SEMANTIC SEARCH APPLICATION

TEAM TRANSFORMERS
SHARAYU MANTRI (SSM171330)
MEHAK BERI (MXB166430)
RAHUL KALRA (RXK163530)

CS 6320 | NATURAL LANGUAGE PROCESSING | FALL 2017

Contents0

Problem Description

Implement a semantic search application that produces improved results using NLP features and techniques. The project implements a keyword-based strategy and an improved strategy using NLP feature and techniques.

Proposed Solution

The solution to the given problem has been broken into the following tasks that need to be performed:

1. **Task 1:**

Create a corpus of News articles. The corpus should contain at least:

- 1,000 articles
- 100,000 words

2. **Task 2:**

Implement a shallow NLP pipeline to perform the following:

- Keyword search index creation
    - Segment the News articles into sentences
    - Tokenize the sentences into words
    - Index the word vector per sentence into search index such as Lucene or SOLR
- Natural language query parsing and search
    - Segment an user's input natural language query into sentences
    - Tokenize the sentences into words
    - Run a search/match with the search query word vector against the sentence word vector (present in the Lucene/SOLR search index) created from the corpus
- Evaluate the results of at least 10 search queries for the top-10 returned sentence matches

3. **Task 3:**

Implement a deeper NLP pipeline to perform the following:

- Semantic search index creation
    - Segment the News articles into sentences
    - Tokenize the sentences into words
    - Lemmatize the words to extract lemmas as features Stem the words to extract stemmed words as features

    - Part-of-speech (POS) tag the words to extract POS tag features
    - Syntactically parse the sentence and extract phrases, head words, OR dependency parse relations as features
    - Using WordNet, extract hypernyms, hyponyms, meronyms, AND holonyms as features
    - Index the various NLP features as separate search fields in a search index such as Lucene or SOLR
- Natural language query parsing and search
    - Run the above described deeper NLP on an user's input natural language and extract search query features
    - Run a search/match against the separate or combination of search index fields created from the corpus
- Evaluate the results of at least 10 search queries for the top-10 returned sentence matches

4. **Task 4:**

Improve the shallow NLP pipeline results using a combination of deeper NLP pipeline features.

Implementation & Results

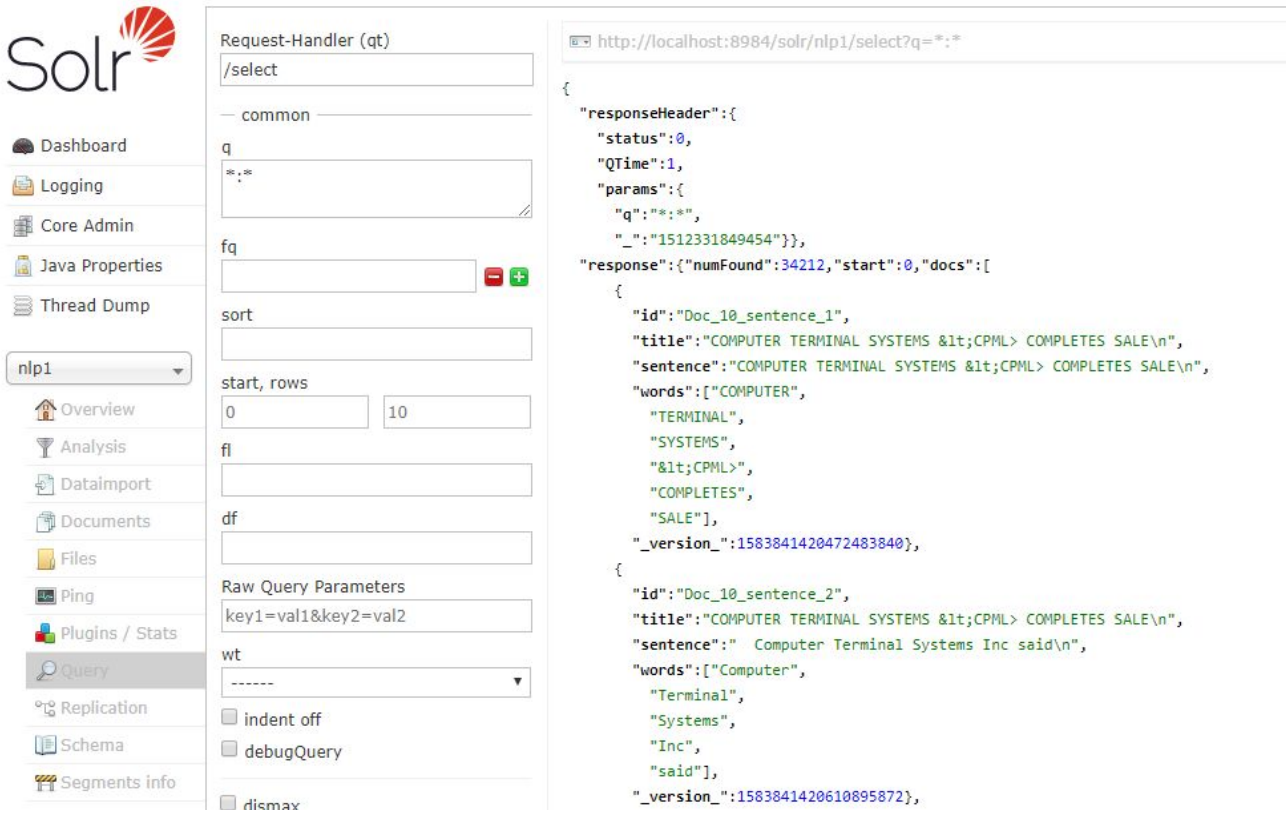
TASK 1:

