Latent Semantic Analysis for Computer Vocabulary

Andrew Stout & Todd Gillette
Dept. of Computer Science
Swarthmore College
{stout,gillette}@cs.swarthmore.edu

Abstract

In the work described in this paper we undertook a fundamental, proof-of-concept exploration of an unsupervised technique for extracting and representing word meanings, called Latent Semantic Analysis. In this paper we detail an implementation of LSA for the purpose of vocabulary acquisition and report preliminary results from testing it on English vocabulary tests akin to the Test of English as a Foreign Language (TOEFL). We encountered several difficulties, which are also described.

1 Motivation

Most natural language processing research, and particularly most statistical natural language processing research, focuses on formal aspects of natural language: parsing and part-of-speech tagging are concerned with the rules of syntax, morphology is concerned with capturing and employing the formal rules of morphology in languages. While the machine translation task is certainly concerned with semantics, current approaches work mostly by considering formal structures and formal methods for word-alignment. In general, rather little attention is given to developing computational systems for understanding the *meaning* of natural language. We feel that in order to achieve the near-human level of natural language competence which must be the ultimate goal of Natural Language Processing, NLP research must confront the problem of meaning.

At its core, the problem is a familiar one, at least in concept, to researchers in Artificial Intelligence: it is the problem of symbol grounding. How can a computer understand what, say, an apple is, that is what the word "apple" means, if it is unable to experience fruit in any of the ways humans do, and by which humans ground their understanding of the symbol in natural language meaning apple? A computer has no experience of taste, or hunger, or touch, or even sight in most NLP settings. Clearly, a conventional computer's understanding of "apple" is going to be fairly limited, but it need not be limited to merely "noun, for plural add -s, Apfel in German", etc. What computers do have access to is a large volume of text, from which subtle statistical features and constraints can be gleaned.

Significant and fruitful (no pun intended) work has been carried out in the construction of semantic ontologies such as WordNet (Fellbaum, 1998), which capture important information about the semantics of words. However, such efforts are dependent on a great deal of human labor, subject to annotator bias, language-specific, and above all still lack a lot of information about the relationships between various words, categories, concepts, etc. What we seek is an unsupervised algorithm which constructs a representation allowing the computer to ground its understanding of the word "apple" in its 'experience' of 'reading' about apples in a vast amount of unannotated text.

2 Latent Semantic Analysis

Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997) is a statistical method for quantifying meaning in natural language, predicated on the notion that the entirety of the contexts in which a word does or does not appear provide a set of constraints capturing the meaning of that word. The basic approach is to represent words as high-dimensional vectors, where similarity in meaning can be calculated as a function of the difference between two vectors.

Our goal in the work described in this paper was to engage in a fundamental exploration of LSA, for the purpose of solidifying our own understanding of it, and to attempt to replicate the encouraging results of others' early work on LSA.

The remainder of this paper is organized as follows.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 92–96
Computer Science Department, Swarthmore College

The next section describes related previous work and attempted or potential applications of LSA. Section 4 contains a detailed description of the LSA algorithm, and an explanation of our implementation. Section 5 explains our testing of our system, and section 6 analyzes our results. Finally, we conclude with a more general discussion in section 7.

3 Related Work

3.1 Method

The Latent Semantic Analysis method was pioneered by Landauer and Dumais (Landauer and Dumais, 1997), and is closely related to Latent Semantic Indexing (Deerwester et al., 1990). The computational basis for LSA is described in (Berry et al., 1993). The method is described in section 4 below.

3.2 Applications

Many potential and already realized applications of LSA exist. Perhaps the most fundamental is simply to apply LSA in order to obtain a computational representation of meanings of words. More advanced work on understanding has also been undertaken, such as understanding metaphors (Kintsch and Bowles, 2002) (Kintsch, 2000). LSA has been trained on aligned corpora in different languages to be applied to machine translation (Landauer et al., 1998b), although LSA by itself has no production capability. By averaging the vectors of a paper and comparing to multiple other source documents one can determine which sources were most useful to the writer of the paper. By modeling the meaning of a large document, sentences which most closely match the meaning of the whole document can be identified for the purposes of summarization (Kintsch et al., 2000). By comparing the vector representations of one sentence to the next, LSA can provide a measure of cohesion within a text (Landauer et al., 1998b). LSA systems can also be trained to estimate the evaluation of an essay, and to choose an appropriate next text to maximize student learning (Landauer et al., 1998b). LSA has also been integrated into computer tutoring systems (Wiemer-Hastings, 2002).

The most exciting applications of LSA, however, are those yet to come: LSA offers the potential to develop NLP systems that understand the meaning of language and can process or do useful things–from better automatic translation to passing the Turing test–based on that understanding.

