Categorization of Web Pages using Text Classification Methods

The Echos
Anandan Sundar
Siddarth Udayakumar

PROBLEM DESCRIPTION

The internet has a lot of web pages and these web pages are made up of a lot of different content that it is not easy to categorize them under different topics. To do this, a person has to manually index and categorize the web page based on the topic that the page is dealing about. The problem is with the current size of the web, it has become cumbersome to try and manually index and categorize its content. For example, a page on tourism in America can be broadly classified under the topic “travel”. But, to label a page, the user has to read the contents and decide the label later on.

PROPOSED SOLUTION

Our project aims at providing a much easier and faster way of categorizing the web pages based on the content available. This would allow a user to get the correct categorization of a page without having to manually find it. To get much better results, we started out with Wikipedia pages on different topics. We created a manually tagged dataset that broadly classifies a feature word into one of the four categories we have – politics, business, technology or travel.

We will be initially going with a simple baseline system for the classification and later follow it up with a more improved strategy that makes use of Natural Language Processing features in lexical, syntactic and semantic terms.

PROGRAMMING TOOLS

We used the following programming tools and APIs to work on this project

- Python.
- Pycharm IDE
- Natural Language Toolkit (NLTK)
- TextBlob library for Python.
- Wikipedia API for Python.

IMPLEMENTATION DETAILS

BASELINE SYSTEM

The simple baseline system uses the corpus obtained from Wikipedia to classify the category of the feature word based on the frequency of topic terms occurring in the tagged dataset. If the term with the highest frequency matches with the tagged dataset we have, the feature word is classified correctly else it will be classified with the word with the highest frequency of occurrence.

For example, if the feature word we want to categorize is “New York City”, it is supposed to be classified as “travel” from the tagged dataset we have. The frequency for other tagged topics from the dataset is also calculated and the tag which has the highest frequency of occurrence in the corpus will be the final classified topic for our feature word.

BASELINE OUTPUT

Based on the available dataset we have, we passed a set of feature words to be classified by the baseline system. The output is as follows:

```
/usr/bin/python3.5
/home/anandan/PycharmProjects/NlpProject/baseline.py

['donald trump', ' politics'] Actual Class : politics Obtained
Class : business ['texas', ' travel'] Actual Class : travel Obtained
Class : travel ['apple', ' technology'] Actual Class : technology
Obtained Class : travel ['android (Operating System)', ' technology']
Actual Class : technology Obtained Class : technology ['iphone', '
technology'] Actual Class : technology Obtained Class : technology
['adobe photoshop', ' technology'] Actual Class : technology Obtained
Class : technology ['television', ' technology'] Actual Class :
technology Obtained Class : technology ['camera', ' technology']
Actual Class : technology Obtained Class : technology ['google', '
business'] Actual Class : business Obtained Class : technology
['toyota', ' business'] Actual Class : business Obtained Class :
technology ['laptop', ' technology'] Actual Class : technology
Obtained Class : technology ['steve jobs', ' business'] Actual
Class : business Obtained Class : business ['barack obama', '
politics'] Actual Class : politics Obtained Class : business ['new
york city', ' travel'] Actual Class : travel Obtained Class :
technology ['bernie sanders', ' politics'] Actual Class : politics
Obtained Class : politics ['Canon Inc.', ' business'] Actual Class :
business Obtained Class : business ['bmw', ' business'] Actual
Class : business Obtained Class : technology ['audi', ' business']
Actual Class : business Obtained Class : technology ['chennai', '
travel'] Actual Class : travel Obtained Class : technology ['paris',
' travel'] Actual Class : travel Obtained Class : business ['dallas',
' travel'] Actual Class : travel Obtained Class : business
['titicaca', ' travel'] Actual Class : travel Obtained Class : travel
['nile', ' travel'] Actual Class : travel Obtained Class : travel
['danube', ' travel'] Actual Class : travel Obtained Class : travel
['seine', ' travel'] Actual Class : travel Obtained Class : business
['oslo', ' travel'] Actual Class : travel Obtained Class : travel
['spotify', ' business'] Actual Class : business Obtained Class :
business ['china', ' travel'] Actual Class : travel Obtained Class :
technology ['japan', ' travel'] Actual Class : travel Obtained
Class : technology ['new zealand', ' travel'] Actual Class : travel
Obtained Class : travel ['volkswagen', ' business'] Actual Class :
business Obtained Class : business ['ibm', ' business'] Actual
Class : business Obtained Class : business ['singapore', ' travel']
Actual Class : travel Obtained Class : business ['microsoft', '
business'] Actual Class : business Obtained Class : business
['zillow', ' business'] Actual Class : business Obtained Class :
business']

Accuracy of baseline: 49.645390070921984%
```

