Latent Semantic Analysis

Peter Wiemer-Hastings
peterwh@cti.depaul.edu
DePaul University
School of Computer Science,
Telecommunications, and Information Systems
243 South Wabash Avenue
Chicago IL 60604, USA

November 10, 2004

Abstract

Latent Semantic Analysis (LSA) is a technique for comparing texts using a vector-based representation that is learned from a corpus. This article begins with a description of the history of LSA and its basic functionality. LSA enjoys both theoretical support and empirical results that show how it matches human behavior. A number of the experiments that compare LSA with humans are described here. The article also describes a few of the many successful applications of LSA to text-processing problems, and finishes by presenting a number of current research directions.

1 Introduction

Latent Semantic Analysis (LSA) is a technique for creating vector-based representations of texts which are claimed to capture their semantic content. The primary function of LSA is to compute the similarity of text pairs

by comparing their vector representations. This relatively simple similarity metric has been situated within a psychological theory of text meaning and has been shown to closely match human capabilities on a variety of tasks.

This article follows the developmental path of LSA, describing its historical context, showing how it computes and uses its vector representations, and then giving examples of the theoretical and empirical support for LSA and its current research directions.

2 How LSA works

LSA (originally known as Latent Semantic Indexing) was developed for the task of Information Retrieval, that is, selecting from a large database of documents a few relevant documents which match a given query . Previous approaches to this task included keyword-matching, weighted keyword matching, and vector-based representations based on the occurrence of words in documents. LSA extends the vector-based approach by using Singular Value Decomposition (SVD) to reconfigure the data. The details of this process are described below, but the intuition is that there is a set of underlying latent variables which spans the meanings that can be expressed in a particular language. These variables are assumed to be independent (and therefore orthogonal in the vector space). SVD is a matrix algebra technique which essentially re-orients and ranks the dimensions in a vector space.

Because the dimensions in a vector space computed by SVD are ordered from most to least significant, if some of the less significant dimensions are

ignored, the reduced representation is guaranteed to be the best possible for that dimensionality. In LSA, the typical assumption is that only the top 300 or so dimensions (out of tens or even hundreds of thousands) are useful for capturing the meaning of texts. By basing the representations on a reduced number of dimensions, words that generally occur in similar contexts have similar vectors and will therefore get a high similarity rating. The discarded dimensions are assumed to be the product of noise, random associations, or some other non-essential factor.

That LSA performed information retrieval better than its rival approaches is not especially surprising. What is more surprising is how well it models human behavior on a variety of linguistic tasks. Before describing these, however, the LSA technique is described in more detail.

Although there are some variations, the most common steps are these:

- Collect a large set of (domain-relevant) text and separate it into "documents". For most applications, each paragraph is treated as a separate document based on the intuition that the information within a paragraph tends to be coherent and related.

- Next, create a co-occurrence matrix of documents and terms. The cell in this matrix corresponding to document x and term y contains the number of times y occurs in x. A term is defined as a word which occurs in more than one document, and no stemming or other morphological analysis is performed to try to combine different forms of the same word.

If there are m terms and n documents, this matrix can be viewed as giving a representation which has an m-dimensional vector for each document, and an n-dimensional vector for each term.

- The values in each cell may be weighted to reduce the effect of common words that occur through the corpus. A common weighting method is "log entropy", based on Information Theory, which multiplies the value by its information gain.

- SVD is invoked with a parameter k which specifies the desired number of dimensions. (In principle, the SVD would be computed with all the dimensions to create three matrices that, when multiplied together, would produce the original data, but due to the amount of memory that this would require, this is not feasible. Instead, the currently used algorithms are optimized for dealing with sparse data spaces and compute only the most significant k dimensions of the matrices.)

The result of the processing above is three matrices. One has a k-dimensional vector for each document, one has a k-dimensional vector for each term in the corpus, and one has the k singular values. The first two matrices define two different vector spaces which are also different from the space defined by the original matrix. The singular values can be used to transform a vector from one space to the other. The use of these matrices depends on the application.

For information retrieval, the document vectors contain the LSA representation of each document. A query is turned into a "pseudodoc" in the

document vector space by combining the vectors for the terms in the query, and dividing by the singular values. Vectors are typically compared by computing the cosine between them. (Some applications use other distance metrics.) The closest vectors from the document vector space correspond to the documents which are closest in meaning to the query (according to LSA).

In most other applications, the original documents are only used for training, that is, creating the semantic space. To compare new texts, the term vectors are combined as described above. Here, no manipulation with the singular values is required because the vectors are compared in the term space.

For more details about the mathematical foundations of LSA, see Golub (1989) and Hu (2005). For more details about the creation of LSA spaces, see Deerwester (1990) and Quesada (2005).

