Lab Assignment 2

DSA460 Data Mining
CIS 492/593 Data Mining

Processing to Build TF-IDF based Document Vectors for Content Analysis

A document can be represented as **a Bag of Words** where a document is represented in a bag of thousand terms (words). To compare the similarity between a document and a user given query, each document in a collection of documents needs to be transformed to a document vector in the given user topics as features.

Construct a document vector for the given documents using the Lexicon based TF-IDF score function respect to the two set of the user query topics as below. The Topic terms are the following sets of keywords and phrases (bi-grams or tri-gram) as below.

Using the analytic method covered in class to measure Similarity between each document to a user query Topics, Find the State of Union Address that addressed the following topics:

1. Find which State of Union Address of US President addressed:
   Freedom, **Freedom of Speech, Freedom of Press**

2. Find which the State of Union Address of US President addressed:
   Security, Peace, Reestablishment of peace, **Preservation of peace**

Use the input file in the Lab2 Section of the Class Webpage.
Those address texts in the input file are your documents to process to count frequency and document frequency of each topic word to construct document vectors as below.

	Freedom	**Speech**	**Press**	**Freedom_of_Speech**	**Freedom_of_Press**
doc1					
doc2					
doc3					
doc4					
doc5					
doc6					

Notes:

1. We do not want to count any subexpressions that are a part of another words.

"Spin" should not be counted as "pin"

2. However, No case sensitive: Insert, insert, INSERT, insert are all counted as a same word.
3. The words from a same stem are counted as a same word.
   For example, program, programming, programed, programmable are all counted as "program".
   You can directly add OR conditions with all the variations of the words that are from a same stem to count all as a same word.

4. We want to count for a phrase word (bi-grams or tri-grams) by counting occurring of 'data mining', for example, when 'data' immediately followed by 'mining'.

For a real-time search engine, this is usually done by adding a discovered bi-gram or tri-gram in the term dictionary of the Inverted Index as a single term, for example, 'data mining' is added as a single term with 'data_mining' in the term dictionary with its frequency.

For this lab, you may omit to build an Inverted Index (your term dictionary) ahead with each unique term frequency and its document frequency to construct the document vectors for the given Topics.

However, Phrase (bi-gram or tri-gram) term handling and IDF is highly recommended for a full credit to add based on the weight score definition of TF and IDF for the collection of the documents for this Lab. You can count the document frequency (even manually) to find IDF per the given 5 topics without building Inverted Index. Make sure to remove all the stop words.

Common NLP Preprocessing Procedures for Text Analysis

Minimum Requirements:

1. Remove all the special symbols like punctuation mark, question mark using the character deletion step of translate
2. Remove all stop words (Search for Stop word Lists or Python or R Libs)
3. Do Stemming to Reduce inflected (or sometimes derived) words to their word stem.
4. Convert uppercase to lowercase

For more accurate Text Preprocessing, see Lab2 Section.

- You can use or adopt any online word count program for this Lab if you need.
  For example, import java.util.StringTokenizer or Python Text Preprocessing library
- You can make any assumptions to simplify the program.
- Briefly make notes on these in your report.

Extra Credit:
Build a Document Vector Matrix for the Entire State Union Address Collection. See Lab2 section for the Input file.

Building Inverted Index for the Entire State Union Addresses Collection will be an Extra Credit as well.

Submission:

Report:

Show each step in screen captures of the process and the intermediate output in each step) and explain BRIEFLY (and clearly) about what you did and why.

Make outputs with each Screenshot of your scripts (codes) and output in. doc file and explain each step briefly. You will lose points if you don’t show and explain each step properly.

Submission:

1) Submit your zip files with all your input and output files, source code files and your report (in .doc file) on Blackboard
2) Your report should show all your source codes with explaining each data processing step with your source codes, each intermediate output in each screen capture.