- We used The Reuters-21578 benchmark corpus, ApteMod version as our main corpus.
- Corpus Source: http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html
- Selected 1324 articles out of the main reuters corpus based on file size. The average word count in each of these articles is around 200 words
- Total file size of our corpus= 1.78 MB
- Total number of articles = 1324
- Total Number of Words in Entire Corpus = 291732
- Average word count per article = 200

TASK 2:

- Created a python file which takes as input all files in the corpus, segments it into sentences and then into words.
- The file outputs data into SOLR in the following format:
  "id": xxx
  "title": xxx
  "sentence": xxx
  "words": [xxx, xx , xxx]
  "__version__": xxx

- The search query vector is matched against the sentence vectors stored in SOLR. If we input "*,score" in *fl* field in SOLR, we can also see scores based on the number of keywords matched in sorted descending order of score.
- Tokenization of corpus is done using "nltk.tokenize", "sent_tokenize" is used for parsing text corpus in sentence and "word_tokenize" is used for parsing sentence into words.
- A screenshot below shows the format in which data is entered in task 2.



- Example Query for Task 2:
  Query String: 'Company suffering from losses'
  Result: (Relevant results highlighted in green)

runfile('C:/Users/Mehak Beri/Desktop/MS/fall 17/nlp/Final Project/task4/test.py', wdir='C:/Users/Mehak Beri/Desktop/MS/fall 17/nlp/Final Project/task4')
Reloaded modules: Task3, Constants, SolrCommunicator

Choose from the following:
Enter 1 to Read Corpus
Enter 2 to execute Task 2
Enter 3 to execute Task 3
Enter 4 to execute Task 4
::2
in task 2

Please input the query sentence/phrase: Company suffering from losses
Query: Company suffering from losses
words:Company words:suffering words:from words:losses
Saw 10 result(s).
Score : 11.770885
Sentence : In an economy suffering from inflation of around 10 pct," he said.

Score : 11.059399

Sentence : The People's Daily said Henan, Shaanxi, Gansu and Hebei are also suffering from drought.

Score : 9.915087
Sentence : Toyota severed ties with its Philippine partner of 20 years, <Delta Motor Corp>, in 1984 because the local company was suffering financial difficulties.

