Introduction

The Vector Space Model provides a way of searching relevant documents in a large database of documents and return a set of documents that match a query string of words.

Method:

- Represent the documents in a *term-document* matrix where the rows are the “dictionary terms” and columns are the “documents”.
- Represent the query as a vector of dictionary terms.
- Find the relevant documents by calculating the cosine of the angle between the “document vectors” and the “query vector” and selecting the set that has cosine values above a threshold.

Low Rank Approximation:

- Large number of entries in the term-document matrix effects queries and produces inaccurate results.

- So we need to find a low rank approximation of term-document matrix. Singular-Valued Decomposition is used to produce the low rank matrix.

- The query vector is also transformed to the low rank space.

S V D for Low Rank Approximation:

$A \in \mathbf{R}^{m \times n}$,is the term-document matrix

$$A = U\Sigma V^T$$

Where, Σ is the diagonal matrix with singular values $\sigma_1..\sigma_r$,

where r is the rank of the matrix

$$\mathbf{Rank}(A) = r$$

Get a low rank approximation A_k with $\mathbf{Rank}(A_k) = k$ for A

Set $\sigma_i = 0 \; for \; i > k$ such that it minimizes $||A - A_k||_F$

Now, we can express the SVD with the lower rank Σ_k, as

$$A_k = U^{m \times k} \Sigma_k^{k \times k} V^{T^{k \times n}}$$

Thus we have the low rank matrix as

$$A_k \in \mathbf{R}^{m \times n}$$

If $A_{k(i,j)} = 0 \;\; \forall i > l \; and \; \forall j \in 1 \to n$, we can remove those rows from A_k and represent it as a $l \times n$ matrix

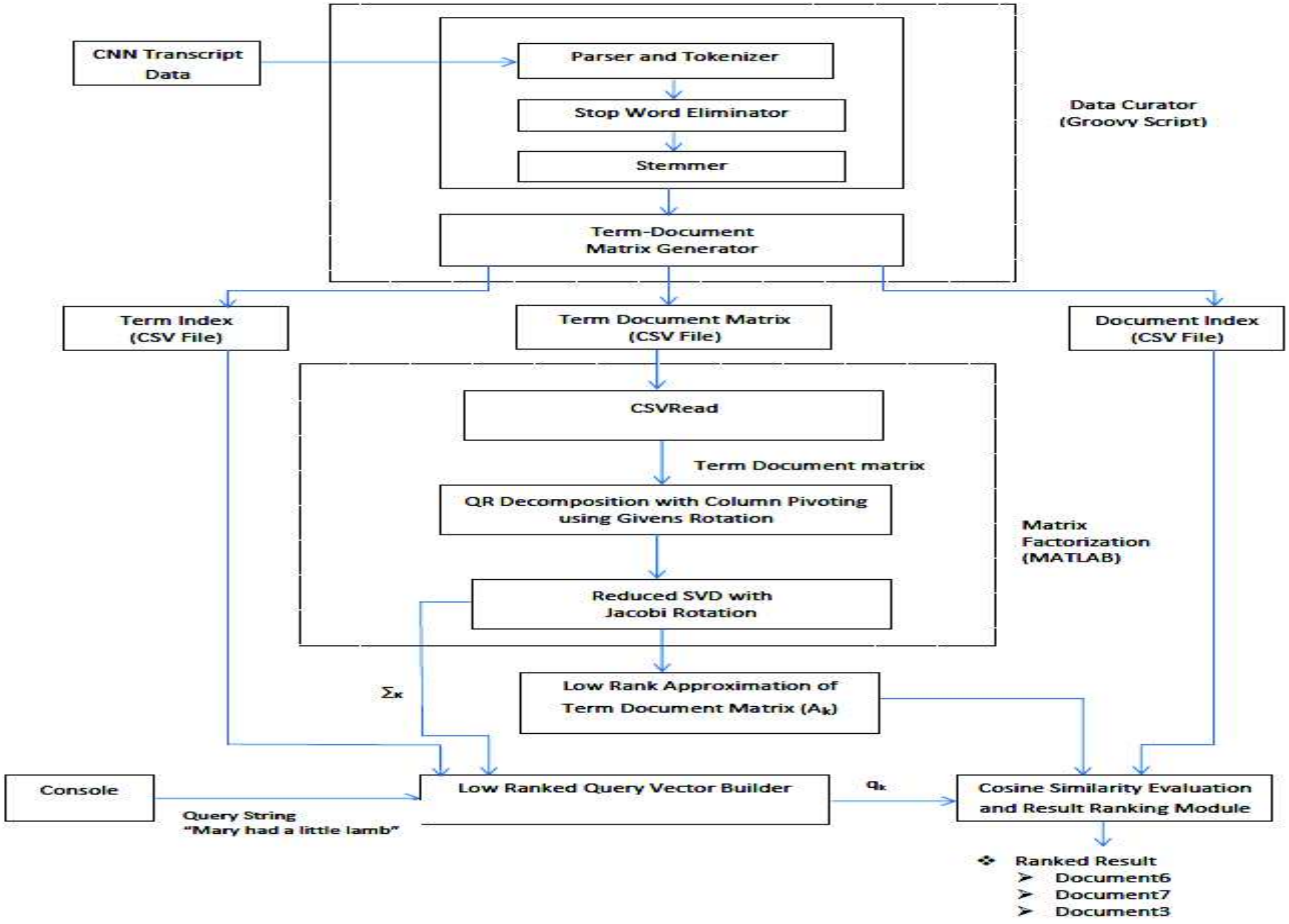
The query vector Q can be mapped to the new space using the following transformation

