CS6370: Natural Language Processing
Project Proposal

Team details:

Team number: **16**

Team members:

1) N Kausik (CS21M037)
2) Karthikeyan S (CS21M028)
3) Kavya Rengaraj (CS21Z018)

1. What limitation(s) of the Vector Space Model you are trying to address?

   Answer:

- **High dimensionality:**
  - If corpus has large documents with many words, the vocabulary size will be huge and hence the dimensionality of the vector space will be large
  - Hence TF-IDF values will be very low
  - Further, during calculation of cosine similarity, dot products of TF-IDF values will be very low and this produces low similarity always
  - Any operations on these vectors will be computationally expensive (Dot products, etc)
- **Word Order:**
  - We consider only frequencies of terms and hence information contained in the orders of words in sentences and orders of sentences in the documents are lost.
- **Synonymy:**
  - Truly relevant documents sometimes can have synonyms of a word and hence our system will not predict them as relevant
  - Eg. Query: Awesome **movie**
    Relevant Doc: The **film** was great.
    Here, movie and film are synonyms and hence the document is relevant but our system might not retrieve it since there are no matching words in the query and document.
  - This causes the system to have low recall since it does not retrieve some relevant documents (TP is low).
- **Polysemy:**
  - Some irrelevant documents are retrieved due to false positive matches of words in documents that carry a different sense (contextual meaning) from the query.
  - Query : Virus affecting hotel servers
  - Retrieved docs : The server crashed because of a **virus**
    Variola **virus** causes smallpox
  - This causes the system to have low precision since it retrieves some documents which are not relevant (TP is low but FP is higher)

- **Human Error:**
  - The retrieved documents given by the system depend on how the query is being entered by the user

    - If there are any spelling errors or incomplete sentences, the system will not be able to give the correct relevant documents
    - Eg. Query: Mone ban
      Relevant Doc: Money is stored in the bank.
      This relevant document will not be retrieved as the words don't match due to the spelling errors.
- **Title:** Titles of documents are not considered and hence their information is lost

2. State your hypothesis for addressing the above limitation(s). Describe how you would realize the above hypotheses in your search engine.
   Answer:
- **High dimensionality:**
    - We will use LSA or other methods to reduce the dimensionality while preserving enough information
    - Embeddings like Word2Vec and BERT embeddings will be tried instead of TF-IDF
    - **Realization:** The tf-idf calculation step will be replaced with word2vec / bert embeddings in informationRetrieval.py, LSA will be used after to reduce dimensions
- **Word Order:**
    - We will be using n-gram (mostly n=2) for preserving the word order and co-occurrence information
    - **Realization:** In the tokenization step (tokenizer.py), we will add a n-gram tokenizer function to the Tokenizer class and implement the n-gram tokenizer using nltk library
- **Synonymy:** Embeddings like Word2Vec/ BERT will address this as synonymous word vectors will be similar. Also LSA helps in capturing this information (2 words meaning same thing).
- **Polysemy:** This is partially addressed by using LSA and n-gram (since neighborhood information is captured)
- **Human Error:**
    - We will use a spell checker on the query to prevent misspelled words from giving wrong results
    - **Realization:** In the lemmatization step (inflectionReduction.py), we will add a spell checker and corrector (from library). Hence all queries and documents will be spelling checked and corrected before they are tokenized
- **Title:** We will add the terms in the title also to the document and weight their vector values by a K > 1 (around 2 or 3)

3. Describe how you would evaluate your system.
   Answer: To evaluate the system,
   We will continue to use the Cranfield dataset.
   We will use MAP @ k and Mean nDCG measures since they provide a better evaluation metric than others. Since in qrels.json, we are provided the relevant scores of the relevant documents, we can make use of Mean nDCG to get a good evaluation of our system. MAP @ k is also calculated for comparing it with nDCG.