IMPROVEMENT STRATEGY

Once the baseline system has been completed, we added further improvements to the system by performing additional operations by adding lexical, syntactic and semantic features to the corpuses associated with each feature word in the test set. We added each feature step by step and allowed them to finally work in sequence to produce the desired result.

EXAMPLE

- For our example how a word is going to be processed by the program by passing a sample dataset. Let us take the example "Morgan Stanley".
- The Wikipedia API provides us the required corpus of text for processing.

```
Topic : Morgan Stanley
Actual Class : business
Data : ['morgan stanley', ' business']
```

LEXICAL FEATURES

We used the following two lexical processes by making use of the functions in the Natural Language Toolkit

Tokenization

Tokenization is the method of breaking text into words, phrases and other meaningful elements. Tokenization of the text allows us to use the text for further processing and parsing.

We tokenized the corpus obtained from Wikipedia for the specific feature word. Tokenization was performed using the NLTK tokenize package (nltk.mwetokenize package). We also used the MWE Tokenizer which allows for merging multiword expressions into single tokens. We also used Stopword removal here to get much concise text data.

```
After tokenizing and removing stopwords :

['morgan_stanley', 'nyse', 'american', 'multinational', 'financial',
'services', 'corporation', 'headquartered', 'morgan_stanley',
'building', 'midtown', 'manhattan', 'new', 'york', 'city',
'morgan_stanley', 'operates', 'countries', '1300', 'offices',
'60,000', 'employees', 'according', 'scorpio', 'partnership',
'global', 'private', 'banking', 'benchmark', 'company', '1.454',
'trillion', 'assets', 'management', 'aum', '2014', 'increase', '17.5',
'2013', 'figure', 'corporation', 'formed', 'j.p.', 'morgan', 'co.',
'partners', 'henry', 'morgan', 'grandson', 'j.p.', 'morgan', 'harold',
'stanley', 'others', 'came', 'existence', 'september', '1935',
'response', 'glass-steagall', 'act', 'required', 'splitting',
'commercial', 'investment', 'banking', 'businesses', 'first', 'year',
'company', 'operated', 'market', 'share', '1.1', 'billion', 'public',
'offerings', 'private', 'placements', 'main', 'areas', 'business',
```

```
'firm', 'today', 'global', 'wealth', 'management', 'institutional',
'securities', 'investment', 'management']
```

Lemmatization

Lemmatization is the process of grouping together the different forms of the same word so they can be analyzed as a single element and identified by dictionaries. This process was necessary since Wikipedia used different tenses and forms of the word in the corpus.

For lemmatization, we used the lemmatizer from the TextBlob Python library for textual processing. The lemmatize method in TextBlob was inherited from the NLTK Word Lemmatizer. For example, the output will be as follows

```
After Lemmatization :

['morgan_stanley', 'nyse', 'american', 'multinational', 'financial',
'service', 'corporation', 'headquartered', 'morgan_stanley',
'building', 'midtown', 'manhattan', 'new', 'york', 'city',
'morgan_stanley', 'operates', 'country', '1300', 'office', '60,000',
'employee', 'according', 'scorpio', 'partnership', 'global',
'private', 'banking', 'benchmark', 'company', '1.454', 'trillion',
'asset', 'management', 'aum', '2014', 'increase', '17.5', '2013',
'figure', 'corporation', 'formed', 'j.p.', 'morgan', 'co.',
'partner', 'henry', 'morgan', 'grandson', 'j.p.', 'morgan', 'harold',
'stanley', 'others', 'came', 'existence', 'september', '1935',
'response', 'glass-steagall', 'act', 'required', 'splitting',
'commercial', 'investment', 'banking', 'business', 'first', 'year',
'company', 'operated', 'market', 'share', '1.1', 'billion', 'public',
'offering', 'private', 'placement', 'main', 'area', 'business',
'firm', 'today', 'global', 'wealth', 'management', 'institutional',
'security', 'investment', 'management']
```

