION

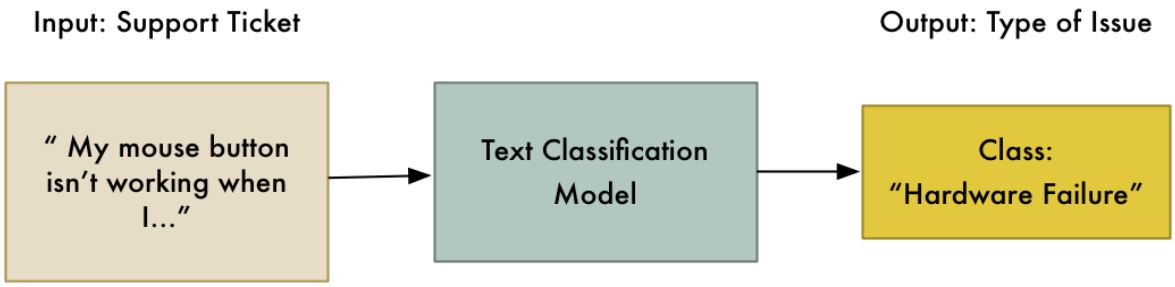
We've seen that we can use text classification to automatically score a user's review text. That's a type of sentiment analysis. Sentiment analysis is where you look at text that a user wrote and you try to figure out if the user is feeling positive or negative.

There's lots of other practical uses of text classification. One that you probably use every day as a consumer without knowing it is the email spam filtering feature built into your email service. If you have a group of real emails marked as **"spam" or "not spam",** you can use those to train a classification model that automatically flags spam emails in the future:

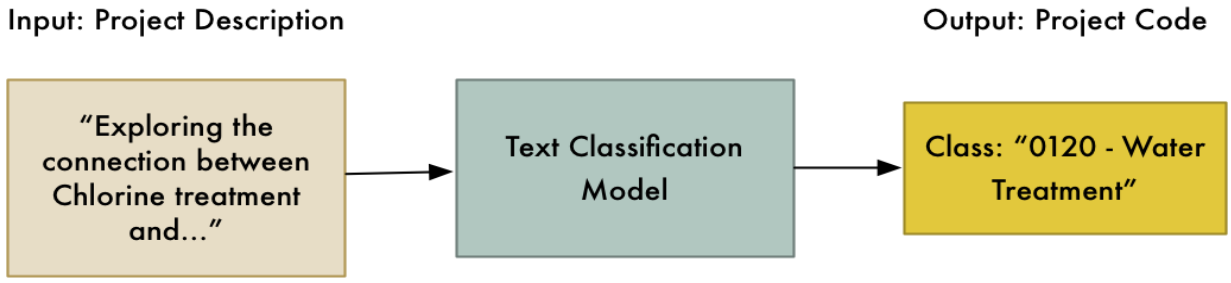


Along the lines of spam filtering, you can also use text classification to identify abusive or obscene content and flag it. A lot of websites use text classification as a first-line defense against abusive users. By also taking the model's confidence score into consideration, you can automatically block the worst offenders while sending the less certain cases to a human moderator to evaluate.

You can expand the idea of filtering beyond spam and abuse. More and more companies use use of text classification to route support tickets. The goal is to parse support questions from users and route them to the right team based on the kind of issue that the user is most likely reporting:



Classification is also useful for sorting and labeling documents. Imagine that your company has done thousands of consulting projects for clients but that your boss wants them all re-organized according to a new government-mandated project coding system. Instead of reading through every project's summary document and trying to decide which project code is the best match, you could classify a random sampling of them by hand and then build a classification model to automatically code the remaining ones:



TEXT CLASSIFICATION PIPELINE

An end-to-end text classification pipeline is composed of following components:

