INTRODUCTION

In this part, we are implementing an extensive language model based on Natural Language Processing for processing query and generating paragraphs.

Firstly, we are preprocessing the query, which includes tokenizing and removing stop words from it. We are also preprocessing the corpus, by giving suitable tags to each paragraph. Then according to query, we are giving points to each paragraph and filtering out most relevant paragraphs.

IMPLEMENTATION DETAILS

Our model primarily works on 4 helper functions:

1. Tokenize_query
2. Interpreter
3. Insert_sentence
4. Parascorer

We are storing a list of common words (stop words) in an AVL tree.

We are also storing a list of synonym of words relevant to the corpus in an AVL tree.

Description of helper functions:

1. Tokenize_query :
   Firstly, we are removing the common words from the query. We are also tagging queries as "who", "when" or "other" type.

2. Interpreter:
   This method processes the tokenized query obtained from the previous function to return a list of key-id pairs and also information about the period if the query is of "when" type. The time period info is extracted using another synonym table of time related words.
   When the query is not of "when" or "who" type, words like "Gandhi" and "Mahatma" are marked as common words and removed from the token list.
   The list of key-id pair stores the words as key and assigns an id to them. A proper word (a word without synonyms) is assigned an id -1. The improper words (words with synonyms) are assigned an integer which identifies its meanings.

3. Insert_sentence:
   In this function we are analysing every word of sentence inserted and doing some preprocessing which include giving tags to the para according to the time related words and synonyms. We are also classifying words as proper (synonym-less) or improper (words with synonyms).
   Important phrases comprising of characterized by occurrence of 2 proper words in close proximity are stored.

   The last and second last sentences are accounted for continuation factor analysis.

4. Parascorer:

   A paragraph is assigned a score and a continuation factor on the basis of the basis of various parameters. If the query is of “when” type, the score of a time-relevant paragraph of the right period is incremented by 10 and of a time irrelevant paragraph is decremented by 10.

   For each unique proper word, score is incremented by 10, and for improper words, score is incremented by 6 for exact matches and 5.4 for synonyms.

   The last and second last sentence are analysed along with previous paragraph’s continuation factor to give the continuation factor of the current paragraph.