SYNTACTIC FEATURES

We went with POS Tagging for the words and shallow parsing for obtaining the syntactic features for the words. We employed POS tagging and shallow parsing since the only parts of speech in the corpus we are interested in are the nouns.

We used the NLTK library's POS Tagger and Shallow Parser. For the Shallow Parser, we used specific grammar rules that allows the program to choose only parts of the text that confirms to the given grammar rules.

We use the RegExParser on NLTK to produce shallow parsed output.

POS Tagging

```
After POS Tagging

[('morgan_stanley', 'NN'), ('nyse', 'JJ'), ('american', 'JJ'),
('multinational', 'JJ'), ('financial', 'JJ'), ('service', 'NN'),
```

```
('corporation', 'NN'), ('headquartered', 'VBD'), ('morgan_stanley',
'NN'), ('building', 'NN'), ('midtown', 'JJ'), ('manhattan', 'JJ'),
('new', 'JJ'), ('york', 'NN'), ('city', 'NN'), ('morgan_stanley',
'NN'), ('operates', 'VBZ'), ('country', 'NN'), ('1300', 'CD'),
('office', 'NN'), ('60,000', 'CD'), ('employee', 'NN'), ('according',
'VBG'), ('scorpio', 'JJ'), ('partnership', 'NN'), ('global', 'JJ'),
('private', 'JJ'), ('banking', 'NN'), ('benchmark', 'NN'),
('company', 'NN'), ('1.454', 'CD'), ('trillion', 'CD'), ('asset',
'NN'), ('management', 'NN'), ('aum', 'NN'), ('2014', 'CD'),
('increase', 'NN'), ('17.5', 'CD'), ('2013', 'CD'), ('figure', 'NN'),
('corporation', 'NN'), ('formed', 'VBD'), ('j.p.', 'NNP'), ('morgan',
'NNP'), ('co.', 'VBD'), ('partner', 'NN'), ('henry', 'NN'),
('morgan', 'NNP'), ('grandson', 'NN'), ('j.p.', 'NN'), ('morgan',
'NN'), ('harold', 'VBD'), ('stanley', 'JJ'), ('others', 'NNS'),
('came', 'VBD'), ('existence', 'RB'), ('september', 'JJ'), ('1935',
'CD'), ('response', 'NN'), ('glass-steagall', 'JJ'), ('act', 'NN'),
('required', 'VBN'), ('splitting', 'VBG'), ('commercial', 'JJ'),
('investment', 'NN'), ('banking', 'NN'), ('business', 'NN'),
('first', 'JJ'), ('year', 'NN'), ('company', 'NN'), ('operated',
'VBD'), ('market', 'NN'), ('share', 'NN'), ('1.1', 'CD'), ('billion',
'CD'), ('public', 'JJ'), ('offering', 'VBG'), ('private', 'JJ'),
('placement', 'NN'), ('main', 'JJ'), ('area', 'NN'), ('business',
'NN'), ('firm', 'NN'), ('today', 'NN'), ('global', 'JJ'), ('wealth',
'NN'), ('management', 'NN'), ('institutional', 'JJ'), ('security',
'NN'), ('investment', 'NN'), ('management', 'NN')]
```

Shallow Parsing

```
>> grammar = """ NP: {<DT>?<JJ>*<NN>}
                      {<NNP>+}
                      {<NN><NN>}
                      {<NNS><VBP>}
                      {<V.*> <TO> <V.*>}
                      {<N.*>(4,)} """
>> NPChunker = RegexpParser(grammar)
>> chunked_result = NPChunker.parse(pos_tagged_word_list)
>> print(chunked_result)
```

