# Latent Semantic Analysis for Local Document Searching

Final Year Project
Semester One Report

Jack Power
20080169
Supervisor: Rob O’Connor
BSc (Hons) in Applied Computing (IoT)

# Table of Contents

## Introduction

In the project proposal, I outlined a piece of software which would implement certain strategies to allow a user to search a document with ease and efficiency. The terms approximate string matching and conceptual searching were mentioned, which I felt represented the most significant aspects of the behaviour. In delving further, I was drawn strongly towards the inner workings of conceptual searching in particular. It represented to me such an abstract problem, the vague intuition which allowed me to reason about basic string manipulation deserted me entirely when trying to imagine how a computer could be taught to extract the meaning from words. The intangible solution lay at a junction of computer science, statistical analysis and linguistics.

In discussion with my project supervisor, we quickly identified that this intriguing problem represented in and of itself an excellent opportunity for thorough investigation. We therefore decided to adopt a research approach, we both felt that to tease out the finer details of this method and to make plain to the reader the mechanism by which it works would result in a piece of work which was of more value to the reader, as well as hopefully being more interesting.

## Background & Context

Fundamentally, our problem consists of associating words with similar meanings. In order to achieve this, we must select some observable feature which will indicate some degree of association between words. One intuitive approach is the distributional hypothesis, the assumption that words with similar meanings will occur with similar distributions. This is more eloquently expressed by John Firth (1957),

> *You shall know a word by the company it keeps.*

This idea has some immediate consequences, we may now link the concepts of occurrence and meaning so that the relationships found in one domain pass naturally to the other. It is instinctive to say that a document is to do with a subject, the document has meaning and so too do the words within. In fact, what is the meaning of a document but a sort of average of the words it contains? And reflexively, the meaning of a word must also be some sort of an average of the various instances in which it is used.

It is helpful at this point to introduce some terminology. As well as simplifying and clarifying the subsequent discussion, I find that it also gives some indication of the mindset, a way to parse the problem as our predecessors conceived it. A *document* is a collection of words, as one would expect. The*body* is our collection of documents, to use the language of statistical analysis our dataset in its entirety. The*dictionary* is all the words across all the documents in our body, and a *topic* is a collection of words that co-occur, and by extension share some common meaning or association.

Our first task is to associate our two conveyors of meaning, documents and words, into some format which yields objective data that can be analysed. This is the document-term matrix, a bag-of-words model being the simplest implementation. The body constitutes the rows of the matrix, the dictionary the columns. Each occurrence of a term in a document is counted, the *term frequency*, and entered into the corresponding cell.

The following is a simple example:

Document 1:

*This is the first document.*

Document 2:

*This is another document.*

| | this | is | the | first | document | another |
|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 1 | 1 | 1 | 0 |
| D2 | 1 | 1 | 0 | 0 | 1 | 1 |

Now that our data has been expressed in the form of a matrix, some fundamental concepts of linear algebra may occur to us to capture the relationship between terms and documents. Since our terms and documents can now be thought of as vectors, an obvious candidate is cosine similarity. Essentially, if two documents contain many of the same terms, terms being the column space of the matrix, the two vectors can be visualised as 'pointing' in similar directions, and hence they are similar documents. The converse can also be applied to terms, if we now treat the documents in which they appear as dimensions in which similar terms will 'point' together.

Another example showcasing this notion of similarity:

"Red and blue are colours."
"Apples are fruit."
"Bananas are fruit too."

| | red | and | blue | are | colours | apples | fruit | bananas | too |
|---|---|---|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| D2 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| D3 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |

We can therefore see that the terms red and blue are related through document 1, both having a magnitude of one in the 'direction' of document 1. This certainly agrees with what we know about these terms, they are both colours. We may expect the terms apples and bananas to be related similarly, however notice that the terms apples and bananas share no common document, they do not co-occur. This demonstrates the problem with this method, two terms may indeed be related but may not necessarily occur within the same document. We can observe however, that the term apples occurs alongside the term fruit, and so too does fruit occur with the term bananas. This suggests the concept of a *topic*, a common theme which ties terms together despite the fact that they are distributed across documents.

Mathematically, this concept is manifested via a dimensionality reduction of the document-term matrix. Implemented using a singular value decomposition (SVD), the operation essentially extracts the most significant components of the matrix, merging columns which affect the matrix in similar ways. In our domain of terms and topics, this means that words which commonly occur together will be merged into a single topic,

linking all words to do with that topic.

