DWDM

Project

Introduction

The project aims to extract names of methods mentioned in scientific research papers.

Dataset

The project used 378 research papers from various fields available in pdf format.

Approach

3.1 Text Extraction

The text was mined from pdf research documents using a python module PDF2Text. The tool was used to convert a given pdf document into string of text with all special objects (like links, images etc.) removed.

3.2 Rule Based Approach

The baseline algorithm to extract methods from the documents is as follows

1. Extract text from the pdf documents,

2. Find all the named entities mentioned in the paper,

3. Filter the extracted named entities

    a. Manual Filtering

    b. Word2Vec

3.4 Named Entity Recognition

Named Entity Recognition (NER) labels sequences of words in a text which are the names of things, such as person or organisation or location names.

For NER, a python Natural Language Toolkit (nltk) was used. The task was carried out as follows.

1. Sentence Tokenization: The raw text obtained after pdfmining (3.2) had arbitrary new lines, and was not divided into sentences. NLTK's sent tokenizer segments such text into a list of sentences. It uses an unsupervised algorithm (Kiss & Strunk, 2006) which takes into account sentence boundary punctuation, sentence starting words, abbreviations and collocations to estimate sentence boundaries.

2. Named Entity Recognition : NLTK provides a NER model which can identify named entities in a given sentence. However, The model has been trained on English corpus and not some science specific corpora. This results in a lot of other type of entities, not relevant for the task, such as names of universities, locations, author names etc to be identified to also get identified.3.4 Filtering Extracted Named Entities

To remove unwanted named entities from the list of entities obtained above, the following filters was done.

1. Location Names: Using pycountry a list of names of locations (cities and countries) was created. Entities which existed in the locations lists were removed.

2. Author Names: A lookup list of author names was created. The lookup list was

created from the set of available documents using certain assumptions, like,

names of authors would be present somewhere near the top of the raw text.

entities matching

3. Names of Universities, Departments, Schools: A simple regex match was used to filter out entities which contained the strings ‘university’, ‘school’ and ‘department’.

4. Cited Methods : Research papers discuss methods used by other authors. To remove such entries, named entities from the citations sections were found. The citations were identified using a regex match. If the Hamming distance between an entity and a citation was greater than a threshold, that particular named entity was dropped.

3.5 Semi Supervised Approach (using Word2Vec)

Word2vec is a two-layer neural net that processes text. Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus.

In the training set, the identified named entities were converted in vector representation. The labels(whether false positive or true positive) and vector representations of the entities were stored in a lookup table.

In the test set, entities were identified and their word vectors were calculated.

Distances(Euclidean) of the word vectors were calculated from the entities in entries of the lookup table formed using the training set.

An entity from the identified entities was labeled a method if its nearest neighbour is labelled as true positive

4) **TF-IDF Approach:**

1) **Using TF-IDF vectorizer of sklearn**
2) **Find k important words**

Results:

The performance of the procedure was measured by calculating precision and recall scores.

P recision =TP/(TP + FP)

Recall = TP/(TP + FN)

F 1 − S core =P recision * Recall /2 P * recision + Recall

Where,

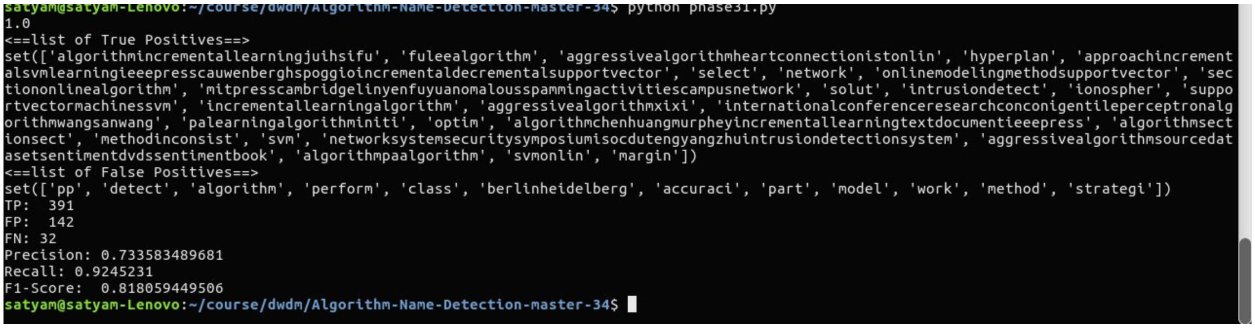
TP(True Positives) is the number of correctly identified methods,

FP(False Positive) is the number of falsely identified methods,

FN(False Negative) is the number of algorithms that the procedure failed to identify.

