IR Assignment 1 – Design Document

Maximize-Min-Frequency Selection Model vs The Vector Space Model

Team:

Sanjiv Yelthimar Shenoy - 2017A7PS0224H

Tulika Jha – 2017AAPS0364H

Manish Kumar Bachhu – 2017A7PS0036H

Sundeep Kumar Ammisetti – 2017A7PS1218H

Data Set Description

The data set consists of lyrics of 210519 songs in the bag of words format. The songs are identified by a unique alphanumeric ID number.

Search Engine Working

Given a query, we output the top 10 documents relevant for that query using our model.

Data Set Preprocessing

1. The data set contains words which are stemmed using the Porter’s stemming algorithm.
2. The word count of all words in a document along with the document IDs is stored in a dictionary. This is loaded from a file at run time.

Search Engine Running

1. Given a query, we first run the Porter’s stemming algorithm on it to get a list of stemmed words.
2. This list is then used to find all documents containing these words. This is done in python using set intersection.
3. Then the algorithm is applied to order these documents and return the results.
4. The algorithm gives priority to documents which contain all query words.
5. In case we don’t find adequate query documents (less than 10) containing all the query words, then we use Inverse Document Frequency to pick subsets of the query as new queries.
6. These are then run again and ordered till we reach 10 documents.
7. Once we get the top 10 documents, we output the results.

Algorithm Working and Comparison with the Vector Space Model:

Example: Table showing word count (term frequency) of 4 query words in 4 Documents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Word1 | Word2 | Word3 | Word4 |
| Document1 | 1 | 2 | 4 | 5 |
| Document2 | 3 | 3 | 1 | 4 |
| Document3 | 2 | 3 | 5 | 3 |
| Document4 | 3 | 4 | 3 | 4 |

So we sort each document list by word count.

Then pick the largest minimum element in among the lists. The list containing that gets highest priority.

If 2 lists, we are comparing have the same minimum element we check for the next minimum element.

Sorted order:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Document1 | 1 | 2 | 4 | 5 |
| Document2 | 1 | 3 | 3 | 4 |
| Document3 | 2 | 3 | 3 | 5 |
| Document4 | 3 | 3 | 4 | 4 |

Therefore, document priority is Doc4 > Doc3 > Doc2 > Doc1

Interestingly the vector space model also gives the same ordering for these set of documents.

Now say we add another Document to compare the difference.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Word1 | Word2 | Word3 | Word4 |
| Document1 | 1 | 2 | 4 | 5 |
| Document2 | 3 | 3 | 1 | 4 |
| Document3 | 2 | 3 | 5 | 3 |
| Document4 | 3 | 4 | 3 | 4 |
| Document5 | 3 | 4 | 3 | 10 |

Intuitively we may give preference to document 5 over Document 4.

Our model gives the following order:

Doc5 > Doc4 > Doc 3 > Doc 2 > Doc1

However, the vector space model gives the following order:

Doc4 > Doc3 > Doc2 > Doc1 > Doc5

Inverse Document Frequency is used in the Vector Space Model to account for rarity of the words. The idea being the document containing the rare word should be given higher preference over documents not containing the rare word.

In our model, therefore when we search through documents containing all query words, IDF is irrelevant.

However, if we don’t find at least 10 documents containing all the query words, then the IDF value becomes relevant. We then pick subset of query words prioritized by IDF values.

IfDF formula – log10(N/df) where N is total number of documents, df is document frequency.

Eg: Suppose IDF values for query words are W1-6, W2-5, W3-5, W4-2, W5-1

Then order of sub query will be:

W1, W2, W3, W4

W1, W2, W3, W5

W1, W2, W3

W1, W2, W4, W5

W1, W3, W4, W5

W1, W2, W4

and so on...

Reasoning behind the Approach:

In the vector space, model the query generally contains each word exactly once.

So what the vector space model tries to do is to sort documents based on how close they are to the 1:1 ratio among all query words. It is also biased based on the frequency and number of no-query words.

A simple example makes it easier to understand:

Here we assume query has 3 words and both documents have these words but with different word counts

|  |  |  |  |
| --- | --- | --- | --- |
|  | Word1 | Word2 | Word3 |
| Document1 | 1 | 1 | 1 |
| Document2 | 50 | 51 | 52 |

Vector Space Model Intuition:

Document 1 is of higher priority as it exactly follows the 1:1:1 ratio of the query and hence has angle 0

Maximize-Min-Frequency Selection Model Intuition

Document 2 has higher min frequency 50. That means each query word must have occurred at least 50 times in this document. Hence it is given higher priority.