3 Support for LSA

Support for LSA might be said to stem from the time of World War II when Wittgenstein wrote (what was later translated as):

> ... for a large class of cases — though not for all — in which we employ the word "meaning" it can be defined as thus, the meaning of the word is its use in language. (Wittgenstein, 1958, p. 20)

There have been a large number of psychological studies which have taken Wittgenstein's words to heart, and shown that LSA's behavior is closely matched with that of humans, for example:

- LSA acquires words at a similar pace to human children, sometimes exceeding the number of words to which it is exposed (see Landauer and Dumais, 1997) .
- LSA's knowledge of synonyms is as good as that of second-language English speakers as evidenced by scores on the Test of English as a Foreign Language (TOEFL, see Landauer, 1997) .
- LSA can tell students what they should read next to help them learn (see Wolfe, 1998).
- LSA can even interpret metaphors like, "My lawyer is a shark" (see Kintsch, 2001).

For textual applications, LSA has another benefit besides its high correlation with human behavior. When compared with the traditional labor-intensive approach to Natural Language Processing — developing a grammar, a lexicon, a semantic representation and the processing engine needed to combine them — developing an LSA-based representation is quite simple. It also has the advantage of graceful degradation. If it doesn't know a word, LSA simply ignores it and bases its representation on the other words. This has led researchers to use LSA for a variety of applications, including:

- intelligent tutoring systems which allow students to enter unconstrained natural language replies to questions (see Graesser, 2000 and Wiemer-Hastings, 2004),
- grading psychology essays by comparing them to pre-graded essays (see Foltz, 1996) ,
- evaluating summaries of documents to help teach summarization skills (Summary Street, described at http://colit.org/),
- helping students learn to properly integrate and cite material from multiple documents (see Britt, 2005), and
- evaluating airplane landings in a flight simulator (see Quesada, 2005).

The only applicability constraints for LSA are that the task is text-based, it can be framed in terms of computing the similarity of texts, and there is an available training corpus. The tutoring systems, for example, compare a student's answer for a question to a set of expected answers. If the student's response is close enough to a good answer, then the system gives positive feedback and moves on to the next question. If the student's answer matches an expected bad answer, then the system steers the student back on track.

4 Issues

The research issues facing the LSA community range from the practical to the philosophical. One basic question addresses the size and substance of the

training corpus. Many effective LSA applications have been developed using relatively small corpora. In one of the successful applications mentioned above, LSA was trained on a corpus of only a couple hundred kilobytes with 2000 word types, 30,000 word tokens, and 325 documents. In contrast, researchers at the University of Colorado have reported that they have trained LSA on a corpus containing 750,000 word types, 550 million word tokens, and 3.6 million documents.

Unfortunately, there is little hard evidence on what the "ideal" size of an LSA corpus might be. The current data suggests that adding additional texts is unlikely to reduce performance, so a basic rule of thumb is, "the more the better."

The obvious follow-up question is, "What kinds of text should be included in an LSA corpus?" The common wisdom holds that the corpus should consist of texts which are relevant to the particular target task. The domain can define a sub-language where words are interpreted in consistent ways. Furthermore, a primary concern is to achieve sufficient coverage of the words which will be encountered in the course of running the application. A domain-specific corpus will have a higher percentage of relevant words and will thus not waste its "representational power" on words that will not be seen by the application.

One critical objection that is raised against the LSA approach is that not only does it ignore the syntactic structure of sentences, it even ignores word order. In other words, LSA treats a text as a bag of words. In practice, LSA does well with longer passages of words (defined by Rehder (1998) as

more than 200 words) where syntactic details may be "washed out", and it also does well with single words (the TOEFL test, for example), but it does not do well on single sentences as shown by Wiemer-Hastings (1999) . There have been a variety of approaches which attempt to deal with this, including surface parsing of sentences so that the components can be compared separately with LSA and using String Edit Theory to infer structural relations (see Wiemer-Hastings, 2001, Kanejiya, 2003, and Dennis, 2005) .

Another notable gap in LSA's competence is negations. thing that LSA "ignores" is negations, either because they are omitted from the LSA training via a "stop words" list, or simply because their widespread use throughout a corpus renders them representationally depleted. Although no satisfactory approach yet exists for dealing with negations, a possibility would be to treat them as an essentially syntactic component that can processed as described above.

A more fundamental question about LSA is what its dimensions "mean". Because they represent latent variables, there is no clear definition. As shown by Hu (2003), there is a high correlation between the first dimension and the frequency of occurrence of the words in the corpus . Beyond that, there are no clear answers. There is also considerable debate as to what extent LSA captures first order co-occurrence or higher order co-occurrence. Recent evidence from Denhière (2005) shows that although second order effects do occur, large changes in the similarity measure between two words can be seen when a document is added to a training corpus in which both words occur (first order co-occurrence) .

LSA's utility and correspondence with human behavor have made it a popular technique for psycholinguistic research and text processing. The issues describe here (along with others, for example, "Is LSA at all psychologically plausible?") will keeep researchers busy for years to come. A collection edited by McNamara (2005) provides considerably more detail about the current practice and research on LSA.