Score : 9.460955
Sentence : The company said over 576 mln dlrs of last year's group losses came from commercial insurance lines and 282 mln dlrs from its personal auto insurance business.

Score : 9.409543
Sentence : "We are already making vast losses in Surinam and you can't expect any company to remain operating with losses," the spokesman said.

Score : 8.881467
Sentence : The company said it omitted the common dividend to cover both losses from a commercial ship conversion contract and increased reserves for previously announced discontinued shipyard operations.

Score : 8.771183
Sentence : Sherwoood said the company's losses were largely attributable to non-recurring events and provisions.

Score : 8.739664
Sentence : Commenting on the year's performance, the company said it suspended operations at the 60 pct-owned Agnew Nickel mine because of losses sustained from declining nickel prices.

Score : 8.204539
Sentence : Flowers Industries Inc said it expects lower earnings for the current year due to operating losses incurred by recent acquisitions and possible nonrecurring losses resulting from its restructuring efforts.

Score : 7.9684324
Sentence : Krestmark had revenues of about 40 mln dlrs and operating losses of three mln dlrs in 1986, the company said.

- RESULT EVALUATION

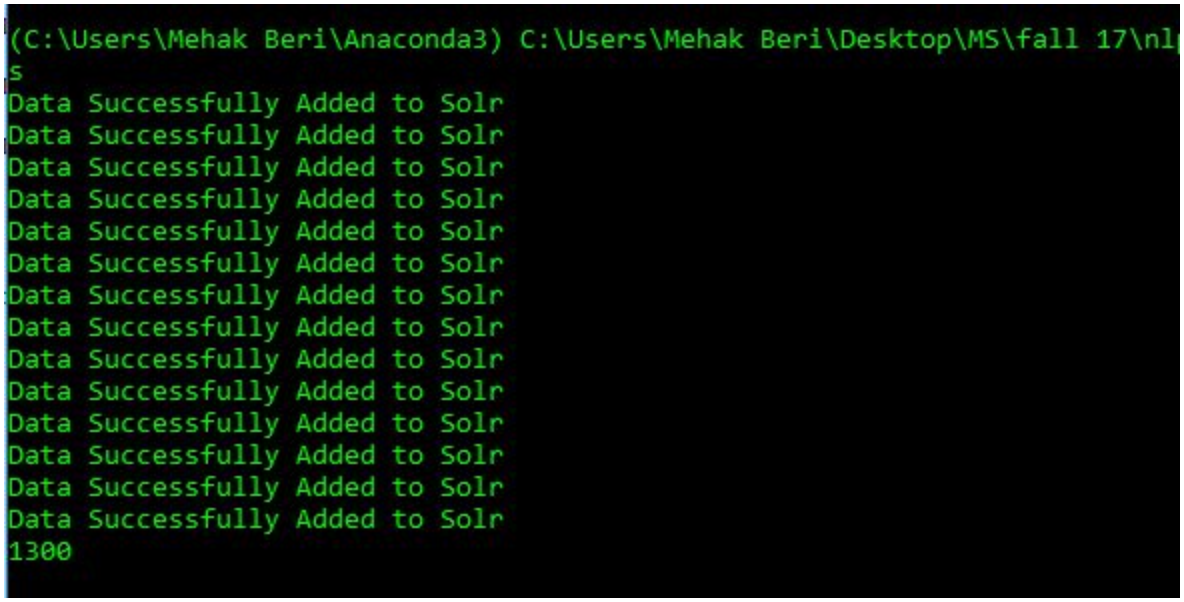
Query	Relevant Results	Irrelevant Results	Accuracy
Company suffering from losses	7	3	70%
Recent government tax reforms	6	4	60%
What are the food prices in US	2	8	20%
Oil exploration agreements	4	6	40%
Investment in grain production	5	5	50%
Cost of grain certificates compared to cash	6	4	60%
Self sufficiency in grain output	5	5	50%

Company acquisitions	10	0	100%
Sales exceeding 1.4 billion	5	5	50%
French maize exports	6	4	60%
Average Accuracy : 56%			

The results are fairly accurate. It can be seen that more generic the query, the more relevant are the results obtained. Less relevant results are obtained if the query is put in a sentence form with extra words like 'what' , 'are' etc.

