Latent Semantic Analysis for Local Document Searching

Final Year Project
Semester One Presentation

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

What is LSA?

Latent Semantic Analysis (LSA) is a technique in natural language processing which allows us to express raw text in terms of *topics*.

It was first developed and patented for use in information retrieval as latent semantic indexing (LSI) (Deerwester et al, 1990). The goal of information retrieval is to identify and return documents which are relevant to a given query.

LSA distinguishes itself from term overlap methods in that it attempts to capture the meaning implied by the query, the 'latent semantics', rather than simply the specific words used.

Searching for the word 'dog', for instance, should return results containing 'canine'. The words are different but the meaning is the same.

How does it work?

LSA is an application of distributional semantics, based upon the distributional hypothesis which states that words with similar meanings will occur with similar distributions.

> *"You shall know a word by the company it keeps."*
>
> John Firth, (1957)

Assuming this relationship between meaning and occurrence, some interesting ideas begin to take shape.

If we consider a *document* to be a collection of words, or *terms*, we may suggest that the meaning of a document is a sort of average of all the terms it contains. It also follows that the true meaning of a term is some average of the various instances in which it is used.

In order to analyse the structure of our data, it must first be put into some workable form. This is the *document-term matrix*, with each document corresponding to a row, and each term to a column. Each occurrence of a term in a document is counted, the *term frequency*, and entered into the corresponding cell.

Document 1:

This is the first document.

Document 2:

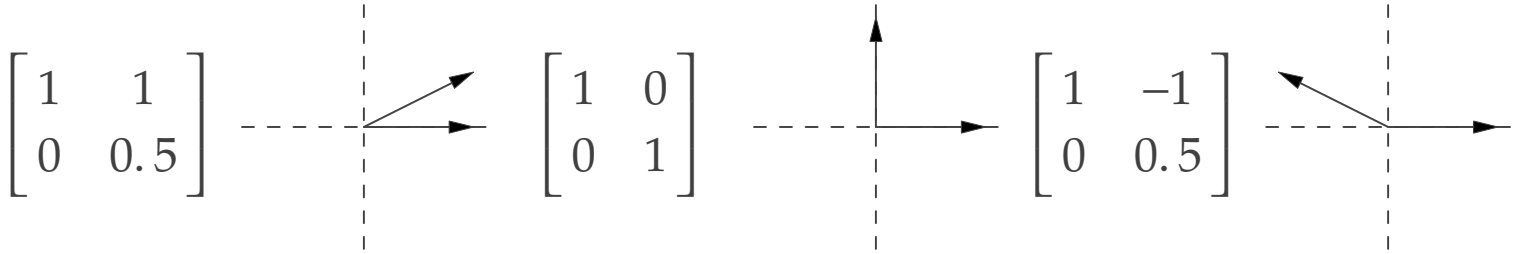
This is another document.

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

We can now apply the sophisticated toolset of linear algebra and statistics in order to extract the meaning from our data.

Cosine similarity

Some understanding of linear algebra is essential to understanding the mechanism of LSA.



When two documents are said to be similar, this is an expression of how closely their term components match. In the same sense, two terms are similar if they appear together across the same documents.

This can be visualised as two documents or terms 'pointing' together, albeit in a higher number of dimensions.

Using this, a *concept space* can be constructed in which each term and document object can be represented as a vector.

In LSA, a dimensionality reduction is used in order to re-encode the data in terms of the topics. Terms and documents can now be expressed as weighted sums of these topic vectors.

To utilise this space, we can construct queries as vector sums of their constituent terms. The document vectors nearby the query are similar and therefore related.

Amazon SageMaker

Amazon SageMaker is a fully managed machine learning service available through AWS (AWS, 2021).

It consists of an EC2 compute instance running Jupyter Notebook geared towards the building, training and deployment of ML models.

For the purposes of this project, SageMaker offers state-of-the-art implementations of key natural language processing algorithms complete with the required toolset to test and evaluate them.

Contemporary Algorithms

Two algorithms offered by SageMaker which utilise the same topic model as LSA are:

Latent Dirichlet Allocation (LDA) word2vec

LDA is an extension of probabilistic latent semantic analysis (PLSA), essentially tuning the model to maximise the probability of reconstructing the original body (Serrano, 2020).

Word2vec aims to preserve linear regularities among words. This allows for vector operations such as

```
vector("King") - vector("Man") + vector("Woman")
```

to yield a vector closest to the vector representation of "Queen" (Mikolov et al, 2013).

