Project Requirement and Specification

on

Wordnet Processing through NLP

(CSE V Semester Mini project)

2021-2022

Name: Aytijha Chakraborty

University Roll No: 1914009

Section: K

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

ACKNOWLEDGMENT

I would like to thank my Project Guide Dr. Rakesh Patra Sir for his patience, support, and encouragement throughout the completion of this project and for having faith in me.

Aytijha Chakraborty

Roll No.- 1914009

CSE-K-V-Sem

Session: 2021-2022

GEHU, Dehradun

ABSTRACT

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software.
The study of natural language processing has been around for more than 50 years and grew out of the field of linguistics with the rise of computers.

Linguistics is the scientific study of language, including its grammar, semantics, and phonetics. Classical linguistics involved devising and evaluating rules of language. Great progress was made on formal methods for syntax and semantics, but for the most part, the interesting problems in natural language understanding resist clean mathematical formalisms.
Broadly, a linguist is anyone who studies language, but perhaps more colloquially, a self-defining linguist may be more focused on being out in the field.

Mathematics is the tool of science. Mathematicians working on natural language may refer to their study as mathematical linguistics, focusing exclusively on the use of discrete mathematical formalisms and theory for natural language (e.g. formal languages and automata theory).

Computational linguistics is the modern study of linguistics using the tools of computer science. Computational linguistics also became known by the name of natural language process, or NLP, to reflect the more engineer-based or empirical approach of the statistical methods.
The statistical dominance of the field also often leads to NLP being described as Statistical Natural Language Processing, perhaps to distance it from the classical computational linguistics methods.

PROJECT INTRODUCTION AND MOTIVATION

What is Wordnet?

WordNet is a Corpus, a large lexical database of English Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet is freely and publicly available for download. Its structure makes it a useful tool for computational linguistics and Natural Language Processing.

About Project

This project (named Caramel Popcorn) is a small-scale implementation of a Movie Analysis System. This means that upon being provided the details of a movie, the probability of it making a huge success can be accurately predicted.

This project attempts to predict how successful a movie will be in the Box Office, even before it is actually released.

The pipeline of the project includes three components as of now:

1. Data Scraping [Modules Used: BeautifulSoup]
2. Feature Exploration and Engineering [Languages Used: Python]
3. NLP Classification [Modules Used: NLTK, MultinomialNB, SVC]
   [Corpus Used: WordNet]

METHODOLOGY

As mentioned above, the model expects the Name and the complete Description of the movie to be analyzed, and classifies the movie into one of over 200 specific genres.

The data used to train this model was scraped from IMDB's list of Top 1000 movies. This was done by parsing each webpage into an HTML document, and then extracting the useful bits of information using BeautifulSoup, a library offered by Python. The extracted data is then formatted into a Dataframe and then exported into a CSV file for further uses.

Once the required data is available in the form of a CSV file/Dataframe, it is then cleaned for our model to work with. This cleaning process is conducted in a 2-step process:

- First, any redundant/garbage column is dropped
- Next, all the Null values, Garbage values and Outliers are handled

The cleaned data is then separated into two parts, one for our model to work with, and the other being unrelated with this phase of the project.

Once we have the data required for our model, we need to preprocess it and extract the features before we move onto the next step, i.e. NLP Classification.

The data preprocessing is conducted in a few steps. First we obtain Clean_text from the provided textual columns. To do that:

- First, we add the contents of all the Textual Columns into one Column
- Next, for each row, we turn the whole string into lower case, strip any leading/trailing whitespaces, and substitute all Punctuation marks and Escape Sequence characters with a whitespace
- Next, we look for Stopwords, i.e. words that do not add much value, and remove them from the text
- Lastly, we Lemmatize all the words, i.e. change variant-forms into their root-words