TASK 3:

- Tokenization in Task 3 done in same way as of task2.
- Features (posTags, lemmas, stemma, hypernyms, hyponyms, meronyms, holonym, headword) were added for each word of each sentence of each document.
- Pos Tags are implemented using nltk.pos_tag, Lemmatization and Stemming is done using nltk.stem package. "WordNetLemmatizer" finds lemmas of the word and "PorterStemmer" is used for stemming.
- Hyponyms, Hypernyms, Meronyms, Holonyms are implemented using "wordnet" class from nltk.corpus package. For task 3, We got all the possible Hyponyms, Hypernyms, Meronyms, Holonyms for all the possible synsets of a given word and added it to solr.
- Headword was implemented using "StanfordDependencyParser" Class of "nltk.parse.stanford" package. We extracted the root or head word on entire dependency parse tree and added it to headWord field of the sentence document in solr.
- Since a huge amount of data was sent, pagination was done in order to break all data in chunks and send it to solr. Output can be seen as follows:



- After implementing deeper NLP pipeline for semantic search, each word in each sentence has information attached to it in the following manner: (information skipped for brevity)

```
  "id":"Doc_7150_sentence_7",
        "title":["  Energy Secretary John Herrington  said he would rule out a
tax on gasoline as one option to help  avert what his departmet has called the
threat of increased  reliance on foreign oil in the coming years."],
        "sentence":["Herrington proposed the increased depletion allowance to
the White House Tuesday but the White House reaction was cool."],
        "words":["Herrington",
          "proposed",
         ---
"],
        "posTags":["Herrington NNP",     //word<space>posTag
          "proposed VBD",
          ----
          ". ."],
        "lemmas":["Herrington",
          "proposed",
          "the",
          ----
          "."],
        "stemma":["herrington",
          "propos",
          -----
          "."],
        "hypernyms":["[]",
          "['declare', 'plan', 'intend', 'choose', 'request']",
            ----
],
        "hyponyms":["[]",
          "['advance', 'feed_back', 'move', 'proposition', 'recommend',
'submit', 'introduce', 'offer', 'nominate']",
----],
        "meronyms":["[]",
          "[]",
],
```

```
          "holonyms":["[]",
            "[]",
],
          "headWord":["proposed"],

          "_version_":1585816080237985803,
          "score":6.845612}
```


- Natural language query which is input by the user is parsed and analyzed like the corpus above. All features for the input query are calculated as above and matched against the corpus.

Result for query 1: company suffering from losses
Saw 10 result(s).

Score : 185.1235

Sentence : Toyota severed ties with its Philippine partner of 20 years, <Delta Motor Corp>, in 1984 because the local company was suffering financial difficulties.

Score : 168.36765

Sentence : In an economy suffering from inflation of around 10 pct," he said.

Score : 164.0864

Sentence : Of them, some 18,000 clients claimed they could get back neither gold or money, suffering an aggregate loss of 150 billion yen, local press reports said.

Score : 163.38423

Sentence : The People's Daily said Henan, Shaanxi, Gansu and Hebei are also suffering from drought.

Score : 155.12308

Sentence : "We are already making vast losses in Surinam and you can't expect any company to remain operating with losses," the spokesman said.

Score : 147.2662

Sentence : The agency said some areas of Guangxi, Hubei, Shanxi and other provinces have been suffering a drought for more than seven months.

Score : 145.79619

Sentence : Sherwoood said the company's losses were largely attributable to non-recurring events and provisions.

Score : 137.34508

Sentence : Krestmark had revenues of about 40 mln dlrs and operating losses of three mln dlrs in 1986, the company said.