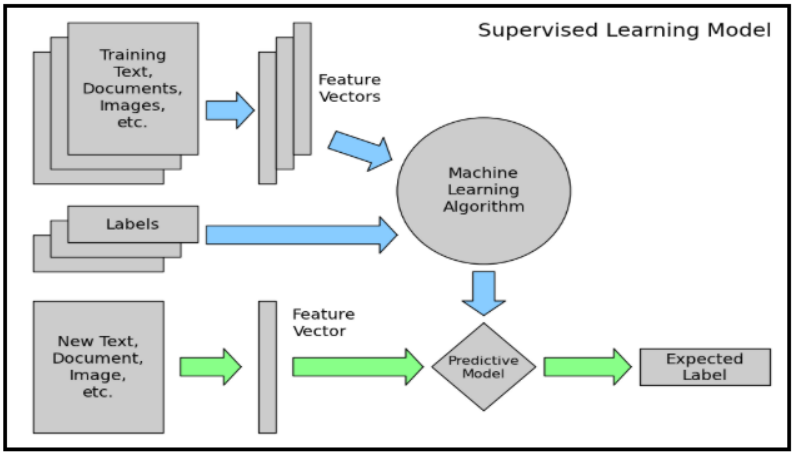
Training text: It is the input text through which our supervised learning model is able to learn and predict the required class.

Feature Vector: A feature vector is a vector that contains information describing the characteristics of the input data.

Labels: These are the predefined categories/classes that our model will predict

ML Algo: It is the algorithm through which our model is able to deal with text classification (In our case : CNN, RNN, HAN)

Predictive Model: A model which is trained on the historical dataset which can perform label predictions.

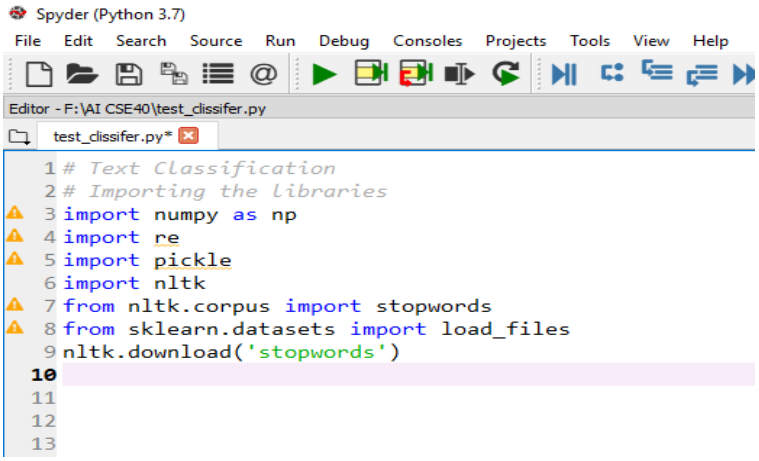


SETTING UP THE ENVIRONMENT

At first we need to setup some editors and libraries. Here we are using Anaconda navigator, Spyder 3.3.1 and python 3.7.0 programming language.

