An Introduction to Latent Semantic Analysis

Pat Reidy

Introduction and Motivation

The question of knowledge induction, i.e. how children are able to learn so much about, say, what words mean without any explicit instruction, is one that has vexed philosophers, linguistics, and psychologists alike. Indeed, inferring the vast amount of knowledge that children learn almost effortlessly from an apparently "impoverished stimulus" seems paradoxical.

The Latent Semantic Analysis model (Landauer & Dumais, 1997) is a theory for how meaning representations might be learned from encountering large samples of language without explicit directions as to how it is structured. To do this, LSA makes two assumptions about how the meaning of linguistic expressions is present in the distributional patterns of simple expressions (e.g words) within more complex expressions (e.g. sentences and paragraphs) viewed across many samples of language.

First, the meaning of sentences and larger documents is assumed to be the sum of the meanings of all the words occurring therein. So, the meaning of multi-word phrases is much more greatly determined by *which* words occur in the phrase, rather than *how* those words are configured. Likewise, the meaning of a certain word is a sort of average across all the documents in which it occurs.

Second, LSA assumes that the semantic associations between words is present not explicitly, but only latently in the large sample of language that humans come in contact with and eventually learn meanings from. For example, two synonyms may never occur together in the same passage, but should nonetheless have highly associated representations. Thus, LSA provides a way in which covert associations among words can be induced from a large scale examination of how these words co-occur with every other word in the language.

Dimensionality reduction is the mathematical tool from which arises knowledge induction about associations between words that are not overtly manifest in their co-distributions across linguistic samples. To explain, consider a words by passages matrix of the kind that LSA uses to infer semantic knowledge. In its unreduced state, the data corresponding to words and passages can be fit into any hyperdimensional space so long as that space does not have more dimensions than the lesser of the number of distinct words and the number of distinct passages. Choosing to represent the data in its maximal dimensionality possible will capture the distribution of words in passages exactly like how they have been encountered; however, in a reduced dimensional space the data will be fit as best as possible, but not perfectly. Thus, some data points will end up closer together or farther apart than the original co-occurrence distributions suggested on the surface. The way in which the position of data points end up nearer or farther from each other in the reduced space is exactly the mathematical correlate to how associations are induced from distributional data.

To bring home the discussion of dimensionality reduction and its effects on knowledge induction of associations between data points in a space we give a tangible example adapted from Landauer & Dumais (1997:214-5). Imagine a situation where three observers are looking out from a hill onto a village below. They want to describe the relative locations of three different landmarks—a statue, a fountain, and a chapel—to a listener who cannot see the scene below. Before hearing anything about the layout of the village, the listener could reason that the distance between two landmarks is in a three dimensional (euclidian) space since the topography of the landscape will be a north-south by east-west plane with possible hills and valleys.

So, the first observer tells the listener that the distance between the fountain and the statue is 100 m. The second observer tells the listener that the chapel is also 100 m from the statue. Even in just a north-south by east-west plane, the listener can accommodate these distances in any number of ways; however, if the third observer steps in and tells the listener that the village is located in a flat valley and that all three landmarks are situated on the same side of the same straight road, then the listener can infer the distance between the fountain and the chapel even without any explicit knowledge of the relation between them. By reducing the number of possible dimensions relevant to the distances between landmarks, the listener can infer that the fountain and the chapel are 200 m from each other with the statue located in the middle.

To draw an analogy to the LSA model at this point, the distances between landmarks given by the first two observers correspond to the word–document co-occurrence counts, while the description of the village's general layout given by the third observer is analogous to dimensionality reduction. Notice, however, that even after mentally placing each landmark on the same straight road, the listener can perfectly recover the distances between landmarks as reported by the observers. This is not how dimensionality reduction works in LSA. Instead, the original data points are usually modified slightly in fitting them into the reduced dimensional space.

An example of this loss of precision in recreating the original data in our example would be similar to if another observer told the listener that the distance between the fountain and the chapel is 150 m before the listener is told that all landmarks are located on the same street. In order to fit the three landmarks in a one-dimensional space, the listener must approximate the distances that she has been told and thus infers that the fountain and chapel are, say, 170 m from each other, and that the statue is in the middle, 85 m from both other landmarks. In this way, the dimensionality of the space serves as a hard constraint in which the data points must be fit as best as possible, which introduces some imprecision.