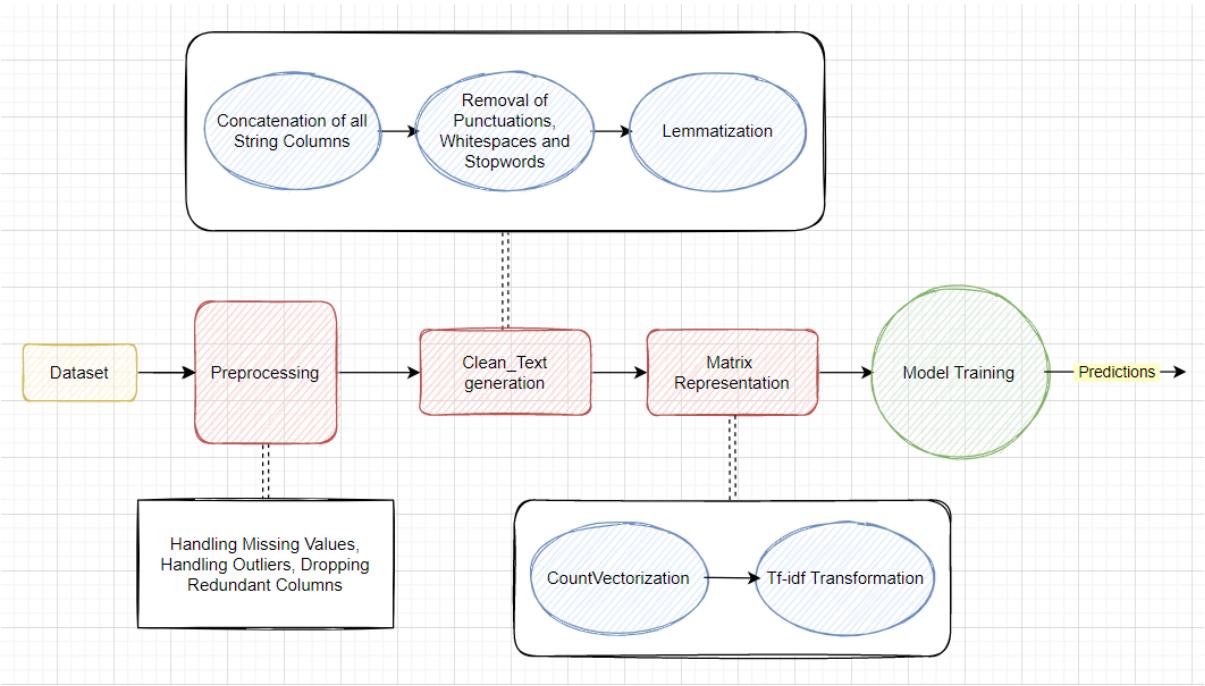
Now we have obtained a Clean_text from the provided textual columns. Next, we need to transform this string data into a numerical format. To do that:

- First, we use CountVectorizer() to convert the collection of string data into token counts
- And then, we TfidfTransformer() to convert this count matrix into a tf-idf (term-frequency times inverse document-frequency) representation.

This is done to scale down the impact of words that occur very frequently in the string collection/document.

Now we simply feed this tf-idf representation (in the form of a matrix) and the class-list, i.e. Genre List to our model for it to train.

FLOW OF PIPELINE



SCREENSHOTS

Preprocessing Section:

```
[7]  # converting text to lowercase, stripping and removing punctuations
     def preprocess(text):
         text = text.lower()
         text=text.strip()
         text=re.compile('<.*?>').sub('', text)
         text = re.compile('[%s]' % re.escape(string.punctuation)).sub(' ', text)
         text = re.sub('\s+', ' ', text)
         text = re.sub(r'\[[0-9]*\]',' ',text)
         text=re.sub(r'[^\w\s]', '', str(text).lower().strip())
         text = re.sub(r'\d',' ',text)
         text = re.sub(r'\s+',' ',text)
         return text
```

```
[8]  # removing stopwords
     def stopword(string):
         a= [i for i in string.split() if i not in stopwords.words('english')]
         return ' '.join(a)
```

```
[9]  # lemmatization
     wl = WordNetLemmatizer()

     # helper function to map NTLK position tags
     def get_wordnet_pos(tag):
         if tag.startswith('J'):
             return wordnet.ADJ
         elif tag.startswith('V'):
             return wordnet.VERB
         elif tag.startswith('N'):
             return wordnet.NOUN
         elif tag.startswith('R'):
             return wordnet.ADV
         else:
             return wordnet.NOUN

     # to tokenize the sentence
     def lemmatizer(string):
         word_pos_tags = nltk.pos_tag(word_tokenize(string)) # Get position tags
         a=[wl.lemmatize(tag[0], get_wordnet_pos(tag[1])) for idx, tag in enumerate(word_pos_tags)]
         return " ".join(a)
```

