Term by Document Matrix Basics

Dr. Goutam Chakraborty

Objectives

- Explain the term-by-document matrix and how it is constructed
- Discus how document and term similarities are calculated
- Define and explain frequency weights and global weights and how these are used
- Provide guidelines for selecting weights.

Term by Document Matrix

Document 1:

I am an avid fan of this sport book. I love this book.

Document 2:

This book is a must for athletes and sportsmen.

Document 3:

This book tells how to command the sport.

Term/Document	I	am	an	avid	fan	of	this	book	love	is	a	must	for	atheletes	and	sportsmen	tells	how	to	command	the	sport
Document 1	2	1	1	1	1	1	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1
Document 2	0	0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	0	0	0	0	0	0
Document 3	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Zipf's Law

- In a corpus of documents,
- $f \ \propto \frac{1}{r} \qquad or \ \ f \ . \ r = k$
- where f is the frequency of a word, r is the rank of the word, and k is a constant.

- This law and its variants (such as power laws) have implications for figuring out importance of terms in a corpus.

Implications of Zipf's Law for Text Analytics

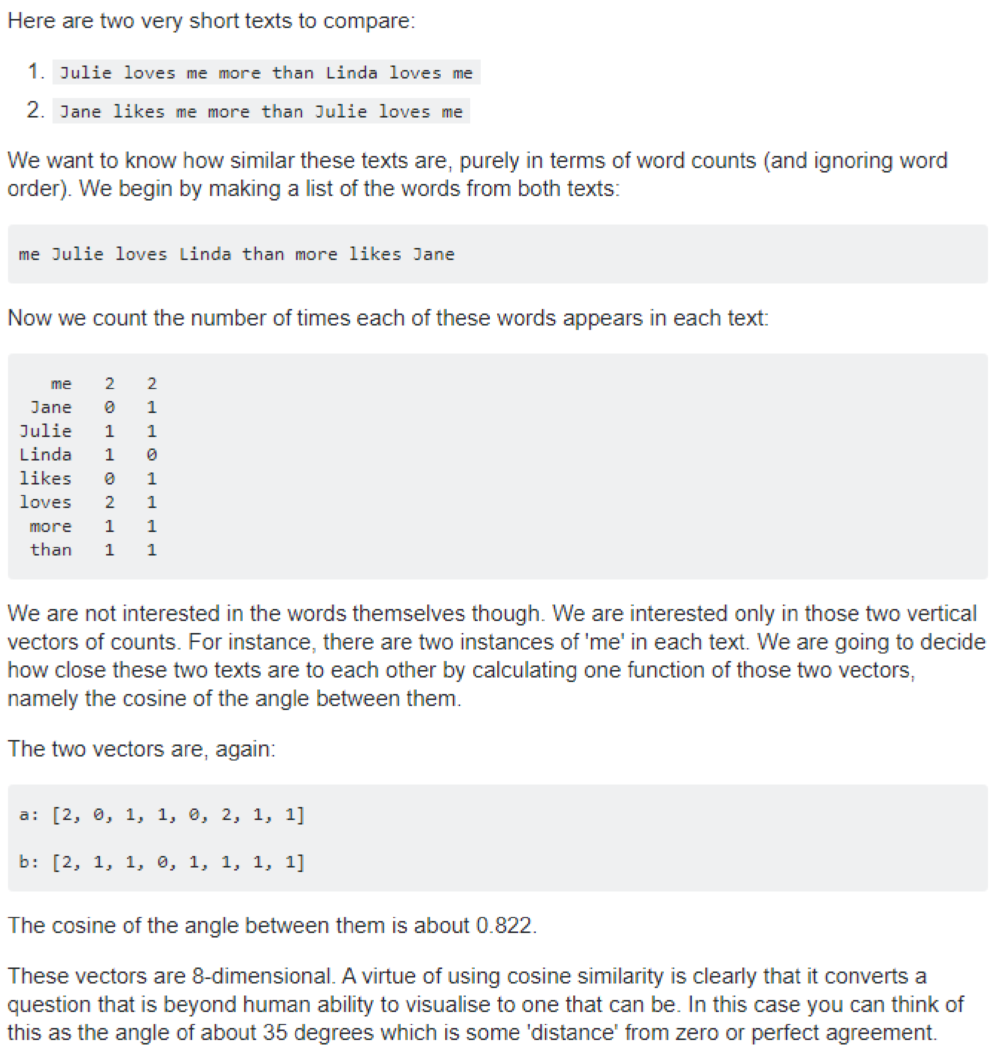
- A typical corpus contains the following:
  - A large number of infrequent terms
  - An average number of average frequency terms
  - A small number of high frequency terms
- Highly frequent terms in a corpus do not help distinguish one document from another.
  - Examples are words such as *the*, *and*, *an*, *or*, and *of*.
  - Often these are used as **stop words**
- Typically, terms with average frequency turn out to be the most informative.