With a qualitative description of how LSA acquires semantic representations for words and passages behind us, we now briefly discuss how the successes of LSA could be interpreted, before turning to an explicit presentation of the algorithm used in the model. The most cautious interpretation of LSA's ability to correctly represent semantic knowledge is that it is merely an expedient way of determining distributional properties of words and phrases.

The other, more far-reaching interpretation is that LSA embodies the psychological processes of humans in acquiring and encoding semantic knowledge from nothing more than direct experience with a large portion of linguistic data. As we have seen, LSA assumes no prior constructs hard-coded in a person's mind that help in acquiring the meaning of words or phrases. Instead, LSA proposes only a general inductive process that parallels whaterver humans could actually be doing to acquire the semantic structure of linguistic expressions. Even then, LSA makes no claim that the algorithm that it employs is one and the same as the psychological process of humans—rather, just that its algorithm captures the spirit of our inferential strategy.

Mathematical Details

Before turning to the applications of LSA, we present the explicit computational algorithm used in LSA to learn semantic representations and infer associations among words. The algorithm takes as input an $m \times n$ words by documents matrix in which each entry a_{ij} is the local frequency of a given word i in a given document j. These raw word-document co-occurrence counts are first transformed to weight each word according to how informative it is in determining the meaning of the document. Next, the transformed matrix of weighted terms has its dimensions factored and reduced via the singular value decomposition (SVD). Finally, a new matrix is "reconstructed" from the reduced, decomposed matrix, and similarity measures are calculated from the reconstructed matrix.

Preprocessing

The $m \times n$ matrix of word-document co-occurrences inputed to the algorithm can be thought of as a rough analogy to memory representations of raw unanalyzed events. Thus, before inferring semantic relationships, the algorithm transforms the data so as to mimic how humans seem to learn associations (Landauer & Dumais, 1997).

First, to approximate the growth rate of simple learning, each entry a_{ij} is converted from its local frequency to its local weight where for each word i in document j,

$$weight_{ij}^{loc} = \log(freq_{ij}^{loc} + 1).$$

Next, the global weight of each word, which incorporates the information theoretic entropy of the word across documents, is calculated as,

$$weight_i^{glob} = \frac{1 + \sum_{j=1}^{n} p_{ij} \cdot \log(p_{ij})}{\log(n)},$$

In the equation for $weight_i^{glob}$, the quantity p_{ij} is defined as the local frequency of the word i divided by the global frequency of that word across all documents j,

$$p_{ij} = \frac{freq_{ij}^{loc}}{\sum_{j=1}^{n} freq_{ij}^{loc}}.$$

The weighted value of each term is, thus, its local weight divided by its global weight,

$$weight_{ij}^{term} = \frac{weight_{ij}^{loc}}{weight_i^{glob}}.$$

Notice that if a word occurs several times in a document, then its local weight will be relatively large, since this is directly related to local frequency. Also, if a word occurs frequently in several documents, then its global weight will be high as this is directly related with the word's entropy. Since term weight is directly related to local weight, but inversely related to global weight, it follows that a word will be highly weighted if it occurs frequently in a document relative to the other words in the document, but infrequently across all documents relative to its frequency in a given document. In effect, term weighting reduces the importance of words whose presence in a document is uninformative to determining the meaning of that document. This closely approximates how associations between items are better formed by their informativeness rather than just their co-occurrence.

Dimension Factorization and Reduction

Once the word-document matrix M has undergone the term-weighting transformation, it can be factored using singular value decomposition. It is a theorem of linear algebra that any $m \times n$ matrix M whose entries are real numbers can be decomposed into three matrices T, Σ, D^T such that

$$M = T\Sigma D^T.$$

In the above decomposition, T is an $m \times m$ matrix, and D^T is an $n \times n$ matrix, both of which have orthonormal columns. The columns of a matrix are called orthonormal if every column vector $\mathbf{v}$ in the matrix is a unit vector (i.e., $\mathbf{v} \cdot \mathbf{v} = 1$) and every two distinct columns $\mathbf{v}$ and $\mathbf{u}$ are orthogonal to each other (i.e., $\mathbf{v} \cdot \mathbf{u} = 0$).

