CS6370: Natural Language Processing
Project Proposal

23rd March 2023

Group: 34
Names: Chatur B (CS20B018), Santosh G (EE19B055)

Introduction:

We have built a search engine to retrieve information from the Cranfield Data Set. The search engine is a Vector Space model-based engine.
The engine is capable of pre-processing the queries and documents by tokenization, lemmatization/stemming, stop-word removal, etc. Similarly, the documents are processed by segmenting, tokenizing, lemmatizing, etc; the documents and the left-over words are all considered features.

The search engine, built as an Information retrieval system, compares the similarity between the query and documents by computing the cosine similarity, which is based on matching terms/features.

Observations:

Upon implementing and performing tests, though it was fairly accurate in retrieving information, we have observed a few anomalies in the retrieved results i.e., though the engine functions as it is coded, the results deviate from the ideal/expected results.
Example: Upon querying for “plant eaters”, document S2 (“Carnivores are typically meat eaters and not plant eaters”) is given a higher score (indicating higher similarity) when compared to S3 (“Deers eat grass and leaves”)

Hypotheses:

- Vector Space Model (VSM) considers every word of the processed documents as a feature
- VSM doesn’t take polysemy and synonymy into account while ranking the documents based on similarity:
    - A word can have multiple meanings (Polysemy), and despite having multiple semantic meanings, all the instances of the given word form will be considered as the same feature, resulting in false positives.
    - A meaning/expression can be expressed using multiple words; despite the different word forms having the same meaning, they will be considered as different features and will be given a low similarity score.

- To summarise, VSM fails to consider context appropriately, and context is essential to understanding and relating the documents.

Project Description and Methodology:

We plan to implement a Search engine that is capable of considering context and word relations, at least to a certain extent, while retrieving information.
We aim to increase the performance of the search engine for information retrieval and will start by implementing Explicit Semantic Analysis (ESA) and exploring other possible methods.

ESA: ESA leverages the external knowledge available and becomes capable of capturing the underlying meaning and context in the documents. The initial stages of implementing ESA involve training the model with a large corpus of documents, such as Wikipedia, etc. That data will be used to extract and establish a semantic representation of words and documents.

The semantic representation will account for synonymy, and polysemy and would take context also into account while performing information retrieval.

There are several libraries in Python such as “gensim”, “scikit-learn” to aid the implementation of ESA. Using such libraries and other methods, we will train the engine using a large corpus and implement ESA.

Evaluation:

While evaluating the search engine, we need to test the accuracy, robustness, quality, and performance (memory and time complexities, etc) of the model to comment on the search engine.

- We can either compile or separate a part of the dataset and use it as test cases to benchmark the model
- Manually checking the quality (similarity) of the retrieved information and their ranking would also aid in judging the model
- Monitoring the time taken and memory used to pre-process the corpus, process the queries, and return the results would indicate the performance (Speed, etc can be compared with peer’s models, etc)

By incorporating the above, we would like to improve the current performance of the search engine and explore other methods and draw inspiration from the existing resources. We would also like to experiment with the above methods to benchmark the performances and ideate on new methods, if possible; and implement the practical & best-performing model.