The process which has been described, construction of the document-term matrix through to the SVD, is known as latent semantic analysis (LSA). It refers to the fact that contained within our data, the body, there are certain latent features which cannot be directly measured, but may be extracted in the form of topics.

**Example**

The following is a small example of LSA in action based on a video series by Databricks Academy (2019).

Recall that an SVD is a dimensionality reduction, like a form of compression. The result is that our original data, the raw text in our body, is now encoded in terms of the topics, each term has been reduced to a *weight* in a given topic. We may examine the internal representation of this data, known as the *encoding matrix*, during the process. In this particular experiment two source texts were used, Goldilocks and Little Red Riding Hood.

Shown is the encoding matrix, sorted in descending order first by the absolute value of topic 1 and then by the absolute value of topic 2.

| topic_1 | topic_2 | terms | abs_topic_1 | abs_topic_2 |
|---|---|---|---|---|
| 0.387635 | -0.128258 | hood | 0.387635 | 0.128258 |
| 0.387635 | -0.128258 | riding | 0.387635 | 0.128258 |
| 0.387635 | -0.128258 | red | 0.387635 | 0.128258 |
| 0.337416 | -0.118552 | little | 0.337416 | 0.118552 |
| 0.261872 | 0.018790 | grandmother | 0.261872 | 0.018790 |
| ... | ... | ... | ... | ... |
| 0.002376 | 0.023842 | table | 0.002376 | 0.023842 |
| 0.001920 | -0.000639 | reply | 0.001920 | 0.000639 |
| 0.000717 | 0.000111 | hug | 0.000717 | 0.000111 |
| 0.000709 | 0.002879 | apron | 0.000709 | 0.002879 |
| 0.000041 | -0.000212 | delighted | 0.000041 | 0.000212 |

| topic_1 | topic_2 | terms | abs_topic_1 | abs_topic_2 |
|---|---|---|---|---|
| 0.059371 | 0.404514 | porridge | 0.059371 | 0.404514 |
| 0.069969 | 0.401925 | bear | 0.069969 | 0.401925 |
| 0.044797 | 0.277285 | chair | 0.044797 | 0.277285 |
| 0.240266 | 0.232649 | said | 0.240266 | 0.232649 |
| 0.098384 | 0.230839 | bed | 0.098384 | 0.230839 |
| ... | ... | ... | ... | ... |
| 0.020070 | -0.000118 | dear | 0.020070 | 0.000118 |
| 0.000717 | 0.000111 | hug | 0.000717 | 0.000111 |
| 0.010879 | -0.000109 | lift | 0.010879 | 0.000109 |
| 0.004964 | 0.000031 | chairs | 0.004964 | 0.000031 |
| 0.004964 | 0.000031 | living | 0.004964 | 0.000031 |

Observing the most heavily weighted terms in each topic, it is clear that the LSA was successful in distinguishing the two texts as two distinct topics. Moreover, the terms determined to be 'most significant' mathematically bear striking similarity to what a human might select as the most significant, effectively a handful of keywords which best characterises each book.

There are some additional details of this implementation which are worth mentioning. Firstly, note the signed versions of the topic weightings, a strongly negative weighting has the connotation of dissociation in the same way a strongly positive weighting represents association.