Also, Σ is an $m \times n$ "diagonal" matrix of the form

$$\Sigma = \begin{bmatrix} D & 0 & \cdots \\ 0 & 0 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

where D is an $r \times r$ diagonal matrix for some r that does not exceed the smaller of m or n, the dimensions of the original matrix M. D is called a *diagonal matrix* because its only non-zero entries are on one of its main diagonals. The values on the main diagonal of D after applying the SVD are called the *singular values* of M, and they are ordered from greatest to least along the main diagonal of D.

The singular values represent the "dimensions of meaning" for words and passages in the language, and when D contains all the singular values of M, the original matrix M could be exactly reconstructed by multiplying the three matrices T, Σ, and D^T.

Representations and Similarities in Reduced Space

The dimensionality of the space of semantic representations can, however, be reduced by replacing some of the singular values with 0. By convention, singular values are zeroed-out from least to greatest. If we were to choose the s greatest number of singular values, such that s is less than the number of singular values that originally result from SVD, then we could construct the matrix M_s, which is the s-dimensional approximation to M with the least error,

$$M_s = T\Sigma_s D^T.$$

The number of dimensions in which to reconstruct the word-document co-occurrence matrix is freely determined by the user as a parameter external to the other workings of the algorithm. As we will see in the next section the performance of the model is significantly affected by the number of dimensions in which the matrix is reconstructed. Throughout the subsequent discussion of the model, it is important to keep in mind that while LSA is successful in obtaining semantic representations by fitting distributional data in lower-dimensional space, the choice of dimensions is not something that is determined by the model itself.

Representations of words and documents can likewise be obtained by multiplying their corresponding decompositions by the reduced space singular value matrix, Σ_s. Thus, the representations of words in s-space is given by

$$T_s = T\Sigma_s,$$

and the reduced representations of documents is

$$D_s^T = \Sigma_s D^T$$

Similarities between two vectors, $\mathbf{v_1}$ and $\mathbf{v_2}$, that each represent a word or document is calculated with the cosine of the angle, θ, between $\mathbf{v_1}$ and $\mathbf{v_2}$,

$$cos(\theta) = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\| \cdot \|\mathbf{v_2}\|},$$

where for a vector $\mathbf{v}$, $\|\mathbf{v}\|$ is the length of that vector and

$$\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}.$$

Note that the similarity between two vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ is directly related to the cosine of the angle between those vectors. If the cosine of the angle between two vectors is 1, then the two vectors are perfectly synonymous.

The similarities between two word vectors $\mathbf{w_i}$ and $\mathbf{w_j}$ in reduced space can all be collected into a single matrix by first right-multiplying the s-dimensional reconstructed matrix by its transpose.

$$\begin{aligned} M_s M_s^T &= T\Sigma_s D_s^T (T\Sigma_s D_s^T)^T \\ &= T\Sigma_s D_s^T D_s \Sigma_s^T T^T \\ &= T\Sigma_s \Sigma_s^T T^T \\ &= T_s T_s^T. \end{aligned}$$

From this matrix the cosine of the angle between two word-vectors $\mathbf{w_i}$ and $\mathbf{w_j}$ can be computed for the corresponding entry on the i^{th} row and the j^{th} column by dividing by $\|\mathbf{w_i}\| \cdot \|\mathbf{w_j}\|$.

Document-document similarities in reduced space are analogously calculated by first left-multiplying the s-dimensional reconstructed matrix by its transpose.

$$M_s^T M_s = D_s D_s^T$$

Again, the cosine of the angle between two document-vectors $\mathbf{d_i}$ and $\mathbf{d_j}$ can be computed for the corresponding entry on the i^{th} row and the j^{th} column by dividing by $\|\mathbf{d_i}\| \cdot \|\mathbf{d_j}\|$.