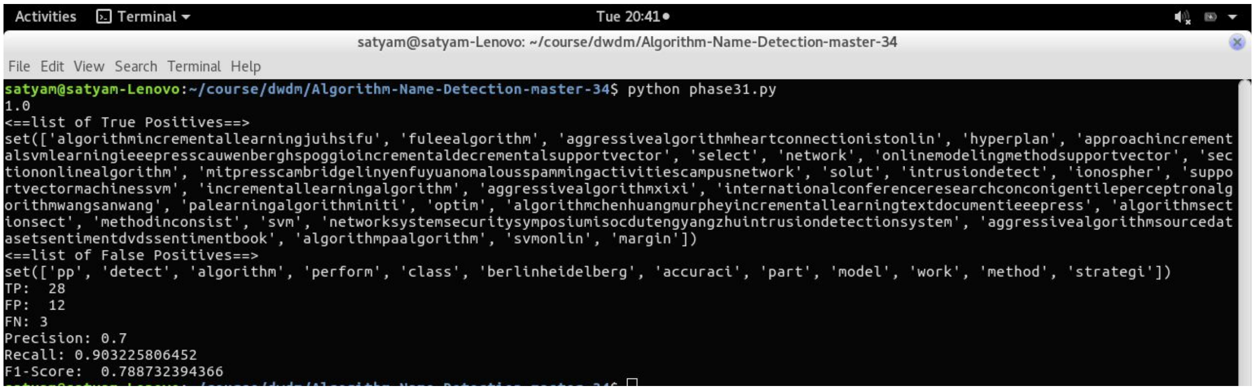
Testing on large dataset of 40-45 documents:

Rule based extraction:



```
satyam@satyam-Lenovo:~/course/dwdm/Algorithm-Name-Detection-master-34$ python phase31.py
1.0
<==list of True Positives==>
set(['algorithmincrementallearningjuihsifu', 'fuleealgorithm', 'aggressivealgorithmheartconnectionistonlin', 'hyperplan', 'approachincrement
alsvmlearningieeepresscauwenberghspoggioincrementaldecrementalsupportvector', 'select', 'network', 'onlinemodelingmethodsupportvector', 'sec
tiononlinealgorithm', 'mitpresscambridgelinyenfuyuanomalousspammingactivitiescampusnetwork', 'solut', 'intrusiondetect', 'ionospher', 'suppo
rtvectormachinessvm', 'incrementallearningalgorithm', 'aggressivealgorithmxixi', 'internationalconferenceresearchconconigentileperceptronalg
orithmwangsanwang', 'palearningalgorithminiti', 'optim', 'algorithmchenhuangmurpheyincrementallearningtextdocumentieeepress', 'algorithmsect
ionsect', 'methodinconsist', 'svm', 'networksystemsecuritysymposiumisocdutengyangzhuintrusiondetectionsystem', 'aggressivealgorithmsourcedat
asetsentimentdvdssentimentbook', 'algorithmpaalgorithm', 'svmonlin', 'margin'])
<==list of False Positives==>
set(['pp', 'detect', 'algorithm', 'perform', 'class', 'berlinheidelberg', 'accuraci', 'part', 'model', 'work', 'method', 'strategi'])
TP:  391
FP:  142
FN: 32
Precision: 0.733583489681
Recall: 0.9245231
F1-Score:  0.818059449506
satyam@satyam-Lenovo:~/course/dwdm/Algorithm-Name-Detection-master-34$
```

Testing on 11 documents:



Activities
Terminal
Tue 20:41
satyam@satyam-Lenovo: ~/course/dwdm/Algorithm-Name-Detection-master-34
File
Edit
View
Search
Terminal
Help

```
satyam@satyam-Lenovo:~/course/dwdm/Algorithm-Name-Detection-master-34$ python phase31.py
1.0
<==list of True Positives==>
set(['algorithmincrementallearningjuihsifu', 'fuleealgorithm', 'aggressivealgorithmheartconnectionistonlin', 'hyperplan', 'approachincrement
alsvmlearningieeepresscauwenberghspoggioincrementaldecrementalsupportvector', 'select', 'network', 'onlinemodelingmethodsupportvector', 'sec
tiononlinealgorithm', 'mitpresscambridgelinyenfuyuanomalousspammingactivitiescampusnetwork', 'solut', 'intrusiondetect', 'ionospher', 'suppo
rtvectormachinessvm', 'incrementallearningalgorithm', 'aggressivealgorithmxixi', 'internationalconferenceresearchconconigentileperceptronalg
orithmwangsanwang', 'palearningalgorithminiti', 'optim', 'algorithmchenhuangmurpheyincrementallearningtextdocumentieeepress', 'algorithmsect
ionsect', 'methodinconsist', 'svm', 'networksystemsecuritysymposiumisocdutengyangzhuintrusiondetectionsystem', 'aggressivealgorithmsourcedat
asetsentimentdvdssentimentbook', 'algorithmpaalgorithm', 'svmonlin', 'margin'])
<==list of False Positives==>
set(['pp', 'detect', 'algorithm', 'perform', 'class', 'berlinheidelberg', 'accuraci', 'part', 'model', 'work', 'method', 'strategi'])
TP:  28
FP:  12
FN: 3
Precision: 0.7
Recall: 0.903225806452
F1-Score:  0.788732394366
```