Part of speech tagging using Hidden Markov Model (HMM) – M2

The following three sentences (S1, S2 and S3) and their corresponding tag sequences (T1, T2 and T3) are given as training data for implementing HMM. **Answer questions A-D. [2X4=8M]**

S1: John cut the paper .	S2: Mary asked for a hair cut .	S3: Sharon asked for a pay cut .
T1: N V D N STOP	T2: N V I P N N STOP	T3: N V I P N N STOP

A. What will be size of Emission and Tag translation matrices?

Note: Include the start and the stop symbols and assume we are working with a bigram model.

Emission Matrix is of size 13X8 and Tag translation matrix would be of size 8X8

B. What will be the Emission Probability P(asked | V) ? 2/3 or 0.667

C. What will be the Tag Translation Probability P(I|V) ? 2/3 or 0.667

D. If a brute force approach is employed to find the tags for the test sentence “***I had a deep cut.***”, how many possible tag sequences need to be evaluated? 5^{5} or 3125

1. You are asked to design an IR system that allows wildcard characters in between a word. Name the appropriate index(s) that can handle such a query. **[1 M]**

   The normal B-tree can handle trailing wildcard queries and Reverse B-Tree handles leading wildcard queries. Since the wildcard is in between a word we need to use both B-tree and Reverse B-Tree.

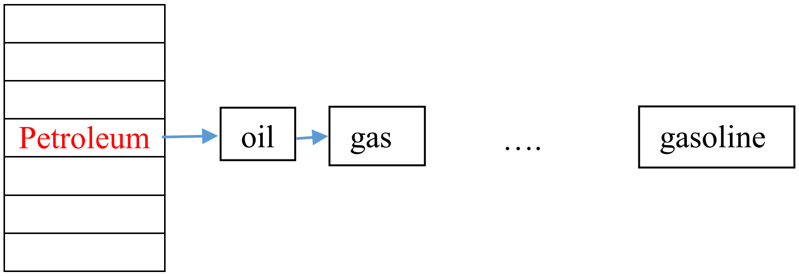
2. A user is searching for all documents that contain the term ***petroleum***. Since the terms like ***oil, gas, crude, barrels, gasoline*** etc. are also considered relevant you are asked to add an extra feature to the search engine that can handle the task. **[3+2=5 M]**

i. Briefly explain the two main approaches we discussed in this module to implement this feature.

This feature can be implemented using equivalence classing and query expansion.

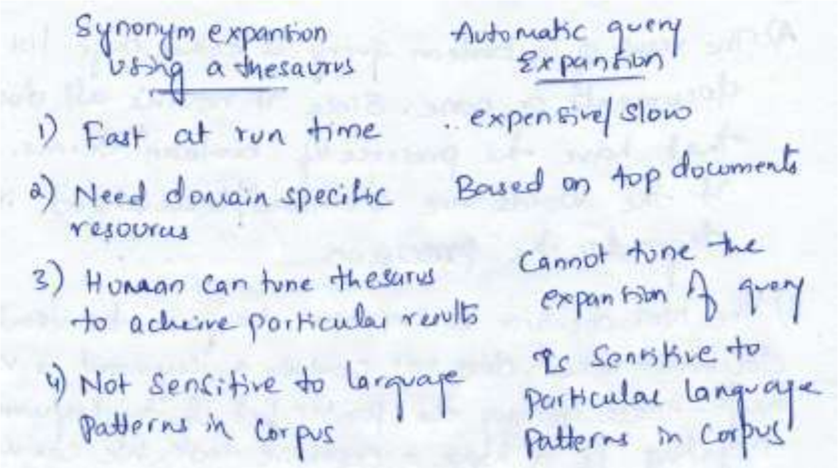
Equivalence classing: In this approach a manually constructed thesauri is used to equate all following tokens ***oil, gas, crude, barrels, gasoline, petroleum*** into a single term and let's say petrol. Hence the inverted index will have a term petrol and the posting list would contain all documents having any of the following words ***oil, gas, crude, barrels, gasoline, petroleum.*** Hence when the user enters petroleum as the query term this would be searching for the term petrol and hence it will return all documents having these variant words in the thesauri.

Query expansion: In this approach all the tokens would appear in the inverted index as different terms and a hash table would be maintained to find the variant terms.



When the user enters petroleum we look the has table and expand the query as petroleum or oil or gas or … or gasoline and fetch the posting for all the terms and return the merged result set to the user.

3. State the pros and cons of each approach.



Synonym expantion using a thesaurus	Automatic query Expansion
1) Fast at run time	.. expensive/ slow
2) Need domain specific resources	Based on top documents
3) Human can tune thesarus to acheive particular results	Cannot tune the expantion of query
4) Not sensitive to language patterns in corpus	Is Sensitive to particular language patterns in corpus