CS6370
NLP

Course Project

- Naveen Vakada (CS20S012)
- Arup Das (CS20S016)

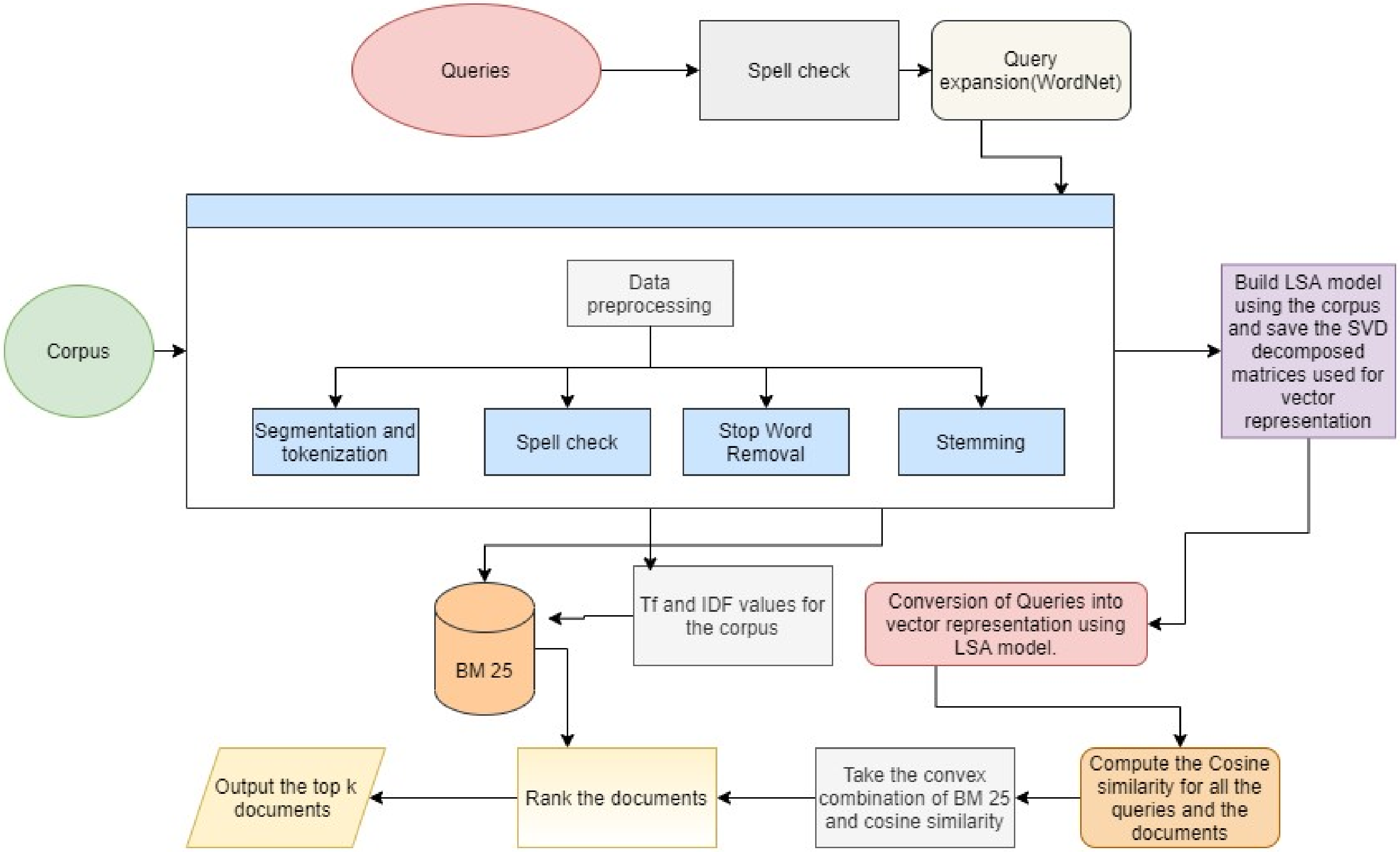
Limitations of the current search engine

Fails to classify documents containing synonymous words or documents with the same context as relevant documents.

The title of the document was not used.

Does not consider the co-occurrence relation between the terms in similar documents.

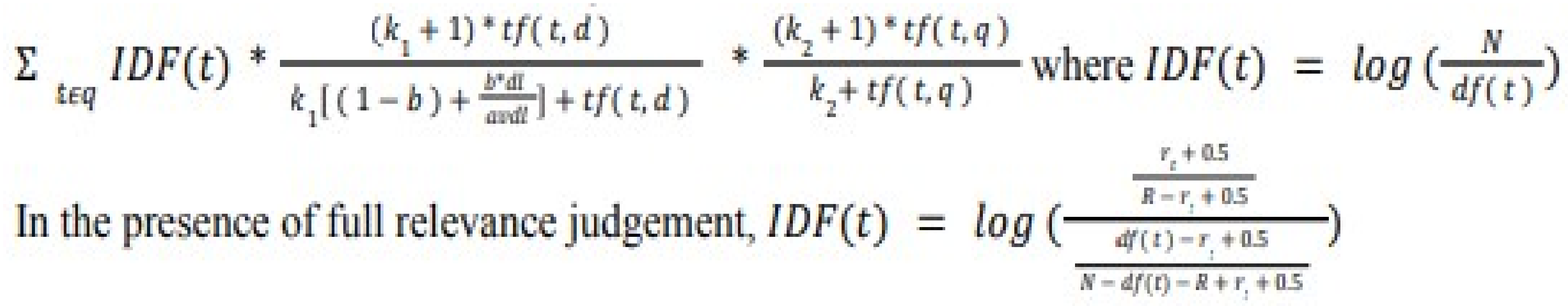
Proposed methodology



Queries
Spell check
Query expansion(WordNet)
Corpus
Data preprocessing
Segmentation and tokenization
Spell check
Stop Word Removal
Stemming
Build LSA model using the corpus and save the SVD decomposed matrices used for vector representation
BM 25
Tf and IDF values for the corpus
Conversion of Queries into vector representation using LSA model.
Output the top k documents
Rank the documents
Take the convex combination of BM 25 and cosine similarity
Compute the Cosine similarity for all the queries and the documents

Completed tasks

Implemented BM25



$$\Sigma_{t \epsilon q} IDF(t) * \frac{(k_1 + 1) * tf(t, d)}{k_1[(1 - b) + \frac{b*dl}{avdl}] + tf(t, d)} * \frac{(k_2 + 1) * tf(t, q)}{k_2 + tf(t, q)} \text{ where } IDF(t) = log(\frac{N}{df(t)})$$

In the presence of full relevance judgement, $IDF(t) = log(\frac{\frac{r_i + 0.5}{R - r_i + 0.5}}{\frac{df(t) - r_i + 0.5}{N - df(t) - R + r_i + 0.5}})$

Used titles of the document to improve the performance.

Completed reading about how to implement LSA in python.

Pending Work

Implementing LSA

Implementing Statistical hypothesis testing to compare the mean nDCG of the old method and the new proposed method.

Implementing query expansion.

Implementing Spell Check.