$$q_k = \Sigma_k{}^{-1} U_k^T Q$$

Then the proximity of the query to each of the document can be got from the $\cos\theta$ of the angle between them as

$$\cos\theta = \frac{q_k^T . A_{k(j)}}{||q_k^T|| ||A_{k(j)}||} \; where \; A_{k(j)} \text{ is each column vector in } A_k$$

Experimental Setup



CNN Transcript Data

Parser and Tokenizer

Stop Word Eliminator

Stemmer

Term-Document Matrix Generator

Data Curator (Groovy Script)

Term Index (CSV File)

Term Document Matrix (CSV File)

Document Index (CSV File)

CSVRead

Term Document matrix

QR Decomposition with Column Pivoting using Givens Rotation

Reduced SVD with Jacobi Rotation

Matrix Factorization (MATLAB)

Low Rank Approximation of Term Document Matrix (A_k)

Σ_K

Console

Query String
"Mary had a little lamb"

Low Ranked Query Vector Builder

q_k

Cosine Similarity Evaluation and Result Ranking Module

- Ranked Result
  - Document6
  - Document7
  - Document3

Results:

- The document term matrix was rank deficient , and of rank 81
- 5 low rank approximations of the matrix of rank, 8,10,20,40,81 where saved to disk in different files
- The search query was executed against each of these saved matrices separately and results where compared

Results:

- The search results from the lower ranked matrices of 8,10,20 where generally more relevant and gave importance to the variation between the documents
- The search results from the higher ranked matrices of 40,80 where similar in most cases

Sample Results:

```
EDU>> vecquery

Enter the string to search (e.g. tripoli peopl hide gadhafi ) :

georg zimmerman claim fire defenc

Searching documents fro query string georg zimmerman claim fire defenc

===================================================

Searching through term doc matrix with rank 8

Closest matching DocNo=93 _document509.txt

===================================================

Searching through term doc matrix with rank 10

Closest matching DocNo=93 _document509.txt

===================================================

Searching through term doc matrix with rank 20

Closest matching DocNo=93 _document509.txt

===================================================

Searching through term doc matrix with rank 41

Closest matching DocNo=93 _document509.txt

===================================================

Searching through term doc matrix with rank 82

Closest matching DocNo=93 _document509.txt
```