After Chunking (Shallow Parsing) :

```
['morgan_stanley', 'service', 'corporation', 'morgan_stanley',
'building', 'york', 'city', 'morgan_stanley', 'country', 'office',
'employee', 'partnership', 'banking', 'benchmark', 'company',
'asset', 'management', 'aum', 'increase', 'figure', 'corporation',
'j.p.', 'morgan', 'partner', 'henry', 'morgan', 'grandson', 'j.p.',
'morgan', 'response', 'act', 'investment', 'banking', 'business',
'year', 'company', 'market', 'share', 'placement', 'area',
'business', 'firm', 'today', 'wealth', 'management', 'security',
'investment', 'management']
```

SEMANTIC FEATURES

We chose using hypernyms and meronyms for getting the semantic features from the processed text.

We made use of the NLTK Wordnet API for semantic features. Wordnet is the lexical database for the English language. We used the hypernyms and Meronyms from the NLTK Wordnet features.

Hypernyms

Hypernym refers to a word which has a broad meaning and may contain more specific words that fall under it. For example, the word Technology is a hypernym for any technology related term including Bluetooth and Wi-FI.

```
Topic has been found in hypernym! business

After Hypernym matching :
['morgan_stanley', 'service', 'corporation', 'morgan_stanley',
'building', 'york', 'city', 'morgan_stanley', 'country', 'office',
'employee', 'partnership', 'banking', 'benchmark', 'company', 'asset',
'management', 'aum', 'increase', 'figure', 'corporation', 'j.p.',
'morgan', 'partner', 'henry', 'morgan', 'grandson', 'j.p.', 'morgan',
'response', 'act', 'investment', 'banking', 'business', 'business',
'business', 'business', 'business', 'business', 'business',
'business', 'business', 'business', 'year', 'company', 'market',
'share', 'placement', 'area', 'business', 'business', 'business',
'business', 'business', 'business', 'business', 'business',
'business', 'business', 'firm', 'today', 'wealth', 'management',
'security', 'investment', 'management']
```

Meronyms

Meronym refers to a word that is used to denote a thing that is a part of something else. For example, wheels is the meronym for the word automobile.

```
Topic has been found in meronym! business

After Meronym matching :

['morgan_stanley', 'service', 'corporation', 'morgan_stanley',
'building', 'york', 'city', 'morgan_stanley', 'country', 'office',
'employee', 'partnership', 'banking', 'benchmark', 'company', 'asset',
'management', 'aum', 'increase', 'figure', 'corporation', 'j.p.',
'morgan', 'partner', 'henry', 'morgan', 'grandson', 'j.p.', 'morgan',
'response', 'act', 'investment', 'banking', 'business', 'business',
'business', 'business', 'business', 'business', 'business',
'business', 'business', 'business', 'business', 'business',
'business', 'business', 'business', 'business', 'business',
'business', 'business', 'business', 'year', 'company', 'market',
'share', 'placement', 'area', 'business', 'business', 'business',
'business', 'business', 'business', 'business', 'business',
'business', 'business', 'business', 'business', 'business',
'business', 'business', 'business', 'business', 'business',
```

```
'business', 'business', 'firm', 'today', 'wealth', 'management',
'security', 'investment', 'management']
```

MODEL CREATION

The output from the above processes, specifically the meronym features gives us a list of words which are closely related to the topics we have to classify. This is given as input to the Naïve Bayes Classifier and is used to create a training model. The model is further used to test the reasonably large test dataset and classify the results there. We stored the model as a pickle file so that it can be used later for faster testing of datasets.

