PROJECT REPORT

Content Based Classification using Natural Language Processing

Hardik Bhadja (hxb162230)

Suraj Poojary (ssp151830)

Problem Description

Document classification or document categorization is a problem in library science, information science and computer science. The task is to assign a document to one or more classes or categories which can be done manually or algorithmically. We aim to use natural language processing concepts to parse these files and develop a machine learning classifier which will be able to classify these documents into their correct categories/classes.

Proposed Solution and Implementation Details

Proposed solution

We target documents belonging to 4 classes in our implementation; namely science, politics, sports and religion. Each class has 250 documents (sentences). Our program takes these sentences and applies NLP processes on it to come up with features that will help classify any unseen document.

Baseline system

Our Baseline system uses a bag of words technique to predict the classes of the documents. Initially, we found the most occurring words for each of our 4 classes. We then manually selected the words that would correctly represent the category and distinguish itself from other categories.

For a tokenized word in the given test document, we calculate the relative probability of that word in each class and assign the class with the highest probability to that word. Finally, the class that occurs most in that document is assigned as the class for that document.

Improvement strategy

Our improvement strategy uses NLP concepts to extract features from the documents which will then be used to differentiate between classes. There are 3 types of features that we consider namely lexical, syntactic and semantic features and 2 features within each type as shown in the examples section.

The process first removes the bad characters and symbols to cleanse the data in the documents. It then removes extra spaces and tabs such that each document is space delimited. Once we have the clean data, we divide it into 80% training and 20% testing. For the testing file, we process each line and extract the NLP features from it and generate a local vector for each line. Next, we generate a global vector with all unique features from the training file and assign appropriate values from the local vector to the global vector. This global vector is basically what gets fed to the classifier.

We follow a similar process to generate the global vector for the test file but this time the NLP features obtained are cross-referenced with the training ones first, post which the test file to be fed to the classifier is ready.

We use a Naive Bayes Gaussian Classifier for the project. It calculates the probability contributions of each feature to prepare a model and then predict the class of the given input.

Examples

Lexical Features:

NLTK Porter Stemmer and WordNet Lemmatizer will be used to extract lexical features.

Eg: For sentence "Do you really think it is weakness that yields to temptation I tell you that there are terrible temptations which it requires strength strength and courage to yield to Oscar Wilde"

Stemmed sentence: "Do you realli think it is weak that yield to temptat I tell you that there ar terribl temptat which it requir strength strength and courag to yield to Oscar Wild""

Lemmatised sentence: "Do you really think it be weakness that yield to temptation I tell you that there be terrible temptation which it require strength strength and courage to yield to Oscar Wilde"

Syntactic Features:

NLTK POS tagger and Dependency Grammar will be used to implement POS tagging and dependency parsing respectively.

Eg: For sentence "This is a simple sentence"
POS tagged Output: [('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('simple', 'JJ'), ('sentence', 'NN')]

For sentence “Ballmer has been vocal in the past warning that Linux is a threat to Microsoft”
Dependency grammar:

Ballmer	NNP	4	nsubj
has	VBZ	4	aux
been	VBN	4	cop
vocal	JJ	0	ROOT
in	IN	4	prep
the	DT	8	det
past	JJ	8	amod
warning	NN	5	pobj
that	WDT	13	dobj
Linux	NNP	13	nsubj
is	VBZ	13	cop
a	DT	13	det
threat	NN	8	rcmod
to	TO	13	prep
Microsoft	NNP	14	pobj
.	.	4	punct

Semantic Features:

NLTK Wordnet for semantic relations hypernymy and practnlptools package within NLTK for semantic role labelling.