Dataset fed to the model:

	Name of movie	Description	Genre	Text	clean_text
0	The Shawshank Redemption	Two imprisoned men bond over a number of years...	Drama	The Shawshank Redemption Two imprisoned men bo...	shawshank redemption two imprisoned men bond n...
1	The Godfather	An organized crime dynasty's aging patriarch t...	Crime Drama	The Godfather An organized crime dynasty's agi...	godfather organize crime dynasty age patriarch...
2	Soorarai Pottru	Nedumaaran Rajangam "Maara" sets out to make t...	Drama	Soorarai Pottru Nedumaaran Rajangam "Maara" se...	soorarai pottru nedumaaran rajangam maara set ...
3	The Dark Knight	When the menace known as the Joker wreaks havo...	Action Crime Drama	The Dark Knight When the menace known as the J...	dark knight menace know joker wreaks havoc cha...
4	The Godfather: Part II	The early life and career of Vito Corleone in ...	Crime Drama	The Godfather: Part II The early life and care...	godfather part ii early life career vito corle...
...	...	...	...	...	...
995	Giant	Sprawling epic covering the life of a Texas ca...	Drama Western	Giant Sprawling epic covering the life of a Te...	giant sprawl epic cover life texas cattle ranc...
996	From Here to Eternity	In Hawaii in 1941, a private is cruelly punish...	Drama Romance War	From Here to Eternity In Hawaii in 1941, a pri...	eternity hawaii private cruelly punish box uni...
997	Gilda	A small-time gambler hired to work in a Buenos...	Drama Film-Noir Romance	Gilda A small-time gambler hired to work in a ...	gilda small time gambler hire work buenos air ...
998	Lifeboat	Several survivors of a torpedoed merchant ship...	Drama War	Lifeboat Several survivors of a torpedoed merc...	lifeboat several survivor torpedo merchant shi...
999	The 39 Steps	A man in London tries to help a counter-espion...	Crime Mystery Thriller	The 39 Steps A man in London tries to help a c...	step man london try help counter espionage age...

1000 rows × 5 columns

Models tested, along with their accuracies:

```
[46] clf = MultinomialNB().fit(train_x, train_y)
     y_score = clf.predict(train_x)
```

```
[47] n_right = 0
     for i in range(len(y_score)):
         if y_score[i] == train_y[i]:
             n_right += 1

     print("Accuracy: %.2f%%" % ((n_right/float(len(train_y)) * 100)))
```

```
     Accuracy: 29.83%
```

```
[48] from sklearn.svm import SVC
     clf1 = SVC().fit(train_x, train_y)
```

```
[ ] ?SVC()
```

```
[49] y_score = clf1.predict(train_x)
     n_right = 0
     for i in range(len(y_score)):
         if y_score[i] == train_y[i]:
             n_right += 1

     print("Accuracy: %.2f%%" % ((n_right/float(len(train_y)) * 100)))
```

```
     Accuracy: 81.26%
```

REFERENCES

1. **Python Official Site: https://www.python.org**
2. **Elaboration on NLP: https://machinelearningmastery.com/natural-language-processing/**
3. **Wordnet Official Site: https://wordnet.princeton.edu/**
4. **Sklearn Documentation: https://scikit-learn.org/stable/index.html**
5. **IMDB for scraping training Data: https://www.imdb.com/**