Measuring Document Similarity

- Several approaches can be used to judge if ***two documents*** are similar to each other using the entries in the term-by-document matrix. These include the following:

1. Using descriptive statistics such as word count, sentence count, and paragraph count
    - Typically, normalized counts (relative frequency within a document) are more useful than raw counts.
2. Using a common set of terms to create a contingency (cross-tab) table and the calculate association metrics (such as Phi) to measure similarity between two documents
3. Using distance-based metric (such as Euclidean or *cosine*) to create a dissimilarity measure

More on Cosine Similarity



Here are two very short texts to compare:

1. `Julie loves me more than Linda loves me`
2. `Jane likes me more than Julie loves me`

We want to know how similar these texts are, purely in terms of word counts (and ignoring word order). We begin by making a list of the words from both texts:

```
me Julie loves Linda than more likes Jane
```

Now we count the number of times each of these words appears in each text:

```
   me   2   2
 Jane   0   1
Julie   1   1
Linda   1   0
likes   0   1
loves   2   1
 more   1   1
 than   1   1
```

We are not interested in the words themselves though. We are interested only in those two vertical vectors of counts. For instance, there are two instances of 'me' in each text. We are going to decide how close these two texts are to each other by calculating one function of those two vectors, namely the cosine of the angle between them.

The two vectors are, again:

```
a: [2, 0, 1, 1, 0, 2, 1, 1]

b: [2, 1, 1, 0, 1, 1, 1, 1]
```

The cosine of the angle between them is about 0.822.

These vectors are 8-dimensional. A virtue of using cosine similarity is clearly that it converts a question that is beyond human ability to visualise to one that can be. In this case you can think of this as the angle of about 35 degrees which is some 'distance' from zero or perfect agreement.

Source: https://stackoverflow.com/questions/1746501/can-someone-give-an-example-of-cosine-similarity-in-a-very-simple-graphical-wa

Measuring Term Similarity

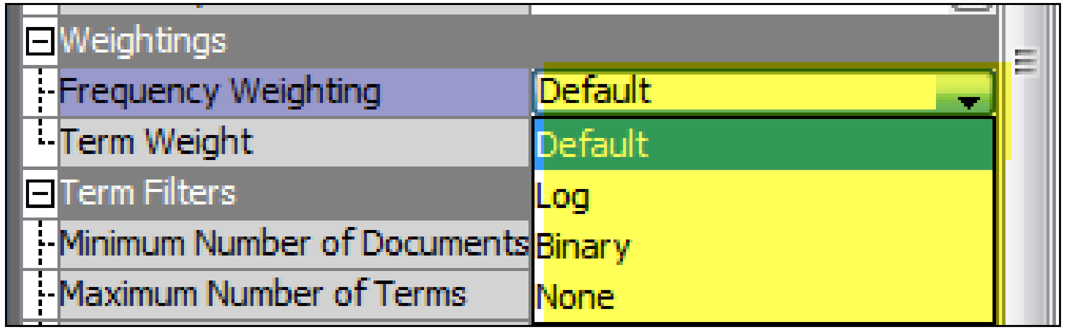
- Similarity between ***two terms across all documents*** in a corpus can be measured using the association metric or the distance metric.
  - Phi coefficient for 2-terms-by-*n*-documents contingency table
  - Euclidean or Cosine distance for two vectors of dimension *n*

From Raw Counts to Weighted Counts

- Often the raw counts are not good discriminators between documents. These are typically transformed using various weighting schemes. There are two broad weighting schemes:
  - Frequency weighting (local weights)
  - Term weighting (global weights)

Frequency Weighting (Local Weights)