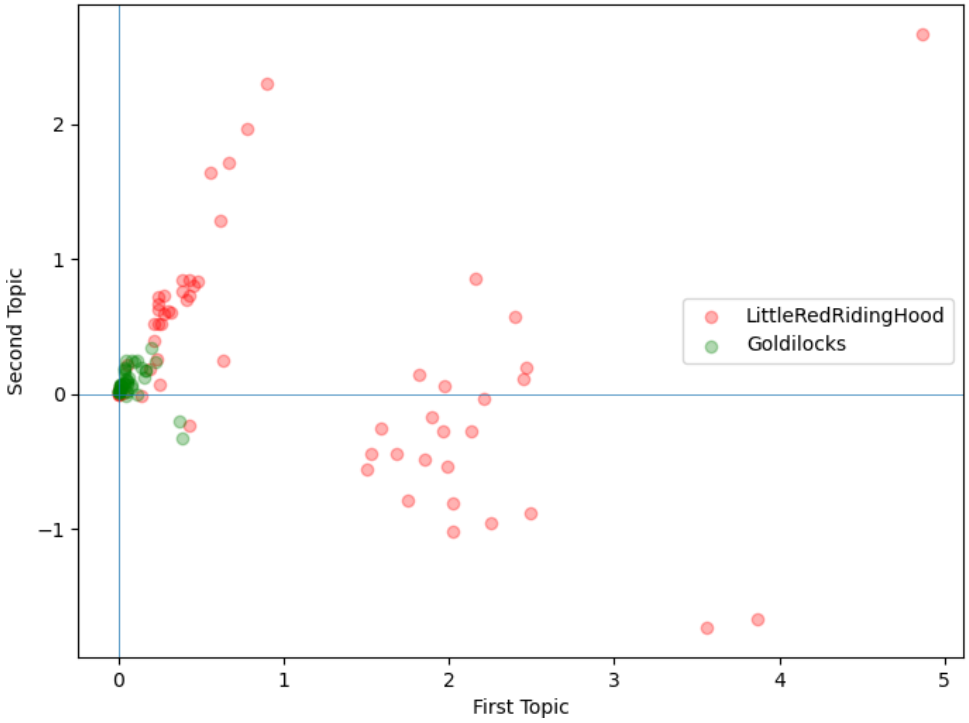
Of particular interest is how LSA differs from many machine learning techniques in that the *hyperparameters*, the parameters used to control the learning process, cannot be tuned as is conventionally possible. This is due to the fact that the result of the SVD, the topic encoded data which represents the most significant parts of the matrix, is inherent to the data itself. Technically speaking, the outputs of the decomposition are the singular values of the matrix, as well as the left- and right-singular vectors. These are analogous to the eigenvalues and eigenvectors in the eigendecomposition of a square matrix. What is significant is that this is simply another way of representing the data, a mathematical result which fundamentally cannot be changed.

This has the consequence that if we wish to improve the analysis, we must look not to the analysis itself but to the data which we feed it, namely the document-term matrix. This constitutes data preprocessing, a common and extremely useful technique is to substitute *term frequency* for another subtly different metric. *Term frequency-inverse document frequency* (tf-idf) has that advantage that, when compared with the simple count used previously, a term is penalised if it appears in many documents. This is intuitive in that a word which appears in many different documents is less meaningful to any one particular topic. The topic-encoded data returned from the SVD can be observed to be less noisy than when measuring term frequency alone.

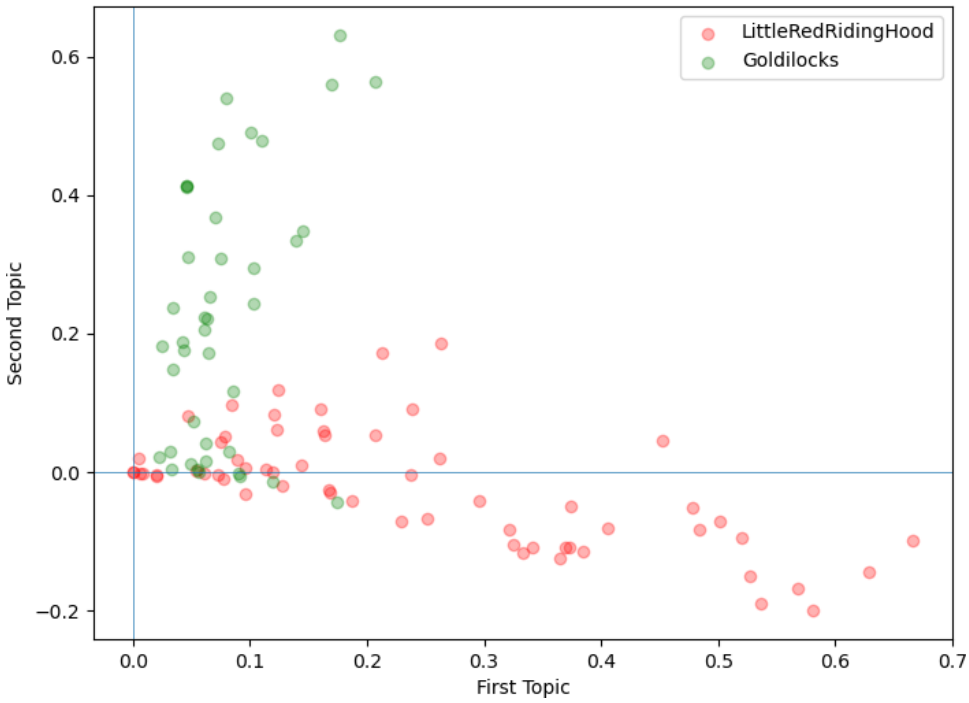
An additional preprocessing step that can be applied to the document-term matrix is the generation of word lemmas. A word lemma is the dictionary definition of a word. For example, the words run, ran and running all stem from the word run. In LSA, these words should be considered to be the same as they all have the same meaning with regard to topics. The same applies for plurals. In addition, common words such as 'the' or 'a' which carry little meaning, known as stop words, can be removed from the analysis when constructing the document-term matrix to improve the analysis.

