With LSA Size DOES Matter

Colin Layfield
University of Malta: Department of CIS
Faculty of ICT
Msida, Malta
colin.layfield@um.edu.mt
ShirWin Knowledge & Learning Systems
Edmonton, Canada
clayfield@shirwin.com

Abstract—Latent Semantic Analysis (LSA) is a technique from the field of Natural Language Processing that enables comparison of semantic similarities between documents using vector operations. This technique has been used in areas from Information Retrieval (IR) to the automated assessment of essays. One property used in document comparison is size. The general philosophy is that more text is better although few concrete examples or guidelines exist that demonstrate this. This paper shows, via a novel concrete example taken from real world data, that larger documents do imply more accurate semantic similarity comparisons.

Keywords-latent semantic analysis; LSA; automated essay assessment; document length; natural language processing; NLP

I. INTRODUCTION

Latent Semantic Analysis (LSA) is a statistical method for creating semantic spaces [1]. It was originally applied to the Information Retrieval domain (IR) [2] but has since been used in many other areas such as patent examination support [3], simulating clinical comprehension [4], predicting the learning that can take place from various texts [5], detecting plagiarism [6], and the automated assessment of essays [7] [8] [6] [9]. The research carried out at Shirwin Knowledge & Learning Systems was with respect to the automated essay assessment application of LSA using real student essay data gathered over a number of years from a course at the University of Alberta.

This paper is organized as follows; first we will introduce LSA and describe how the documents are transformed using this process. The mechanisms used for comparing documents will be described next as well as the relevance of the length of documents in the system. The methods for using LSA to automatically assess essay grades are explored and the experimentation methodology is explained. At the end of the paper the results are discussed and possible future experiments are suggested.

II. LATENT SEMANTIC ANALYSIS

Many excellent tutorials and papers on LSA already exist so only a brief explanation of LSA will be given. The reader is referred to publications such as [10] [2] [11] [12] for a more detailed description.

In essence, LSA is the process of converting a set of documents into a semantic space such that vector operations can be used to determine the semantic similarity of the documents (or terms). The process begins with the construction of a \( m \times n \) term by document matrix, say \( A \), where

\[
\begin{align*}
    m &= \text{Number of unique terms in corpus of documents} \\
    n &= \text{Number of documents in the corpus}
\end{align*}
\]

Common words with little semantic value are left out (from a pre-defined stop-words list). Term weighting is also applied in order to both de-emphasize common terms that may appear in the corpus of documents as well as emphasize less commonly used terms (term frequency \( \times \) inverse document frequency was used). Words should also have stemming/lemmatization applied to them to help ensure they are cut down to a common form (car, cars, car’s and cars’ should appear once as the term ‘car’). An implementation of Porter’s stemming algorithm is used for this [13]. The resulting term by document matrix is very sparse.

Vectors can now be formed from the rows and columns of \( A \). Any row is a vector for the term it represents with respect to all of the documents. In a similar fashion every column is a vector representing a document and the terms it contains.

Once the term by document matrix has been constructed and weighted a matrix decomposition technique known as Singular Value Decomposition (SVD) is applied to it which factors the matrix into three matrices giving us:

\[
A = TSD^T
\]
where
\[ A = m \times n \text { term by document matrix} \]
\[ T = m \times r \text { left singular vectors} \]
\[ S = r \times r \text { diagonal matrix of singular values} \]
\[ D = r \times n \text { right singular vectors} \]

The power of LSA comes from the reduction in dimension of \( A \). The singular value diagonal matrix, \( S \), only has \( r \) values (in decreasing order) in the diagonal of the matrix. If the lowest \( x \) values are removed from this matrix such that \( k = r - x \) it is said we have a \( k \)-dimensional approximation of \( A \). This matrix tends to be non-sparse as most of the zero values will become nonzero. Now we have a lower dimensional approximation of \( A \)
\[ A \approx \hat{A}_k = T_k S_k D_k^T \]

Each vector representing a document or term is now \( k \)-dimensional. Some of the underlying assumptions used by [2] in the development of this technique is that there is some underlying structure to a set of documents and the patterns of words present can give clues to the likely presence of others. It is this underlying "latent semantic" structure that we are trying to get to through the process of dimensional reduction of the original term by document matrix. The comparison metric of choice is to test the angle between two such vectors (documents would be compared with documents and terms with terms). For example, to compare two document vectors (columns)\(^1\) from \( \hat{A}_k \), say \( \vec{d}_1 \) and \( \vec{d}_2 \) we calculate the cosine between them as follows:
\[ \cos(\theta) = \frac{\vec{d}_1 \cdot \vec{d}_2}{\|\vec{d}_1\|\|\vec{d}_2\|} \]

The larger the value of \( \cos(\theta) \) the more semantically similar the documents.