For word-document similarities in reduced space, an entry in the i^{th} row and j^{th} column in M_s is divided by $\|\mathbf{w_i'}\| \cdot \|\mathbf{d_j'}\|$, where $\mathbf{w_i'}$ and $\mathbf{d_j'}$ are respectively a word- and document-vector that has been translated into an intermediate space such that,

$$\mathbf{w_i'} = \mathbf{w_i}\sqrt{\Sigma}$$

and

$$\mathbf{d_j'} = \sqrt{\Sigma}\mathbf{d_j}.$$

Querying

LSA can also be used to determine the similarity of words or documents that were used to construct the concept space, with documents external to it. In order to make such a comparison, it is first necessary to transform the query document into a pseudo-document of the reduced space,

$$query = q^T U \Sigma_s^{-1},$$

where q^T is the term-weighted query.

Once the query has been transformed into the appropriate space, query-doc and query-word similarities can be calculated via cosine measures as described above (Martin & Berry, 2007).

Applications

In this section, we give a brief overview of the various domains to which LSA has been applied, ranging from simulations acurately modeling humans' conceptual knowledge, to practical applications that aid in information retrieval, essay grading, and education.

Learning Correct Word Meanings

At its core, LSA purportedly models how humans acquire semantic knowledge. The degree to which the inferences concerning word meaning drawn by LSA accord with the actual meanings of the words in the training corpus is measured with word similarity tests in which the similarity between words is measured by the cosine of the angle between them. In one LSA simulation, the angle between synonym and antonym pairs were found to have an average cosine of 0.18, a value 12 times higher than the same similarity measure between two unrelated words (Landauer, Foltz & Laham, 1998).

As an evaluation of the model's ability to learn proper word meaning representations, Landauer & Dumais (1997) tested how well the model performed on an 80-question synonym test where, given a test word, the model had to choose the most highly associated answer from a group of four choices. Decisions were made by selecting the answer choice that had the highest cosine value between it and the question word. The model performed on a par with proficient non-native English speakers.

This experiment also brought out the effect that the dimension of the reduced matrix can have on the model's performance. Given their training set, Landauer & Dumais found that performance peaked around 300 dimensions with a shard drop off in performance both above and below this dimensional threshhold.

The LSA model has also been shown to learn meanings in a quantitatively and qualitatively similar fashion to human learners. It is estimated that adolescent language learners acquire 7–15 words per day (Nagy & Herman, 1987), and most of these words are learned indirectly rather than their meanings being learned directly from the context in which it is read. To simulate semantic knowledge acquisition, Landauer and Dumais first trained their model on a corpus that was approximately equivalent in size to the amount of text that an older grade school student would have encountered. The model was then given paragraphs of text one at a time, and the amount of information learned directly and indirectly was calculated after the addition of each paragraph. Assuming that adolescent students read about 50 paragraphs per day (Anderson, Wilson & Fielding, 1988 and Carver, 1990), Landauer and Dumais found that an equivalent amount of additional training led to their model acquiring 10 more words, of which three fourths were learned indirectly; thus, the LSA model fully replicated the lexical semantic knowledge acquisition of human learners.

Subject-matter Comprehension

LSA has also been shown to successfully determine related concepts from passages on the same subject. Foltz, Britt & Perfetti (1996) performed one such simulation in which an LSA model was trained on documents about the history of the Panama canal and subsequently used to determine the relatedness of concepts that appeared in the texts. These judgments were made for 120 concept pairs and the results were compared to norms from both novices and experts on the Panama canal who completed the same task. The LSA model's results significantly correlated with both groups, but showed a stronger correlation with the experts.

Information Retrieval

The computational algorithm employed by LSA has also been used in a number of information retrieval tasks. For the task of retrieving relevant documents based on user-entered search terms, LSA gives a 16% improvement over algorithms that return documents based on only word-matching with the search term (Dumais, 1994). LSA has also successfully been used to recommend relevant papers for researchers to read based on a selection of papers that the researcher has read and liked (Foltz & Dumais, 1992).

Educational Applications

In the domain of educational testing, Landauer, Laham & Foltz (1998) have shown that LSA can be used to develop a number of methods for holistically grading essay responses. Each of the methods assigned scores to the test set of essays such that the scores given from the LSA methodology correlated with gold-standard scores assigned by trained human graders to the same degree as the intercorrelation among just the gold-standard scores themselves.