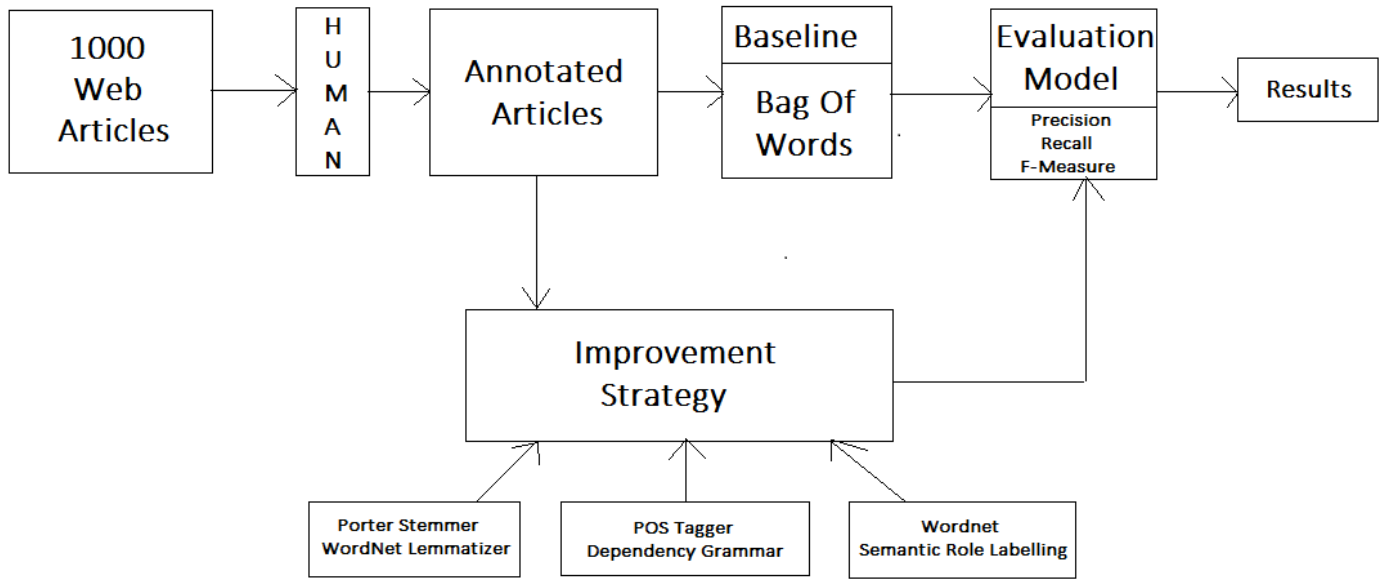
Eg: dog.hypernyms() -> [Synset('canine.n.02'), Synset('domestic_animal.n.01')]

For sentence “He created the robot and broke it after making it”, the semantic role labeling yields
[{'A1': 'the robot', 'A0': 'He', 'V': 'created'}, {'A1': 'it', 'A0': 'He', 'AM-TMP': 'after making it.', 'V': 'broke'}, {'A1': 'it.', 'A0': 'He', 'V': 'making'}]

Programming tools

Python as the core programming environment.
NLTK package for Natural language processing.
Machine learning python library scikit learn for modelling classifiers.
Stanford Dependency Parser package for dependency parsing.
PractNLPTools package for semantic role labelling

Architectural diagram



Results

Baseline:

Some documents could not be classified into any categories as shown below
Cannot Be classified::politics.test.txt Document Content:::farzinapollo3.ntt.jp Farzin Mokhtarian writesFrom Kayhan Havai 1026 ...
Cannot Be classified::politics.test.txt Document Content:::shut up andi
Cannot Be classified::science.test.txt Document Content:::
Cannot Be classified::sports.test.txt Document Content:::Bo Bilinsky

	politics	religion	science	sports	total-predicted
politics	38	8	7	13	|66
religion	4	37	5	5	|51

science	4	5	35	10	|54
sports	2	0	2	21	|25
Total	48	50	49	49	

Class: politics Recall: 0.7916666666666666 Precision: 0.5757575757575758
Class: religion Recall: 0.74 Precision: 0.7254901960784313
Class: science Recall: 0.7142857142857143 Precision: 0.6481481481481481
Class: sports Recall: 0.42857142857142855 Precision: 0.84

Improvement strategy:

Accuracy: 0.81

	precision	recall	f1-score	
politics	0.82	0.82	0.82	
religion	0.76	0.75	0.75	
science	0.80	0.75	0.78	
sports	0.86	0.93	0.90	
avg / total	0.81	0.81	0.81	200

Analysis

Below are the average results after each of the feature was left out of the model.

	Accuracy	Precision	Recall	F-Measure
Without semantic	0.80	0.80	0.80	0.80
Without syntactic	0.81	0.81	0.81	0.81
Without lexical	0.625	0.625	0.625	0.625

It is evident that lexical features contribute more to the accuracy of the model than the semantic and syntactic ones.

Implementation Challenges

Implementing baseline system specific to multi label classifier is a challenging task in its own, determining the bag of words was one task that took a lot of manual effort. We programmatically calculated the frequency of all words in each class. Then, we had to manually pick up words that occurred very often and those which effectively represent the class as well as distinguish the class from other classes. Other than that augmenting the probability to the very naïve strategy e.g. frequency of words, in the bag of words of each class was a little challenging.

Learning how to map the NLP features to a particular structure as required by the Machine learning classifiers was one other challenging task, as we were doing this for the first time. Coming up with the local features for each line in the training set wasn't enough as each line may have different types and number of features. Whereas the classifier requires a standard set of columns to be fed. To achieve this, we built a global vector by taking all the unique features in the entire training dataset and then assigned

corresponding values to the columns by cross-referencing it to the local vector of each line. This global vector was then fed to the classifier.

Pending Issues
None

Potential Improvements
NLP: Syntactic and semantic features could be replaced by Bi-Grams as Bi-Grams have often been found to yield better accuracy. Also, as evident from our analysis, they did not contribute much to the accuracy of the model.

Machine Learning: Due to the huge amount of diverse features, the final global vector for machine learning was found to be sparse. Applying feature extraction will allow us to eliminate features that do not contribute much to the prediction which may then increase the accuracy of the model.