Score : 130.10054

Sentence : <Billiton International Metals B.V.>, the Dutch mining company, has urged Surinam to change policies it says are causing heavy losses on bauxite mining operations there, a company spokesman said.

Score : 126.82169

Sentence : Commenting on the year's performance, the company said it suspended operations at the 60 pct-owned Agnew Nickel mine because of losses sustained from declining nickel prices.

- RESULT EVALUATION

Query	Relevant Results	Irrelevant Results	Accuracy
Company suffering from losses	7	3	70%
Recent government tax reforms	8	2	80%
What are the food prices in US	2	8	20%
Oil exploration agreements	7	3	70%
Investment in grain production	4	6	40%
Cost of grain certificates compared to cash	3	7	30%
Self sufficiency in grain output	2	8	20%
Company acquisitions	8	2	80%
Sales exceeding 1.4 billion	7	3	70%
French maize exports	7	3	70%
Average Accuracy : 55%			

It can be seen that results improve in most queries. But it is also seen that relevant results do not always have the maximum score.

TASK 4:

1. We Used Lesk Algorithm for Word Sense Disambiguation. Given an ambiguous word and the context in which the word occurs, Lesk returns a Synset with the highest number of overlapping words between the context sentence and different definitions from each Synset.
2. Also, We have removed stop Words from the search query and then performed search, this improved our search these removed many false positives as those sentences that were matching only stop words or mostly stop words were now ignored entirely.
3. We provided different weights for different factors to improve the accuracy.
   a. As we removed the stop words from our query string and kept the relevant words we gave the words a higher weight i.e. 5. As a sentence that matches relevant words can be considered as a good result.
   b. Weight to Part of Speech tags were also given a higher value. i.e. 6 as POS helps in disambiguation and knowing the correct POS for each term can help us building an

accurate search.

c. Lemmas were given a high weight as well i.e. 3 as naïve word matching doesn't match the words which are lemmas to each other, and hence to increase the accuracy we gave it a higher value.

d. Lesk Algorithm weight was given the highest i.e. 8 as Lesk algorithm worked well for queries in disambiguating the words sense and hence yields better results for our queries.

e. Stemmas weight was kept at 1 as we did capture the same results in Lemmas and increasing the weight doesn't increase our accuracy.

f. If Hypernyms, Meronyms and Holonyms are given higher weights and most words in a sentence does not have Hypernyms this will drop the score of the sentence for a given query on the other hand the irrelevant match of Hypernyms , Holonyms will increase the score of sentence for given search query giving false positive outcomes.

g. Headwords weights were increased and were given a weight of 2, as Head words captures the most important term of a sentence so if it matches there is a high probability that the sentence is a good result.

4. Also, Among various weights we experimented, Higher Negative weight gave less accurate results. Reason for this was, the possible matches had negative value and the matches which were incorrect got higher weight.
5. Initially we gave Higher weight to Hyponyms, but after experimenting we found out that giving zero weight to hyponym yields better results.
6. Instead of matching each POS tag of a search query with the POS tags of words in the corpus, we inserted WORD<space>POSTAG into the corpus for each word, so that for each word being queried, indexing is done based on the part of speech tag as well as the word , of the word being queried. This removes a lot of noise.