Finally, LSA can aid in selecting educational documents that match the reading level of the student so as to maximize learning. Research suggests that the student learns the most from a reading when the level of sophistication of the text is neither too complex nor too simple for the student to understand (Kintsch, 1994). LSA helps in this task, specifically, by comparing a short writing sample of the student's to educational documents ranging over various difficulties and choosing the best fit according to the documents' similarity as measured by LSA.

Exercise

Similarity of Article Titles: A Classic LSA Example

Landauer & Dumais (1997) demonstrate the ability of an LSA model to infer the similarities between titles of journal articles. They used nine titles in all, five of which came from articles about computing and four of which came from articles about mathematical graph theory. The nine titles used are shown below.

c1: *Human* machine *interface* for ABC *computer* application.
c2: A *survey* of *user* opinion of *computer system response time*.
c3: The *EPS user interface* management *system*.
c4: *System* and *human system* engineering testing of *EPS*.
c5: Relation of *user* perceived *response time* to error measurement.
m1: The generation of random, binary, ordered *trees*.
m2: The intersection *graph* of paths in *trees*.
m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering.
m4: *Graph minors*: A *survey*.

Notice that some of the words in the titles above are italicized and others are not. Before analyzing the data with their model, Landauer and Dumais manually "preprocessed" the word-document co-occurrence matrix to include only content words that occurred in at least two of the titles. In the titles above, the words that were included in their co-occurrence matrix are italicized, and all non-italicized words were thrown out.

This exercise is designed to gently introduce you to using the LSA model while at the same time give you an idea of how the correlations between words and documents can be changed by allowing more words to be analyzed by the model.

PART 1. Recreate Landauer and Dumais' results by using the `lsa.py` library provided for you. First, make a text document containing all the titles, in the order in which they appear above. In order to build the model with only the content words from above you'll have to modify the code slightly. Once you've made the proper modification, calculate the inter-title correlations for the data in a 2-dimensional space. Plot the inter-title similarity measures in a fashion that you find readable.

PART 2. Above we described a preprocessing procedure that is meant to weight each term for its importance to the meaning of the document. Given how small the data set is for this exercise, how effective do you think term weighting will be in recreating the manual content word filtering performed by Landauer & Dumais? Use the `lsa.py` library to calculate the title-title similarities in a 2-dimensional LSA model that preprocesses the corpus using term weighting, and then plot the similarities so that you can compare them with your plots from Part 1.

Further Reading

For the interested reader, more information on the motivation and spirit of Latent Semantic Analysis can be found in the original Landauer & Dumais (1997) paper. Additional mathematical clarifications are offered by Martin & Berry (2006), which was consulted frequently when writing the corresponding section of this tutorial. Landauer, Foltz & Laham (1998) offer an overview of even more simulations and applications of LSA.

Answer Key

Answer to *Similarity of Article Titles*

In order to build the model using only the content words used by Landauer and Dumais, you should modify the `type_find` function in the `lsa.py` library so that it returns only those content words. To do this, first make a copy of the library and name it `lsaTitles.py`. Then replace the `type_find` function therein with:

```
def type_find(file_name):
    return ['human', 'interface',
            'computer', 'user',
            'system', 'response',
            'time', 'eps',
            'survey', 'trees',
            'graph', 'minors'
           ]
```

Save your modifications to `lsaTitles.py`, and then make a document `titles.txt` that contains all nine titles. In the Python shell import the library and calculate the similarities between the titles in 2 dimensions by using the `doc_sim` function:

```
>>> from lsaTitles import *
>>> simsMP = doc_sim('titles.txt', 2)
```

Feel free to plot your results in any way that you find perspicuous. We give just one example of what to enter into the Python shell in order to plot the inter-title similarities relative to each title.

```
>>> from matplotlib import pyplot as plt
>>> xlabs = ['c1', 'c2', 'c3', 'c4',
             'c5', 'm1', 'm2', 'm3',
             'm4'
            ]
>>> plt.figure(1)
>>> for i in range(0, len(simsMP)):
...     qdrnt = int("2" + "5" + str(i+1))
...     plt.subplot(qdrnt)
...     plt.plot(simsMP[i], 'ro')
...     plt.axis([-1, 9, -1.2, 1.2])
...     plt.xticks(range(0, 9), xlabs)
...     plt.title(xlabs[i])
>>> plt.suptitle('Manual Preprocessing')
>>> plt.show()
```

This will produce the plot in Figure 1. Notice that each plot shows clear clustering among the titles according to their subject matter.