- Let f_{ij} be the raw frequency of the ith term in the jth document. Consider a function $g(.)$ to transform the raw frequency, $g(f_{ij})$ and, let w_i be the weight of the ith term. The weighted frequency of the ith term for jth document in a term-by-document matrix is given by, $g(f_{ij}) * \ w_i$
- SAS EM uses three different options:
  - Log: $g(f_{ij}) = \log(f_{ij} + 1)$
  - Binary: $g(\ f_{ij}) = 1$, if a term is present in a document and $g(\ f_{ij}) = 0$, if a term is absent in a document
  - None: $g(\ f_{ij}) = \ f_{ij}$

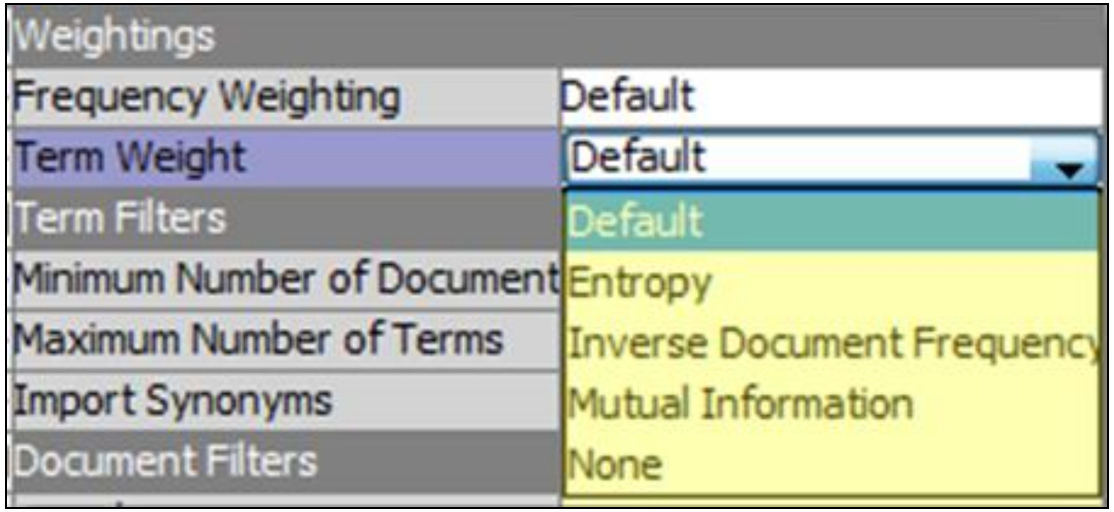


⊟ Weightings
- Frequency Weighting: Default
- Term Weight

⊟ Term Filters
- Minimum Number of Documents
- Maximum Number of Terms

Term Weights (Global Weights)

- These weights are used to adjust for both *document size* and *term distribution* across documents.
- Four options available in SAS EM for these weights:
  - Entropy
  - Inverse Document Frequency (IDF)
  - Mutual Information
  - None



Weightings	
Frequency Weighting	Default
Term Weight	Default
Term Filters	Default
Minimum Number of Document	Entropy
Maximum Number of Terms	Inverse Document Frequency
Import Synonyms	Mutual Information
Document Filters	None

More on Entropy

- The formula for calculating entropy weights is

- $w_i = \mathbf{1} + \sum \boldsymbol{j} \ \frac{(f_{ij}/g_i).\boldsymbol{log_2}\,(f_{ij}/g_i)}{\boldsymbol{log_2}\,(\boldsymbol{n})}$

- where f_{ij} is the raw frequency of the ith term in the jth document, g_i is the number of times that the term ‘i’ appears in the document collection, and $\boldsymbol{n}$ is the number of documents in the collection.

More on IDF

- A term that appears infrequently is considered more important and is given a higher score, whereas a term with high frequency of appearance is considered less important and gets a lower score.

- $w_i = \boldsymbol{log_2}\,(\frac{\boldsymbol{n}}{\boldsymbol{df_i}}) + \ \mathbf{1}$

- where, $\boldsymbol{df_i}$ is the document frequency or the number of documents that contain the ith term and $\boldsymbol{n}$ is the number of documents in the collection.

More on Mutual Information

- This weight is defined as
- $w_i = \boldsymbol{max}_{C_k}\left[\boldsymbol{log}\left(\frac{\boldsymbol{P(t_i\,,C_k)}}{\boldsymbol{P(t_i)\,P(C_k)}}\right)\right]$
- where, $\boldsymbol{P(t_i)}$ is the proportion of documents that contain the term $\boldsymbol{t_i}$, $\boldsymbol{P(C_k)}$ is the proportion of documents that belong to category$\boldsymbol{C_k}$, and $\boldsymbol{P(t_i\,,C_k)}$ is the proportion of documents that contain the term $\boldsymbol{t_i}$ and belong to category $\boldsymbol{C_k}$. Log(.) is taken to be 0 if $\boldsymbol{P(t_i\,,C_k)=0}$ or $\boldsymbol{P(C_k)=0.}$

Which Is the Best Weight?