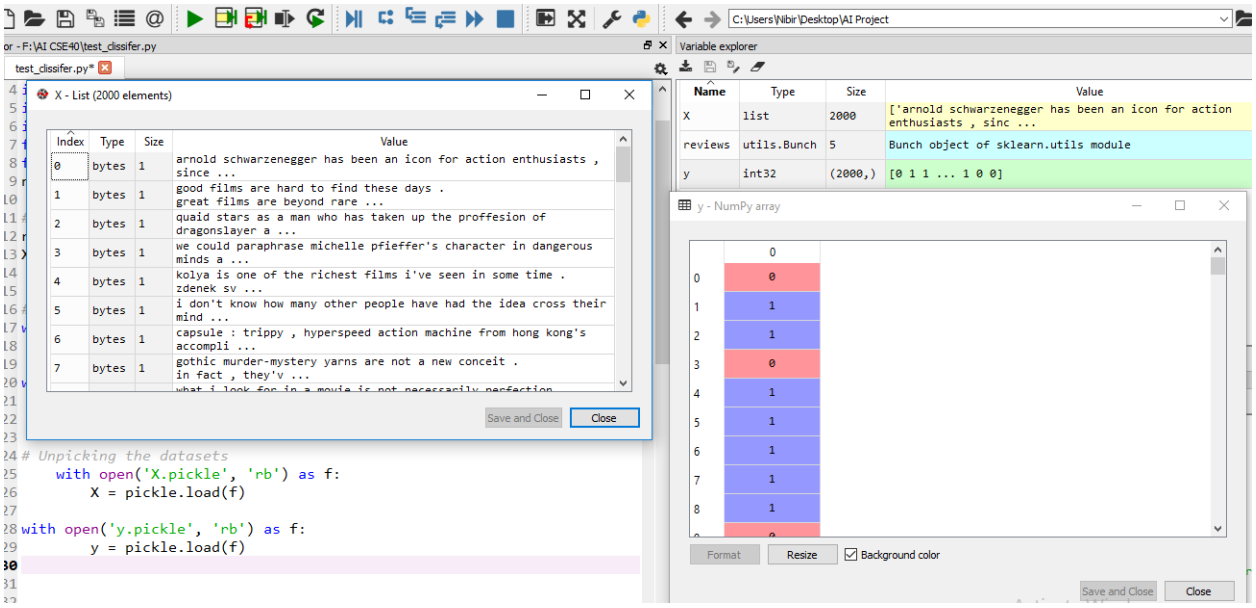
Also, little bit of python and ML basics including text classification is required. We will be using scikit-learn (python) libraries.

Install all the things that told at first of this section. Then open the spyder and there we import some libraries and download 'stopwords'



Here we download and import Sentiment polarity datasets (V2.0) which contents 5331 positive and 5331 negative processed sentences.

Then we write some necessary codes for our projects.



After completing some parts of our projects we can see the list of 2000 elements and other side the num py libraries array data.

TRAINING SETS

Training set and a Test set: To get an accurate measure of how well our model performs, we need to test it's ability to classify text using text that it didn't see during training. If we test it against the training data, it is like giving it an open book test where it can memorize the answers.

So we need to extract some of the strings from the training data set and keep them in separate test data file. Then we can test the trained model's performance with that held-back data to get a real-world measure of how well the model performs.

You can train a classifier using the fastText command line tool. You just call fasttext, pass in the supervised keyword to tell it train a supervised classification model, and then give it the training file and and an output name for the model:

fasttext supervised -input `fasttext_dataset_training.txt` -output reviews_model

It only took 3 minutes to train this model with 580 million words on laptop [4].

FUTURE SCOPE

Features resulting from count-based vectorization methods like TF-IDF have some disadvantages. For instance:

They don't account for word position and context (despite using N-grams, which is only a quick fix).

TF-IDF word vectors are usually very high dimensional (>1M features if using bi-grams).

They are not able to capture semantics.

For this reason, many applications today rely on word embeddings and neural networks, which together can achieve state-of-the-art results.

CONCLUSION

This paper presented an efficient technique for text classification. The existing techniques require more data for training as well as the computational time of these techniques is also high. In contrast to the existing algorithms, the proposed hybrid algorithm requires less training data and less computational time. In spite of the randomly chosen training set we achieved 90% accuracy for 50% training data. Though the experimental results are quite encouraging, it would be better if we work with larger data sets with more classes.

REFERRENCE

[1] Loper Edward, "NLTK Tutorial: Text Classification, 2004

[2] Agarwal R., Mannila H., Srikant R., Toivonan H., Verkamo, "A Fast Discovery of Association Rules," Advances in Knowledge Discovery and Data Mining, 1996.

[3] Chowdhury Mofizur Rahman, Ferdous Ahmed Sohel, Parvez Naushad, Kamruzzaman S. M,

"Text Classification Using the Concept of Association Rule of Data Mining," In Proceedings of International Conference on Information Technology, Nepal, 2003, pp 234-241.

[4] https://medium.com/@ageitgey/text-classification-is-your-new-secret-weapon-7ca4fad15788