Documents that are not present in the SVD operation can be still transformed into a vector that is in the same semantical space thus enabling comparisons with documents that were not involved in the SVD calculation [11]. This process is known as 'folding-in' a document [11]. Indeed, this is how it was originally intended to be used as a IR mechanism as the query would be modelled as a vector, folded-in to the semantic space and compared with the documents present to gauge similarity [2], [14].

One issue with LSA is the topic of selecting value of \( k \) to be used for the reduction in dimensionality. This is probably the most difficult parameter to decide upon using LSA as it is completely dependent on the individual SVD that is calculated and generally has to be chosen empirically by choosing the one that gives the best results [9] [15] (other attempts have been tried, however, such as the attempt by [16] to derive a formula for the optimal dimensional reduction. The author has found this not to work in practise however).

III. AUTOMATED ESSAY ASSESSMENT

Automated essay assessment can be modelled as a data mining or machine learning classification problem. The method described here is similar to the Holistic Method described in [17].

Essentially we have a set of essays that have been graded already and the grades represent classes. All of the ungraded papers need to be compared to the graded papers and classified according to some variant of the \( k \) nearest neighbours (KNN) algorithm [18].

In our case the graded papers would make up the semantic space. Using the graded essays a term-by-document matrix would be constructed and a SVD would be computed. Ungraded essays that are to be assessed would be folded-into the semantic space and compared against all graded essays. The KNN algorithm would be used to assign it a grade (class) based on the essays it is matched closest with.

In practise when experimenting with essay data sets the entire set of essays has been graded already. We use 75% of the essays (chosen at random) to use in order to construct the SVD. The remaining essays are used to be "assessed" using this semantic space. In order to ascertain the effectiveness of the algorithms the Spearman correlation between the actual grades of the essays and the computed grades is calculated. This metric seems common in the literature and indeed has been calculated between human graders [9] [19] [11]. In experiments done with different human graders assessing the same materials the correlation is generally found to be in the range of .7 and up. If an automated essay grading system can be shown to grade papers with a correlation of .7 or above it is asserted that the system can, therefore, grade as well as a human grader [6].

Other methods have been proposed and tried that use educational materials as the content of the semantic space and compare the texts to be graded against them [9]. This will be a direction of future research for the automated essay assessment component of this work.

A. Document Size

The size of documents to be used in LSA is a subject of debate. As essays, in the context of a written exam, tend to be fairly short it is likely that the longer the essay the easier it will be for a LSA based essay assessment tool to generate accurate results. This stands to reason at a high level as there will be more semantical information and relationships to exploit in the SVD semantic space transformation and, thus, more semantic relationships can be captured.

\(^1\)As an aside, it’s not necessary to calculate \( \hat{A}_k \) directly to extract the term/document vectors. The calculations can be made from the TSD matrices directly.
Document size is the crux of this paper. During the work on automated essay assessment at ShirWin Knowledge & Learning Systems it was serendipitously noted that two datasets out of the data being used were each answers to one question repeated across two different exam sittings. What makes this particular pair of essay sets interesting is the fact that in one of the exam sittings where students were provided with half a page to answer the question, while in the other sitting a full page was provided. Naturally the students tried to use the space available for their response and this resulted in two sets of essay answers to the same question with one of them having 56% more text than the others. This presented an excellent opportunity to apply the KNN approach to see if there is any noticeable difference in the quality of results when assessing the two sets of essays as this type of data is unusual to encounter. Both exam sittings were given and marked by the same instructor.

B. KNN Algorithms

Any set of essays that are to be assessed can have different grade distributions. This goes both for the training set used to compute the SVD as well as the essays that are to be graded. Different KNN algorithms should be available to try and take this into account to some degree.

1) KNN - Classic Voting System: The first algorithm employed is the classic KNN voting system. Some value of k is specified to be used (the number of nearest neighbours considered) and the k nearest essays are computed (via cosine comparison operator). Each grade for each neighbour gets a vote. The grade with the most votes is chosen as the grade to be assigned to the essay (ties are broken randomly). If the grade distribution is very skewed this may create a bias towards the region in the semantic space where the most common grades are.

2) KNN - Weighted: This method of calculating the grade tries to take into account that the distribution of grades is not uniform. Essentially it uses a similar approach to the previous algorithm but each grade is given a weight rather than just a straightforward vote. The weight of a neighbour’s vote for their grade or class \( C_i \) is calculated as:

\[
w_{C_i} = \frac{\text{Total Number of Essays}}{\text{# of Essays of class } C_i}
\]

This calculation tries to ensure that grades that are more common in the overall distribution carry a lower weight in the voting process (and ensure less common grades have a higher weight).