Experimental Method

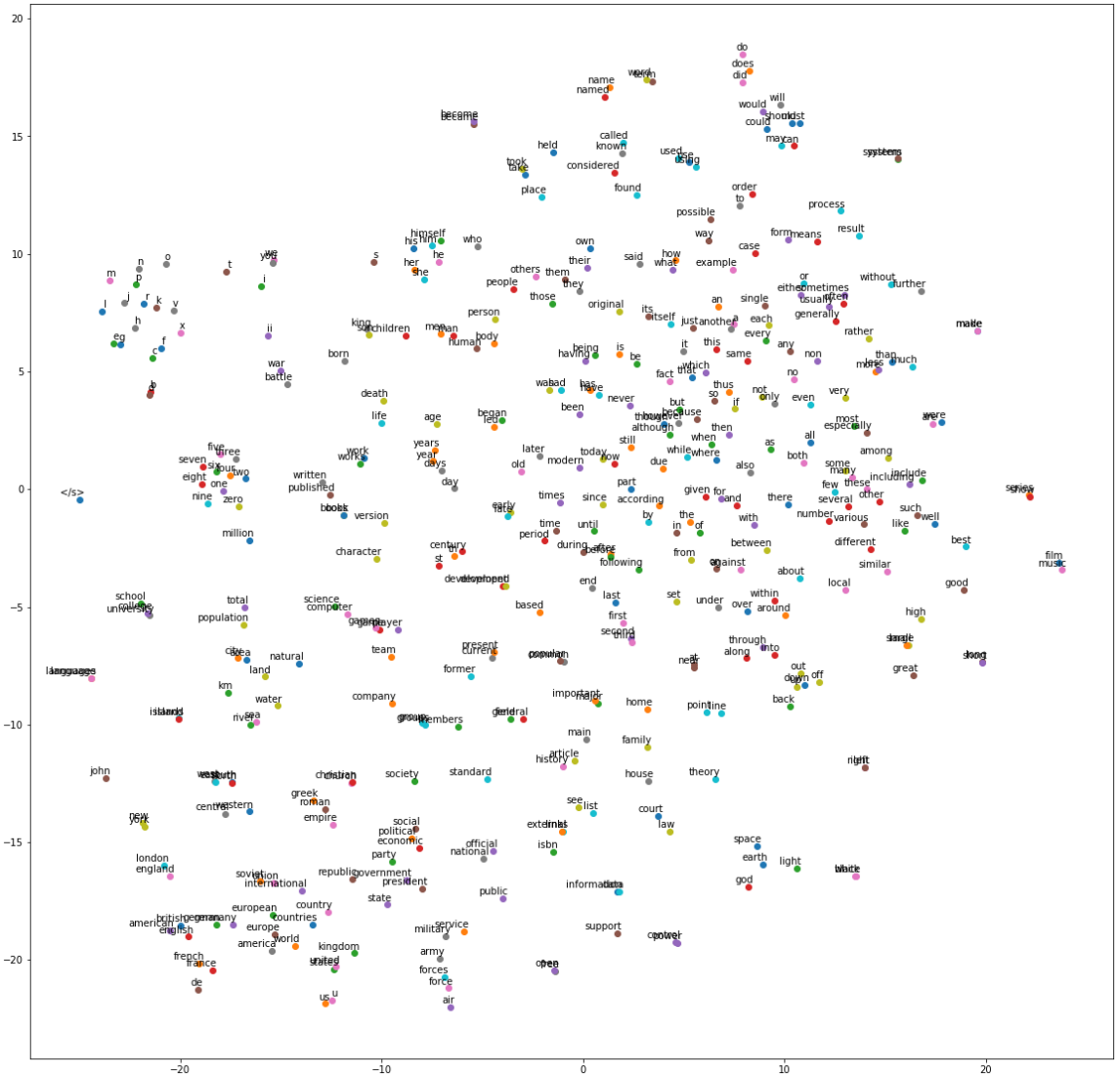
The wealth of NLP literature provides many well-used datasets to train, test and evaluate models.

For instance, the SemEval workshops produce tasks used to benchmark semantic analysis systems.

SemEval-2012 Task 2, for measuring relational similarity, was used to great effect by Mikolov, Yih and Zweig (2013) in demonstrating their vector offset method's capabilities compared with LSA approaches.

In a similar fashion, I aim to interrogate the topic modelling techniques available to me, with the expectation that performance will improve as newer developments are introduced. This hypothesis will doubtlessly be refined as further details are uncovered.

A t-SNE plot of word2vec (text8 dataset) (AWS, 2018)



References

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

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

AWS. (2021). *Amazon SageMaker*. [online] Available at: https://aws.amazon.com/sagemaker/ [Accessed 18 Jan. 2021]

Serrano, L. (2020). *Latent Dirichlet Allocation*. [video] A vailable at: https://www.youtube.com/watch?v=T05t-SqKArY [Accessed 18 Jan. 2021]

Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *The International Conference on Learning Representations*.

Mikolov, T., Yih, W., Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. *North American Chapter of the Association for Computational Linguistics*.

AWS. (2018). *Amazon SageMaker Examples BlazingText word2vec text8*. [image] Available at: https://github.com/aws/amazon-sagemaker-examples/tree/master/introduction_to_amazon_algorithms/blazingtext_word2vec_text8/tsne.png