FINAL CLASSIFICATION OUTPUT

```
Topic : morgan stanley
Actual : business
Classified :  business
```

RESULTS

BASELINE OUTPUT

```
/usr/bin/python3.5
/home/siddarth/PycharmProjects/NlpProject_Submission/baseline.py
['donald trump', ' politics']
Actual Class :  politics
Obtained Class : business
['texas', ' travel']
Actual Class :  travel
Obtained Class : travel
['apple', ' technology']
Actual Class :  technology
Obtained Class : travel
['android (Operating System)', ' technology']
Actual Class :  technology
Obtained Class : technology
['iphone', ' technology']
Actual Class :  technology
Obtained Class : technology
['adobe photoshop', ' technology']
Actual Class :  technology
Obtained Class : technology
['television', ' technology']
Actual Class :  technology
Obtained Class : technology
['camera', ' technology']
Actual Class :  technology
Obtained Class : technology
['google', ' business']
Actual Class :  business
Obtained Class : technology
['toyota', ' business']
```

```
Actual Class :  business
Obtained Class : technology
['laptop', ' technology']
Actual Class :  technology
Obtained Class : technology
['steve jobs', ' business']
Actual Class :  business
Obtained Class : business
['barack obama', ' politics']
Actual Class :  politics
Obtained Class : business
['new york city', ' travel']
Actual Class :  travel
Obtained Class : technology
['bernie sanders', ' politics']
Actual Class :  politics
Obtained Class : politics
['Canon Inc.', ' business']
Actual Class :  business
Obtained Class : business
['bmw', ' business']
Actual Class :  business
Obtained Class : technology
['audi', ' business']
Actual Class :  business
Obtained Class : technology
['chennai', ' travel']
Actual Class :  travel
Obtained Class : technology
['paris', ' travel']
Actual Class :  travel
Obtained Class : business
['dallas', ' travel']
Actual Class :  travel
Obtained Class : business
['titicaca', ' travel']
Actual Class :  travel
Obtained Class : travel
['nile', ' travel']
Actual Class :  travel
Obtained Class : travel
['danube', ' travel']
Actual Class :  travel
Obtained Class : travel
['seine', ' travel']
Actual Class :  travel
Obtained Class : business
['oslo', ' travel']
Actual Class :  travel
Obtained Class : travel
['spotify', ' business']
Actual Class :  business
Obtained Class : business
['china', ' travel']
Actual Class :  travel
Obtained Class : technology
['japan', ' travel']
```

```
Actual Class :  travel
Obtained Class : technology
['new zealand', ' travel']
Actual Class :  travel
Obtained Class : travel
['volkswagen', ' business']
Actual Class :  business
Obtained Class : business
['ibm', ' business']
Actual Class :  business
Obtained Class : business
['singapore', ' travel']
Actual Class :  travel
Obtained Class : business
['microsoft', ' business']
Actual Class :  business
Obtained Class : business
['zillow', ' business']
Actual Class :  business
Obtained Class : business
['san francisco', ' travel']
Actual Class :  travel
Obtained Class : business
['sacramento', ' travel']
Actual Class :  travel
Obtained Class : business
['las vegas', ' travel']
Actual Class :  travel
Obtained Class : business
['rockies', ' travel']
Actual Class :  travel
Obtained Class : travel
['david cameron', ' politics']
Actual Class :  politics
Obtained Class : politics
['brexit', ' politics']
Actual Class :  politics
Obtained Class : business
['jayalalithaa', ' politics']
Actual Class :  politics
Obtained Class : politics
['narendra modi', ' politics']
Actual Class :  politics
Obtained Class : business
['hillary clinton', ' politics']
Actual Class :  politics
Obtained Class : politics
['python (programming language)', ' technology']
Actual Class :  technology
Obtained Class : business
['java language', ' technology']
Actual Class :  technology
Obtained Class : technology
['bill gates', ' business']