Shown is the significant noise reduction afforded by the use of tf-idf over simple term frequency. Note the improved division of topics in the second figure.

term frequency




term frequency-inverse document frequency



## Literature Review

In LSA's seminal paper by Deerwester et al (1990), several problems facing preexisting information retrieval techniques are outlined. In this context of information retrieval, where a set of related documents are to be returned for some specific query, the method was introduced as latent semantic indexing (LSI). The main difficulties in basic term overlap schemes, where the terms in the query are matched against words in the documents, are the phenomena of *synonymy* and *polysemy*. Synonymy refers to when multiple terms convey the same meaning. Polysemy refers to when a single term has many possible meanings, depending on the context. For instance, the word 'chip' has different meanings in the contexts of computing versus cooking.

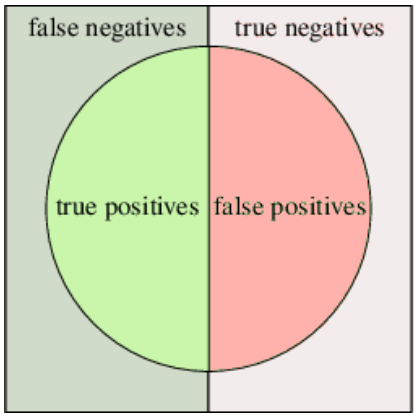
These idiosyncrasies of language create issues when a human is tasked with describing the documents they are looking for. They may by chance use every word applicable to a topic except for the one which appears in the document. They may also experience frustration when in searching for literature on chip design they receive cooking recipes instead. This unreliability of term overlap culminates in the following realisation. Any particular document can be considered to be only a small selection of all the possible discourse on its associated topic. Similarly, the query string is only a small selection of words which may be associated with the implied topic. Thus, extracting index terms from either is inherently fallible, there always exists the possibility of omission on either end. What the user truly desires is to retrieve all documents associated with the topic that this query implies, the 'latent semantics'.

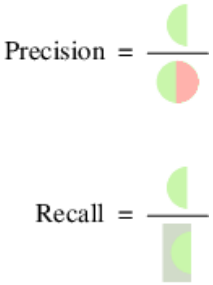
This largely eliminates the problems mentioned earlier. Based on a well-developed body of existing documents, which now exist in a sort of *concept space* where discrete topics each occupy their own dimension, the model now knows which topics are implied given certain combinations of words. If the word 'chip' were to appear with the words 'fry' and 'oil', the model will associate this instance of chip with its appropriate meaning, cooking.

One additional benefit of this representation is that both documents and terms can be represented in space simultaneously. As mentioned before, documents can be thought of as an average of the words they contain, and terms likewise are an average of the documents in which they appear. Taking advantage of this, new documents can be 'folded-in' to the existing model by placing them at the centroid of their constituent terms in space. The computational implications are significant when compared with repeating the analysis entirely each time new information must be added. Note however, that this introduces a degree of temporal dependency. The state of the model after folding-in a new document is distinct from the space obtained had the document been present in the original analysis. The point at which this becomes detrimental is a subject of further research.

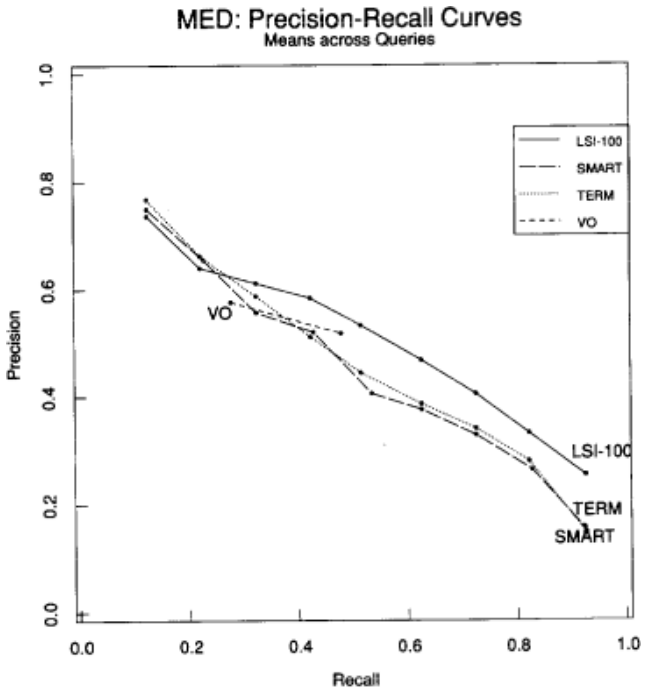
The same technique can be used in the execution of queries. A query can be thought of as a *pseudo-document*, a weighted sum of its component term vectors, which can be inserted into the space as normal. The documents nearby the query object are determined to be related to the query, and are returned.