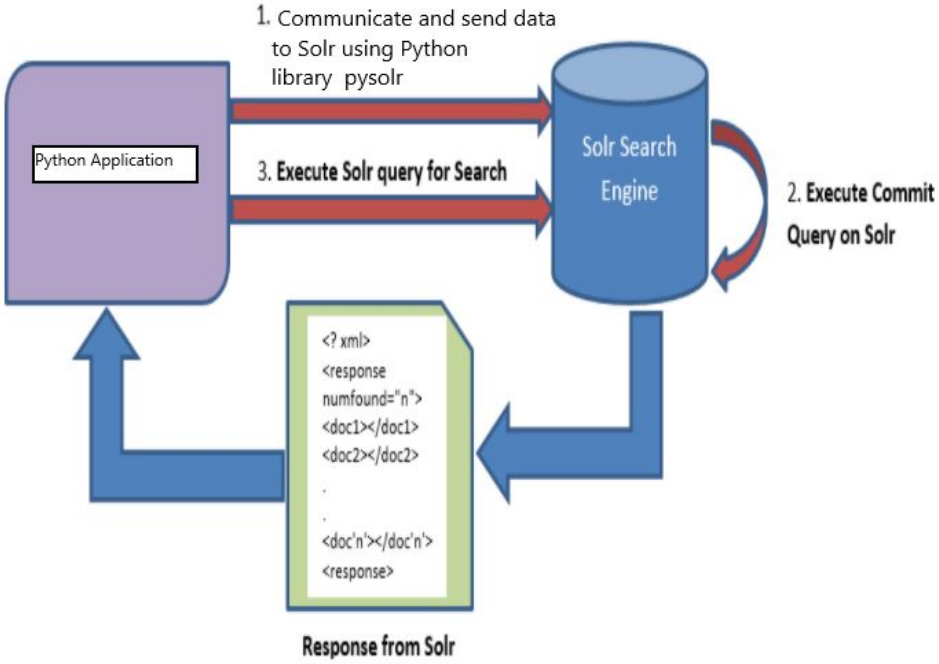
RESULT EVALUATION – task 4 with weights

Query	Relevant Results	Irrelevant Results	Accuracy
Company suffering from losses	7	3	70%
Recent government tax reforms	9	1	90%
What are the food prices in US	2	8	20%
Oil exploration agreements	8	2	80%
Investment in grain production	3	7	30%
Cost of grain certificates compared to cash	3	7	30%
Self sufficiency in grain output	4	6	40%
Company acquisitions	10	0	100%
Sales exceeding 1.4 billion	7	3	70%
French maize exports	8	2	80%
Average Accuracy : 61%			