Actual Class :  business
Obtained Class : business
['warren buffet', ' business']
```

```
Actual Class :  business
Obtained Class : business
['nasdaq', ' business']
Actual Class :  business
Obtained Class : business
['korea', ' travel']
Actual Class :  travel
Obtained Class : technology
['tim cook', ' business']
Actual Class :  business
Obtained Class : business
['flipkart', ' business']
Actual Class :  business
Obtained Class : business
['uber (company)', ' business']
Actual Class :  business
Obtained Class : business
['africa', ' travel']
Actual Class :  travel
Obtained Class : politics
['usa', ' travel']
Actual Class :  travel
Obtained Class : business
['russia', ' travel']
Actual Class :  travel
Obtained Class : technology
['nexus 6p', ' technology']
Actual Class :  technology
Obtained Class : business
['chicago', ' travel']
Actual Class :  travel
Obtained Class : business
['cincinatti', ' travel']
Actual Class :  travel
Obtained Class : business
['sacramento', ' travel']
Actual Class :  travel
Obtained Class : business
['sony', ' technology']
Actual Class :  technology
Obtained Class : business
['samsung', ' technology']
Actual Class :  technology
Obtained Class : business
['laptop', ' technology']
Actual Class :  technology
Obtained Class : technology
['macintosh', ' technology']
Actual Class :  technology
Obtained Class : technology
['Near field communication', ' technology']
Actual Class :  technology
Obtained Class : technology
['bluetooth', ' technology']
Actual Class :  technology
Obtained Class : technology
['lyft', ' business']
```

```
Actual Class :  business
Obtained Class : business
['mac os', ' technology']
Actual Class :  technology
Obtained Class : technology
['wi-fi', ' technology']
Actual Class :  technology
Obtained Class : technology
['las vegas', ' travel']
Actual Class :  travel
Obtained Class : business
['abraham lincoln', ' politics']
Actual Class :  politics
Obtained Class : business
['nigel farage', ' politics']
Actual Class :  politics
Obtained Class : politics
['headphones', ' technology']
Actual Class :  technology
Obtained Class : technology
['television', ' technology']
Actual Class :  technology
Obtained Class : technology
['radio', ' technology']
Actual Class :  technology
Obtained Class : travel
['benjamin franklin', ' politics']
Actual Class :  politics
Obtained Class : politics
['franklin roosevelt', ' politics']
Actual Class :  politics
Obtained Class : business
['manmohan singh', ' politics']
Actual Class :  politics
Obtained Class : technology
['boris johnson', ' politics']
Actual Class :  politics
Obtained Class : politics
['michael bloomberg', ' politics']
Actual Class :  politics
Obtained Class : business
['ubuntu (operating system)', ' technology']
Actual Class :  technology
Obtained Class : business
['nigeria', ' travel']
Actual Class :  travel
Obtained Class : technology
['ruby on rails', ' technology']
Actual Class :  technology
Obtained Class : business
['barbados', ' travel']
Actual Class :  travel
Obtained Class : business
['visual c', ' technology']
Actual Class :  technology
Obtained Class : business
['amazon.com', ' business']
```

```
Actual Class :  business
Obtained Class : business
['jawaharlal nehru', ' politics']
Actual Class :  politics
Obtained Class : politics
['mahatma gandhi', ' politics']
Actual Class :  politics
Obtained Class : politics
['george washington', ' politics']
Actual Class :  politics
Obtained Class : business
['al gore', ' politics']
Actual Class :  politics
Obtained Class : technology
['taiwan', ' travel']
Actual Class :  travel
Obtained Class : business
['mexico', ' travel']
Actual Class :  travel
Obtained Class : business
['costa rica', ' travel']
Actual Class :  travel
Obtained Class : business
['twitter', ' business']
Actual Class :  business
Obtained Class : business