4 Implementation

4.1 Corpus

Latent Semantic Analysis learns word meanings through processing a large volume of unannotated training text; this corpus corresponds to what the system 'knows' after training. We started LSA runs on a subset of the Encyclopedia Britannica (EB) containing 4,148 unique words, factoring out all one letter words, two letter words, and words that appeared in only one document. There were 588 documents corresponding to encyclopedia entries, each of which was approximately a paragraph long. After running through the system with the Encyclopedia Britannica corpus, we chose the 1989 Associate Press corpus for our full-size corpus, using the natural division of an article as a document. The corpus contains over 98,000 documents and over 470,000 unique words.

4.2 Preprocessing

Once the training corpus has been divided into documents, our system builds a large matrix X of word occurrences. Each entry $x_{i,j}$ corresponds to the number of times word i appears in document j. Each entry is then transformed to "a measure of the first order association of a word and its context" by the equation

$$\frac{\log(x_{i,j}+1)}{-\sum_j \left(\left(\frac{x_{i,j}}{\sum_j x_{i,j}} \right) \cdot \log \left(\frac{x_{i,j}}{\sum_j x_{i,j}} \right) \right)}$$

(Landauer et al., 1998b).

4.3 Singular Value Decomposition

The mathematical underpinning of LSA is a linear algebraic technique called Singular Value Decomposition (SVD), which is a form of eigenvector-eigenvalue analysis which states that any rectangular matrix X can be decomposed into three matrices W, S, and C which, when multiplied back together, perfectly recreate X:

$$X = WSC^T$$

where

- X is any w by c rectangular matrix
- W is a w by m matrix with linearly independent columns (also called *principle components* or *singular vectors*)
- C is a m by c matrix with linearly independent columns
- S is a m by m diagonal matrix containing the *singular values* of X

WSC^T is guaranteed to perfectly recreate X provided m is as large as the smaller of w and c. Integral to LSA is the fact that if one of the singular values in S is omitted, along with the corresponding singular vectors of W and C, the resulting reconstruction $W'S'C'^T = X'$ is the best least-squares approximation of X given the remaining dimensions. By deleting all but the n largest singular

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 92–96
Computer Science Department, Swarthmore College

values from S, SVD can be used to compress the dimensionality of W. When X is a words by documents matrix, the compressed W' is called the *semantic space* of the training corpus from which it was generated.

Before compression, the matrices used for Latent Semantic Analysis are very large: $100,000 \times 98,000 = 8.9 \times 10^9$ elements. However, as any document will only contain a few hundred distinct words, the matrices are very sparse. As we discovered the hard way, cookbook algorithms for Singular Value Decomposition are completely impractical for matrices of this size. Fortunately, libraries for large sparse matrix Singular Value Decomposition do exist (Berry et al., 1993).[1]

It has been found empirically (Landauer et al., 1998a) that an n (dimension) of between 300 and 400 is generally the optimal level of compression to synthesize knowledge for LSA. The exact optimal number of dimensions depends on the particular corpus.

A diagram of our system's architecture is shown in Figure 1.