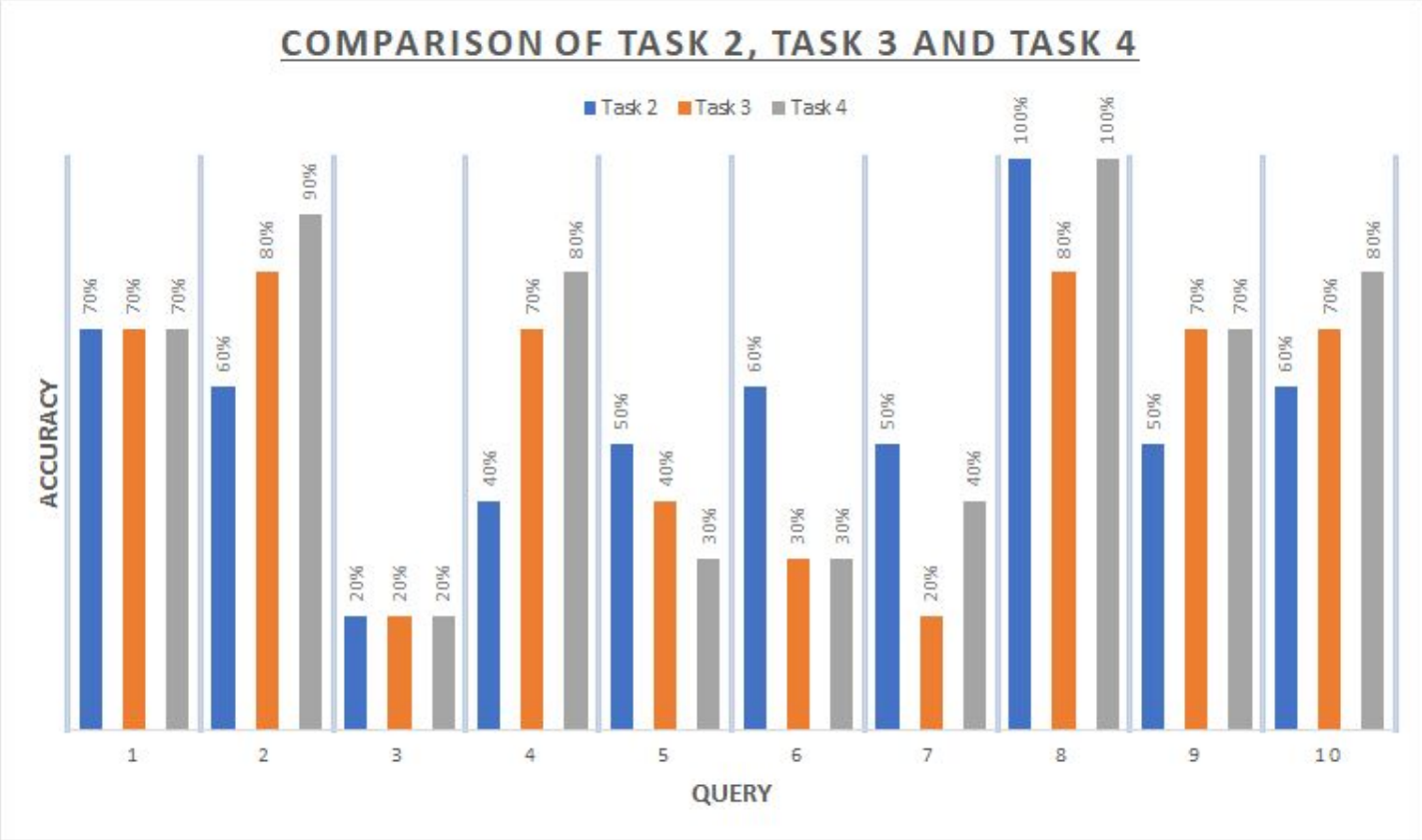
Programming Tools

- Python 3.6.1
- SOLR 7.1.0
- NLTK 3.2.5
- Stanford Core NLP 2017-06-09

Architectural Design


1. Communicate and send data to Solr using Python library pysolr
Python Application
3. Execute Solr query for Search
Solr Search Engine
2. Execute Commit Query on Solr
<? xml>
<response
numfound="n">
<doc1></doc1>
<doc2></doc2>
.
.
<doc'n'></doc'n'>
<response>
Response from Solr

Result Analysis



Problems Faced

- Selecting good files from the corpus was a task initially. To solve the issue, we sorted the files based on their size and took medium sized files and made sure that the average number of words per file was nearly 200
- When we added per word data for NLP features, the data stream became too big to upload at once on Solr. So we paginated the content and divided data into chunks so that the communication pipeline between python and Solr does not get flooded
- Solr was throwing errors if the sentence sizes were too large for it to parse. So we discarded sentences with number of words greater than 50
- While looking for libraries which could help us find the headwords of sentences in our corpus, we were initially considering the headword finder that NLTK provides, but using it would require dependency grammar trees for each sentence tree being evaluated as input, so we researched and came across the Stanford core nlp library and used it in the project
- Calculation of headwords for all sentences in the corpus took approximately 7.5 hours.

Pending Issues

We have Tested Task 4 Weights distribution on entire corpus, but Task 4 other improvements like additional field for Word Sense Disambiguation and Instead of POS tag, it has 'word<space>posTag' is not yet tested on entire corpus. It is tested only on small selected set of corpus. We need to check the accuracy of this improvement on entire corpus.

Potential Improvements

We are yet to add all NLP features namely hypernym, hyponym, meronym etc for a given word and for specific tag selected by pos tagger.. Currently the features like hypernym, holonym have been added by considering the 'word' and not the 'word & its posTag'.

Extraction of HeadWord for every sentence is very time consuming. This can be added in one of potential improvement tasks.

Also, we have added **Title** as a field but we have not indexed it and not included it in search query. But, Searching for headWord or Synonyms of HeadWord in Title with higher weight can definitely improve the accuracy of the results.