['morgan stanley', ' business']
Actual Class :  business
Obtained Class : business
['stock', ' business']
Actual Class :  business
Obtained Class : business
['share (finance)', ' business']
Actual Class :  business
Obtained Class : business
['congo basin', ' travel']
Actual Class :  travel
Obtained Class : business
['venezula', ' travel']
Actual Class :  travel
Obtained Class : travel
['mexican stock exchange', ' business']
Actual Class :  business
Obtained Class : business
['office', ' business']
Actual Class :  business
Obtained Class : business
['mongolia', ' travel']
Actual Class :  travel
Obtained Class : business
['hilton worldwide', ' business']
Actual Class :  business
Obtained Class : travel
['david koch', ' politics']
Actual Class :  politics
Obtained Class : technology
['rick scott', ' politics']
```

```
Actual Class :  politics
Obtained Class : business
['wells fargo', ' business']
Actual Class :  business
Obtained Class : business
['chase bank', ' business']
Actual Class :  business
Obtained Class : business
['siberia', ' travel']
Actual Class :  travel
Obtained Class : business
['shimla', ' travel']
Actual Class :  travel
Obtained Class : business
['radio-frequency identification', ' technology']
Actual Class :  technology
Obtained Class : technology
['infrared', ' technology']
Actual Class :  technology
Obtained Class : technology
['x-ray', ' technology']
Actual Class :  technology
Obtained Class : technology
['pebble (watch)', ' business']
Actual Class :  business
Obtained Class : technology
['Android Nougat', ' technology']
Actual Class :  technology
Obtained Class : business
['taipei', ' travel']
Actual Class :  travel
Obtained Class : business
['lte (telecommunication)', ' technology']
Actual Class :  technology
Obtained Class : technology
['wikipedia', ' technology']
Actual Class :  technology
Obtained Class : technology
['mike pence', ' politics']
Actual Class :  politics
Obtained Class : business
['jimmy carter', ' politics']
Actual Class :  politics
Obtained Class : business
['seagate technology', ' business']
Actual Class :  business
Obtained Class : technology
['safeway inc', ' business']
Actual Class :  business
Obtained Class : business
['periscope (app)', ' technology']
Actual Class :  technology
Obtained Class : travel
['meerkat (app)', ' technology']
Actual Class :  technology
Obtained Class : business
['facebook', ' business']
```

```
Actual Class :  business
Obtained Class : technology
['whatsapp', ' business']
Actual Class :  business
Obtained Class : business
['skype', ' business']
Actual Class :  business
Obtained Class : business
['starbucks', ' business']
Actual Class :  business
Obtained Class : business
['malaysia', ' travel']
Actual Class :  travel
Obtained Class : business
['java', ' travel']
Actual Class :  travel
Obtained Class : business
['bali', ' travel']
Actual Class :  travel
Obtained Class : travel
['bill clinton', ' politics']
Actual Class :  politics
Obtained Class : travel
['amsterdam', ' travel']
Actual Class :  travel
Obtained Class : business
['instagram', ' technology']
Actual Class :  technology
Obtained Class : business
['motorola', ' business']
Actual Class :  business
Obtained Class : business
['lg corporation', ' business']
Actual Class :  business
Obtained Class : business
['laser', ' technology']
Actual Class :  technology
Obtained Class : technology
['3d printing', ' technology']
Actual Class :  technology
Obtained Class : technology
['shashi tharoor', ' politics']
Actual Class :  politics
Obtained Class : travel
['kerala', ' travel']
Actual Class :  travel
Obtained Class : travel
['chennai', ' travel']
Actual Class :  travel
Obtained Class : technology
['bangalore', ' travel']
Actual Class :  travel
Obtained Class : technology
Accuracy of baseline : 49.645390070921984%

Process finished with exit code 0
```