3) KNN - Weighted Score: The final KNN voting/weighting method tried is similar to the previous algorithm except the weight of the various neighbours each contribute a portion of the final grade to be assigned. This is described in the equation below which shows the assignment of a grade to essay \( E_i \) given its \( k \) nearest neighbours’ grades (the weights \( w \) in the formula below are calculated as per equation 1 above):

\[
G_j = \text{Grade of neighbour } j \text{ where } 1 \leq j \leq k
\]

Grade of \( E_i \) = \[
\sum_{j=1}^{k} G_j \left( \frac{w_j}{\sum_{l=1}^{k} w_l} \right)
\]

In a similar spirit to the weighted approach in the previous section the grade itself is calculated as the sums of the fractional contributions of each of the \( k \) nearest neighbours giving more emphasis on essay grades that are less common (so if they are neighbours it carries more weight and, thus, has more influence on the overall grade).

IV. Method

As outlined in section III we have seen the basic algorithm for the assessment of essays. In a real assessment of an essay grading system we would have a training set and some method of empirically selecting the dimension to use for the essay comparisons and employ some sort of training validation method (\( k \)-fold cross validation, bootstrap, etc.) [18]. Features such as style, coherence, length and language usage would also have to be taken into account. For our purposes, in testing the KNN algorithms applied to the two datasets to gauge any differences, we simply tried every dimension in the SVD and all possible values of \( k \) (for the KNN algorithms) to determine the best results (this is not unlike other experiments with KNN in the context of LSA such as [20]). The three KNN algorithms, as described in the previous section, are applied and the results are calculated via the Spearman correlation. Ten runs of this algorithm were performed and the average correlations are reported.

A. Datasets

The two datasets come from a final year course in the field of Occupational Therapy (OT) and answer a question on Functional Capacity Evaluation (FCE). As previously described they were answers for exactly the same question given on an exam in different years. One year allowed a half page for the response, the other year allowed a full page. Details of the datasets can be found in the table below. The first column simply gives the dataset name; the second column gives the average number of words in the essay response to the question whilst the third column indicates the maximum number of marks available for the essay. The last column indicates the size of the dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Average Size</th>
<th>Max Grade</th>
<th>Dataset Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCE-A</td>
<td>137</td>
<td>6</td>
<td>86</td>
</tr>
<tr>
<td>FCE-B</td>
<td>88</td>
<td>8</td>
<td>88</td>
</tr>
</tbody>
</table>

Table I

Dataset Information
As can be seen from the table the first dataset, on average, contains essays that are 56% longer than the second. It is also interesting to note that the number of essays in each set are nearly identical (86 vs 88).

V. RESULTS AND ANALYSIS

Ten trials of each algorithm against each dataset were run and the average correlation values were calculated and can be seen in the table below. It can be seen that the dataset with the larger essays has superior performance overall. The classic KNN algorithm and the Weighted KNN algorithm especially give support to this assertion. The Scaled and Weighted version of the KNN algorithm, however, seems to favour FCE-B dataset (although given the small difference the results can be viewed as comparable).

Table II

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>FCE-A</th>
<th>FCE-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classic KNN</td>
<td>0.84</td>
<td>0.60</td>
</tr>
<tr>
<td>Weighted KNN</td>
<td>0.71</td>
<td>0.63</td>
</tr>
<tr>
<td>Scaled KNN</td>
<td>0.65</td>
<td>0.69</td>
</tr>
</tbody>
</table>

The results indicated that, overall, the first dataset containing the larger essays had the better results. The experiment using the scaled and weighted grade assessment indicated the second essay set (which had a smaller average length) was slightly better but the distribution of the grades and algorithm used explains this partially. Even in this case the results between the two data sets were comparable. Overall it seems the best results were obtained using the dataset with the larger essays.

Future work with these datasets could include applying them to a more rigorous automated essay assessment environment. Rather than employing a holistic-like method, as was described in this paper, it would be interesting to see how they fare when assessed against the course materials the knowledge is drawn from like the work done by [9] [15].

Other metrics could also be employed to compare two documents in the semantic space. Most literature on the topic employ the cosine comparison methodology but others exist. For example, Euclidean distance could be used instead of the cosine comparison operator [22]. Additionally values such as the vector length of a document could be interpreted to indicate the quantity of information represented and incorporated into any comparison metric utilised [8].

ACKNOWLEDGMENT

The author would like to thank Shirwin Knowledge & Learning Systems for supporting this research and allowing this work to be shared. Gratitude is also extended towards Professor Sharon Brintnell of the University of Alberta for participating in this research with the kind provision of the datasets used.

REFERENCES