For Part 2, first build the model using term-weighting preprocessing by using the original LSA code in the `lsa.py` library. Open a new Python shell and then enter the following:

```
>>> from lsa import *
>>> simsTW = doc_sim('titles.txt', 2)
```

Then plot these new results in a similar fashion to how you did in Part 1. For the style of plot that we offered above, you would need to enter:

```
>>> from matplotlib import pyplot as plt
>>> xlabs = ['c1', 'c2', 'c3', 'c4',
             'c5', 'm1', 'm2', 'm3',
             'm4'
            ]
>>> plt.figure(1)
>>> for i in range(0, len(simsTW)):
...     qdrnt = int("2" + "5" + str(i+1))
...     plt.subplot(qdrnt)
...     plt.plot(simsTW[i], 'ro')
...     plt.axis([-1, 9, -1.2, 1.2])
...     plt.xticks(range(0, 9), xlabs)
...     plt.title(xlabs[i])
>>> plt.suptitle('Term-Weighting Preprocessing')
>>> plt.show()
```

The plots for inter-title similarities from an LSA model using term-weighting are shown in Figure 2. When we allow the model to filter the raw data on its own, we see that it is much more difficult for the model to infer the latent correlations between the related titles. The model still seems capable of assigning high correlations to titles that share the same subject matter; however, it is exceedingly generous in finding a similarity between titles from unrelated domains.

In Part 2 of this excercise, we kept the dimensions of the reduced concept space the same as in Part 1, but varied which words were in that space, which shows that the model's performance can be affected by more than just the choice of how many dimensions the concept