FINAL CLASSIFICATION OUTPUT

```
Topic : nigeria
Actual : travel
Classified :  travel
Topic : ruby on rails
Actual : technology
Classified :  technology
Topic : barbados
Actual : travel
Classified :  travel
Topic : visual c
Actual : technology
Classified :  technology
Topic : amazon.com
Actual : business
Classified :  business
Topic : jawaharlal nehru
Actual : politics
Classified :  politics
Topic : mahatma gandhi
Actual : politics
Classified :  travel
Topic : george washington
Actual : politics
Classified :  politics
Topic : al gore
Actual : politics
Classified :  politics
Topic : taiwan
Actual : travel
Classified :  travel
Topic : mexico
Actual : travel
Classified :  travel
Topic : costa rica
Actual : travel
Classified :  travel
Topic : twitter
Actual : business
Classified :  technology
Topic : morgan stanley
Actual : business
Classified :  business
Topic : stock
Actual : business
Classified :  business
Topic : share (finance)
```

```
Actual : business
Classified :  business
Topic : congo basin
Actual : travel
Classified :  travel
Topic : venezula
Actual : travel
Classified :  travel
Topic : mexican stock exchange
Actual : business
Classified :  travel
Topic : office
Actual : business
Classified :  technology
Topic : mongolia
Actual : travel
Classified :  travel
Topic : hilton worldwide
Actual : business
Classified :  business
Topic : david koch
Actual : politics
Classified :  business
Topic : rick scott
Actual : politics
Classified :  politics
Topic : wells fargo
Actual : business
Classified :  business
Topic : chase bank
Actual : business
Classified :  business
Topic : siberia
Actual : travel
Classified :  travel
Topic : shimla
Actual : travel
Classified :  travel
Topic : radio-frequency identification
Actual : technology
Classified :  technology
Topic : infrared
Actual : technology
Classified :  technology
Topic : x-ray
Actual : technology
Classified :  technology
Topic : pebble (watch)
Actual : business
Classified :  technology
Topic : Android Nougat
```

```
Actual : technology
Classified :  technology
Topic : taipei
Actual : travel
Classified :  travel
Topic : lte (telecommunication)
Actual : technology
Classified :  technology
Topic : wikipedia
Actual : technology
Classified :  business
Topic : mike pence
Actual : politics
Classified :  politics
Topic : jimmy carter
Actual : politics
Classified :  politics
Topic : seagate technology
Actual : business
Classified :  business
Topic : safeway inc
Actual : business
Classified :  business
Topic : periscope (app)
Actual : technology
Classified :  technology
Topic : meerkat (app)
Actual : technology
Classified :  technology
Topic : facebook
Actual : business
Classified :  business
Topic : whatsapp
Actual : business
Classified :  technology
Topic : skype
Actual : business
Classified :  technology
Topic : starbucks
Actual : business
Classified :  business
Topic : malaysia
Actual : travel
Classified :  travel
Topic : java
Actual : travel
Classified :  travel
Topic : bali
Actual : travel
Classified :  travel
Topic : bill clinton
```

```
Actual : politics
Classified :  politics
Topic : amsterdam
Actual : travel
Classified :  travel
Topic : instagram
Actual : technology
Classified :  technology
Topic : motorola
Actual : business
Classified :  technology
Topic : lg corporation
Actual : business
Classified :  technology
Topic : laser
Actual : technology
Classified :  technology
Topic : 3d printing
Actual : technology
Classified :  technology
Topic : shashi tharoor
Actual : politics
Classified :  travel
Topic : kerala
Actual : travel
Classified :  travel
Topic : chennai
Actual : travel
Classified :  travel
Topic : bangalore
Actual : travel
Classified :  travel

Accuracy : 80.0
```

ISSUES ENCOUNTERED

- Initially we tried process the words into a set in which we performed all the text processing activities. In this case, a set cannot contain duplicate values which hinders the efficiency of the classifier.
- Since training set is relatively huge, creation of a model took a long time. Since the words are stored in the list, the size of list and occupied more size. So, even in case of a small change, there was significant time taken to create the model.
- As per the original objective of this project, we wanted to categorize different webpages into topics but since most of the webpages we obtained are inconsistent and hence we chose to work with Wikipedia for this demonstration since it provided more consistent data.

- We used BeautifulSoup initially but owing to the inconsistency in webpages, we used Wikipedia API to get consistent, workable data.

PENDING ISSUES

- We have restricted ourselves to four broad topic categories to make it easier for classification since the model building took a lot of time. We require more topics for better categorization of different content.
- The program can work perfectly with Wikipedia data but needs work to work on any webpage.

IMPROVEMENTS

- Changing the way how our program handles the input data, the program can be used for classifying both Wikipedia and other web pages.
- Adding more categorization topics ensures that the program can work on any webpage and allows for any content to be categorized under that topic.
- Using better classification methods including Support Vector Machines instead of Naïve Bayes Classifier can provide more accurate results.