Two significant metrics in the evaluation of information retrieval methods are *precision* and *recall*. Precision refers to the fraction of retrieved documents that are relevant to the query. This is more generally referred to as*positive pr edictive value*. Recall is the fraction of all relevant documents that were successfully retrieved. This is also commonly known as the *sensitivity*.




In the introduction of LSA, the technique compared very favourably to contemporary term overlap methods. Even without the benefit of tf-idf or word lemma generation as discussed earlier, something which had already been integrated into preexisting techniques, the latent semantic approach demonstrated significant inherent potential. Performance is evaluated by measuring the precision at different levels of recall. Deerwester et al observed that for all but the two lowest levels, the precision of LSA significantly exceeded other methods. Shown are the precision-recall curves for the 100-dimension LSA used (LSI-100), term matching (TERM), SMART and a vector retrieval system utilising Boolean queries by Voorhees (1985) (VO).



Significant challenges encountered early in the implementation of LSA related to scalability and performance, essentially limited by the computing resources of the time (Deerwester et al, 1990). As these restrictions have fallen away with the availability of greater processing power and inexpensive memory, implementations have achieved the indexing of millions of documents with hundreds of conceptual dimensions (Bradford, 2008). The optimal number of dimensions has been the topic of much research. Higher numbers allows the model to capture more of the structure present in the data, however too many will result in the modelling of noise or irrelevant features. Recall that LSA functions by approximating the document-term matrix into its most significant parts, if the original data is not sufficiently reduced topic merging will not take place as desired.

Latent semantic analysis was first described and patented in the late 80s. It first found use in 1992 to automatically assign submitted manuscripts to reviewers based on the reviewer's interests (Dumais and Nielsen, 1992). It has since been used in applications such as document classification (Foltz and Dumais, 1992), spam filtering (Gee, 2003), and even automated scoring of essays (Foltz, Laham and Landauer, 1999).

## Experiment and Goals

It is my intention to showcase in detail the operation and behaviour of latent semantic analysis. To achieve this I will construct a scientific testbed in the form of AWS computing services equipped with implementations of LSA and some standard datasets to process. I intend to replicate the findings observed in the seminal paper and to then introduce various alterations which have been developed since. I have discussed a sample of these already, the use of tf-idf and word lemmatising are notable in particular. It is my hope that by first utilising a clear example of the core of LSA, then introducing transparent modifications and observing the associated outcomes, the result will be a contemporary, modular implementation which will be of great educational benefit both to myself and to the reader.

## References

Firth, J. (1957). *Papers in Linguistics 1934-1951*. London: Oxford University Press.

Databricks Academy. (2019). *Introduction to Latent Semantic Analysis*. [video] Available at: https://www.youtube.com/watch?v=hB51kkus-Rc [Accessed 12 Jan. 2021].

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R. (1990). Indexing by Latent Semantic Analysis. *J ournal of the American Society for Information Science*, 41(6), pp. 391-407.

Voorhees, E. (1985). The cluster hypothesis revisited. In:*Pr oceedings of SIGIR*, pp. 188-196.

Bradford, R. (2008). An Empirical Study of Required Dimensionality for Large-Scale Latent Semantic Analysis Indexing Applications. In: *CIKM '08: Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 153-162.

Dumais, S., Nielsen, J. (1992). Automating the Assignment of Submitted Manuscripts to Reviewers. In:*Resear ch and Development in Information Retrieval*, ACM Press, pp. 233-244.

Foltz, P, Dumais, S. (1992). Personalised Information Delivery: An Analysis of Information Filtering Methods. *Communications of the ACM*, 34(12), pp. 51-60.

Gee, K. (2003). Using Latent Semantic Indexing to Filter Spam. In: *Proceedings of the 2003 ACM Symposium on Applied Computing*, pp. 460-464.

Foltz, P., Laham, D., Landauer, T. (1999). The Intelligent Essay Assessor: Applications to Educational Technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 1.