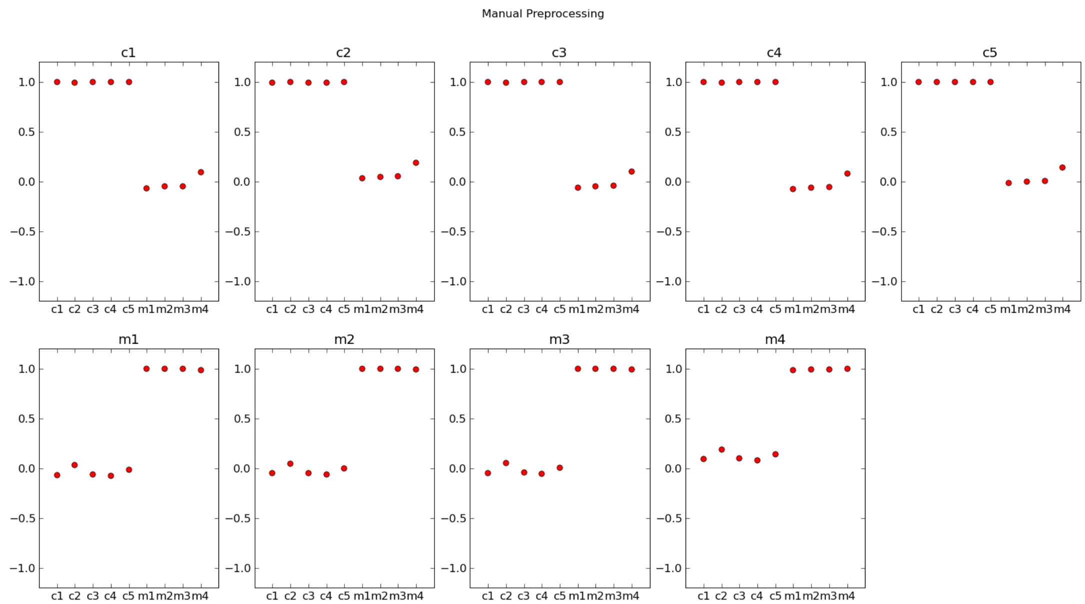

Figure 1: Inter-title similarities after manual preprocessing

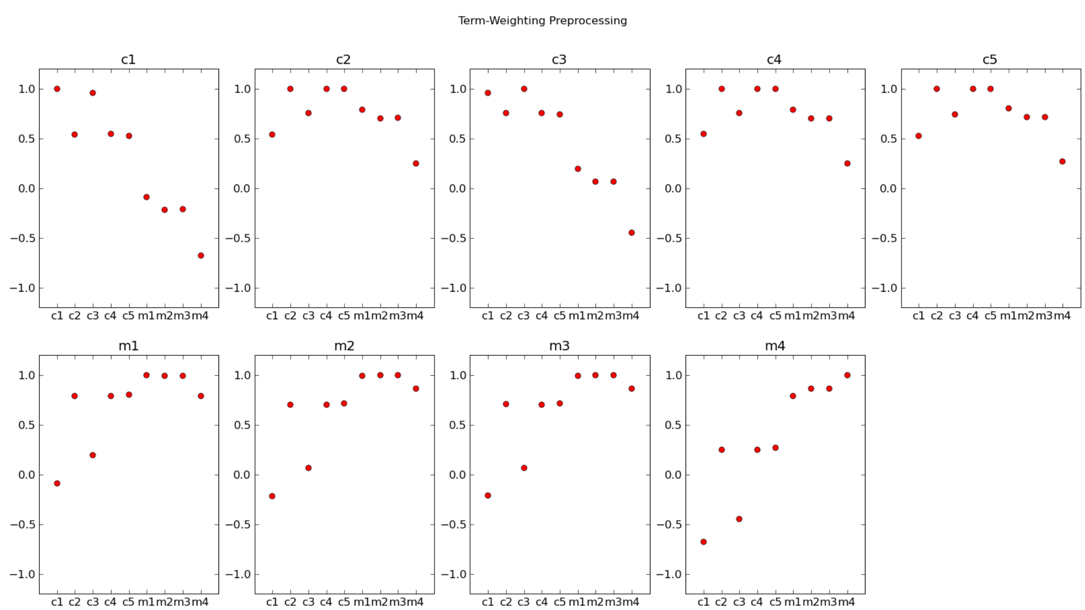

Figure 2: Inter-title similarities after term-weighting preprocessing

space is reduced to. The data set for this excercise is, thus, too sparse for the term-weighting preprocessing to have the optimal effect. Because of this, every word is treated as a content word, even content-less words like *the*, which explains why cross-group inter-title correlations are higher in Part 2 than in Part 1.

References

[1] Anderson, R.C., Wilson, P.T., & Fielding, L.G. 1998. Growth in reading and how children spend their time outside of school. *Reading Research Quarterly*, *23*(3), 285–303.

[2] Carver, R.P. 1990. *Reading rate: A review of research and theory.* San Diego, CA: Academic Press.

[3] Dumais, S.T. 1994. Latent semantic indexing (LSI) and TREC-2. In D. Harman (ed.), *The Third Text Retrieval Conference (TREC3)* (NIST Publication No. 500-225, pp. 219–230). Washington, DC: National Institute of Standards and Technology.

[4] Foltz, P.W., Britt, M.A., & Perfetti, C.A. 1996. Reasoning from multiple texts: An automatic analysis of readers' situation models. In G. Cottrell (ed.), *Proceedings of the* $18^t h$ *Annual Cognitive Science Conference*. Hillsdale, NJ: Lawrence Erlbaum Associates.

[5] Foltz, P.W. & Dumais, S.T. 1992. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, 35, 51–60.

[6] Kintsch, W. 1994. Text comprehension, memory, and learning. *American Psychologist*, 49, 294–303.

[7] Landauer, T.K. & Dumais, S.T. 1997. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review.* Vol. 104, No. 2, 211–240.

[8] Landauer, T.K., Foltz, P.W., & Laham, D. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25, 259–284.

[9] Martin, D. & Berry, M. 2007. Mathematical Foundations Behind Latent Semantic Analysis. In T. Landauer, D. McNamara, S. Dennis, W. Kintsch (eds.), Lawrence Erlbaum Associates, Inc.: Mahwah, NJ. pp. 35–55.

[10] Nagy, W.E. & Herman, P.A. 1987. Breadth and depth of vocabulary knowledge: Implications for acquisition and instruction. In M.C. McKeown & M.E. Curtis (eds.), *The nature of vocabulary acquisition* (pp. 19–35). Hillsdale, NJ: Erlbaum.