References

[Britt et al., 2005] Britt, A., Wiemer-Hastings, P., Larson, A., and Perfetti, C. (2005). Using intelligent feedback to improve sourcing and integration in students' essays. *International Journal of Artificial Intelligence in Education*. In press.

[Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41:391–407.

[Denhière et al., 2005] Denhière, G., Lemaire, B., Bellisens, C., and Jhean, S. (2005). A semantic space for modeling a child semantic memory. In McNamara, D., Landauer, T., Dennis, S., and Kintsch, W., editors, *LSA: A Road to Meaning*. Erlbaum, Mahwah, NJ. In press.

[Dennis, 2005] Dennis, S. (2005). Introducing word order. In McNamara, D., Landauer, T., Dennis, S., and Kintsch, W., editors, *LSA: A Road to Meaning*. Erlbaum, Mahwah, NJ. In press.

[Foltz, 1996] Foltz, P. (1996). Latent Semantic Analysis for text-based research. *Behavior Research Methods, Instruments, and Computers*, 28:197–202.

[Golub and Van Loan, 1989] Golub, G. H. and Van Loan, C. F. (1989). *Matrix Computations*. Johns Hopkins, Baltimore, MD, 3rd edition.

[Graesser et al., 2000] Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., and the Tutoring Research Group (2000). Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, 8(2):129–147.

[Hu et al., 2003] Hu, X., Cai, Z., Franceschetti, D., Penumatsa, P., Graesser, A., and Louwerse, M. (2003). LSA: The first dimension and dimensional weighting. In Alterman, R. and Hirsh, D., editors, *Proceedings of the 25th Annual Conference of the Cognitive Science Society*.

[Hu et al., 2005] Hu, X., Cai, Z., Wiemer-Hastings, P., Graesser, A., and McNamara, D. (2005). Strengths, limitations, and extensions of LSA. In McNamara, D., Landauer, T., Dennis, S., and Kintsch, W., editors, *LSA: A Road to Meaning*. Erlbaum, Mahwah, NJ. In press.

[Kanejiya et al., 2003] Kanejiya, D., Kumar, A., and Prasad, S. (2003). Automatic evaluation of students' answers using syntactically enhanced

LSA. In *Proceedings of the Human Language Technology Conference (HLT-NAACL 2003) Workshop on Building Educational Applications using NLP*. available at: http://www.cse.iitd.ernet.in/~eer99010/pub/hlt-naacl03.pdf.

[Kintsch, 2001] Kintsch, W. (2001). Predication. *Cognitive Science*, 25:173–202.

[Landauer and Dumais, 1997] Landauer, T. and Dumais, S. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

[Landauer et al., 1997] Landauer, T. K., Laham, D., Rehder, R., and Schreiner, M. E. (1997). How well can passage meaning be derived without using word order? a comparison of Latent Semantic Analysis and humans. In *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pages 412–417, Mahwah, NJ. Erlbaum.

[McNamara et al., 2005] McNamara, D., Landauer, T., Dennis, S., and Kintsch, W., editors (2005). *LSA: A Road to Meaning*. Erlbaum, Mahwah, NJ. In press.

[Quesada, 2005a] Quesada, J. (2005a). Creating your own LSA space. In McNamara, D., Landauer, T., Dennis, S., and Kintsch, W., editors, *LSA: A Road to Meaning*. Erlbaum, Mahwah, NJ. In press.

[Quesada, 2005b] Quesada, J. (2005b). Understanding representation in problem solving, judgment and decision making as the creation of multidimensional space. In McNamara, D., Landauer, T., Dennis, S., and Kintsch, W., editors, *LSA: A Road to Meaning*. Erlbaum, Mahwah, NJ. In press.

[Rehder et al., 1998] Rehder, B., Schreiner, M., Laham, D., Wolfe, M., Landauer, T., and Kintsch, W. (1998). Using Latent Semantic Analysis to assess knowledge: Some technical considerations. *Discourse Processes*, 25:337–354.

[Wiemer-Hastings et al., 2004] Wiemer-Hastings, P., Allbritton, D., Efron, J., and Arnott, E. (2004). RMT: A dialog-based research methods tutor with or without a head. In *Proceedings of the ITS2004 Seventh International Conference*, Maciao, Brazil. Elsevier. Available at http://reed.cs.depaul.edu/peterwh/papers/its2004.pdf.

[Wiemer-Hastings et al., 1999] Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. (1999). How latent is Latent Semantic Analysis? In *Proceedings of the Sixteenth International Joint Congress on Artificial Intelligence*, pages 932–937, San Francisco. Morgan Kaufmann.

[Wiemer-Hastings and Zipitria, 2001] Wiemer-Hastings, P. and Zipitria, I. (2001). Rules for syntax, vectors for semantics. In *Proceedings of the* 23^{rd} *Annual Conference of the Cognitive Science Society*, Mahwah, NJ. Erlbaum.

[Wittgenstein, 1958] Wittgenstein, L. (1958). *Philosophical Investigations*. Blackwell, Oxford, 2nd edition. Translated by G. E. M. Anscombe.