- In general, entropy and IDF are most widely used in text analytics applications, with entropy being more effective for smaller documents and IDF for larger documents.
- In practice, you are advised to try all methods and then compare your results.

Document Search and Retrieval in SAS EM

- The Interactive Filter Viewer enables you to search for documents in the corpus based on search expressions.
- A search expression can be a single term or a list of terms.
  - Documents that match at least one of the terms are returned.
  - The search results contain a relevance score for each document that indicates how well each document matches the search expression.

How Does Interactive Search Work?

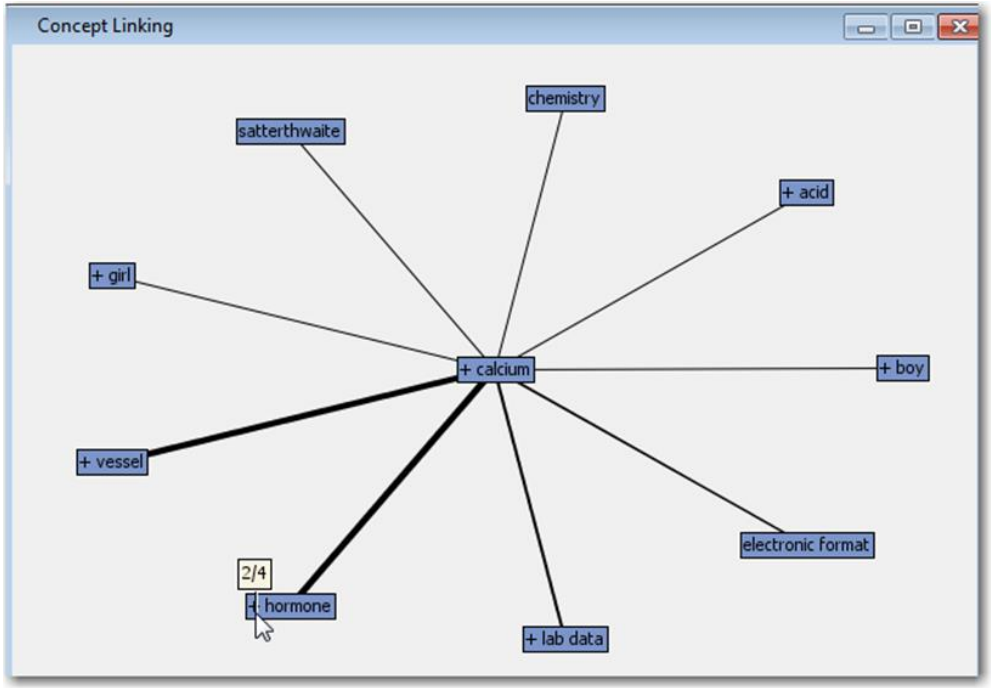
- The matching documents are retrieved using the vector space model. In the vector space model:
  - All documents and search expressions are represented as vectors in the term space. The vector components are the term weights for the terms in each document.
  - A similarity measure between two documents d1 and d2 is calculated using dot product or (*cosine* similarity) of the two vectors V1 and V2.
- $similarity\,(d1, d2) = \dfrac{V_1 \,.\, V_2}{|V_1|\ |V_2|}$
  - Euclidean length of the document is, $|V_i|\ = \sqrt{\sum V_i^2}$
- Hence, considering the unit vector $v_i = \frac{V_i}{|V_i|}$,
- $similarity\,(d1, d2) = \quad v_1 \,.\, v_2$

Enhancing Search in Interactive Filter

- **+term**
  returns only documents that include the term.
- **-term**
  returns only documents that do ***not*** include the term.
- **"text string"**
  returns only documents that include the quoted text.
- **string1*string2**
  returns only documents that include a term that begins with string1, ends with string2, and has text in between.
- **>#term**
  returns only documents that include term or any of the synonyms that have been assigned to the term.

Concept Links

- Concept links help in understanding the relationship between words (terms) based on the co-occurrence of words (terms) in the documents.



- The term **calcium** is more strongly associated with the terms **vessel** and **hormone** than with the other words.