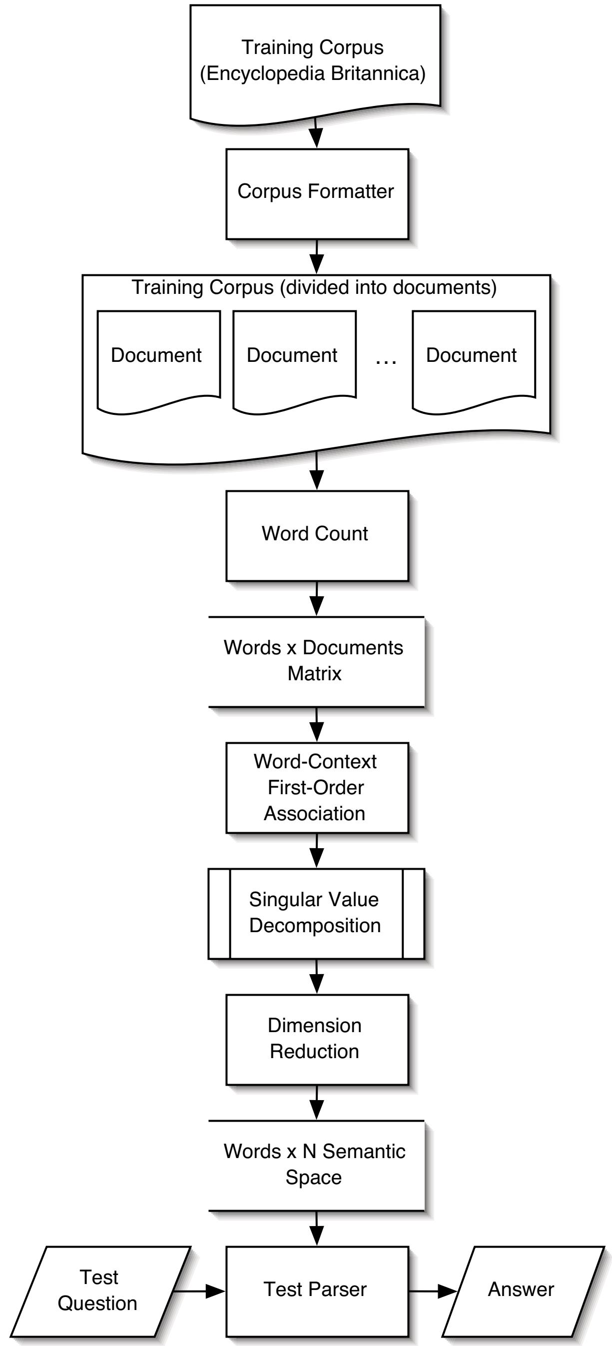

Figure 1: System Architecture

5 Testing

Since we were engaging in a fundamental, proof-of-concept exploration of LSA's potential as a means of capturing semantic information, we chose to test our system on simple vocabulary tests inspired by the Test of English as a Foreign Language (TOEFL) and manually generated using WordNet synonyms from the lexicon of the AP corpus. An example question is shown in Figure 2.

A question is evaluated in the following manner. The LSA vectors of each word in the target phrase ("*levied*" in the example in Figure 2) are averaged to give an average meaning for the phrase. The possible options are then averaged in the same way. Each possible answer is then compared to the target using cosine similarity: the cosine (or, equivalently, normalized dot product) between the target vector and each option vector is computed, and the largest cosine indicates the option with meaning most similar to the target.

6 Results

We tested the Encyclopedia Britannica corpus semantic space on questions developed for the AP corpus. As we expected, the results were poor due to the small size of the EB corpus and the mis-match between training and testing. Many of the words were not even in the EB corpus, and the few questions that had both question and answers produced poor results. We have so far been unable

[1]Unfortunately, the best of them was originally written in FORTRAN, uses counterintuitive data file formats, has completely incomprehensible source code, and is poorly documented (length and mathematical rigor of the accompanying "manual" notwithstanding).

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 92–96
Computer Science Department, Swarthmore College

For the following question, choose the answer phrase whose meaning most closely matches the meaning of the target phrase.
Target: *levied*
Possible answers:
a. *imposed*
b. *believed*
c. *requested*
d. *correlated*

Figure 2: A sample test question.

to complete the preprocessing on the AP corpus so we do not have results for that data set.

6.1 Complications

Our initial tests on a small (5000 word) subset of the Brown corpus made it clear that the computational complexity of the SVD algorithm would be a serious obstacle. Standard cookbook SVD recipes (Press et al., 1997) (which is what we were using) were simply not viable for matrices of the size required for LSA, and both the memory requirements and running time proved to be prohibitive.

The sparse matrix SVD package (Berry et al., 1993) provided an solution for the SVD problem, however a new problem arose when considering a corpus large enough to produce good results. The time necessary to count, process, and compress over 100,000 unique words and nearly 100,000 documents was too great to fit into the time available. We suspect that the first steps will take something on the order of a day or two, but it is still unknown how long the SVD will take. Given more time it might be feasible to carry out the test to completion, but we must at this time be satisfied with tests on a smaller corpus.

7 Conclusion

Latent Semantic Analysis has many useful applications and has been implemented successfully, however there are many difficulties in producing an effective implementation. Once a corpus has been processed and a semantic space has been generated, using it is very efficient and, based upon the work of others, effective.

There is still the question of size. How big must a corpus be to have a certain accuracy? Answering such a question would be more time consuming than testing other aspects and uses of LSA. We do know that the size of the EB corpus was too small, though it was processed very quickly (less than an hour). A corpus of a size in between our EB and AP corpora could be appropriate for testing LSA given the resources of our work space.

8 Future Work

With extra time, within reason, it will be possible to test the AP corpus on the questions. This will allow for a true test of the success of our implementation of LSA. In addition, a subset of the AP corpus might provide positive results in a shorter time frame. In a practical sense, much of our short-term future work would need to be devoted to efficient processing of the extremely large amount of data required for effective Latent Semantic Analysis.

8.1 Domain Specific LSA

Another solution would be to use a corpus of reduced size, though one larger than the EB corpus. Domain specific corpora might provide a means to test the system effectively in a more timely manner. If the lexicon itself were effectively reduced by reducing the scope of the domain, a smaller corpus might turn out to be useful. Some domains would be relatively small, such as news articles about a specific topic, such as baseball. There is a relatively small, consistent lexicon centered around baseball involving bases, hits, home runs, etc. It may be that a relatively small selection of articles from newspapers and/or magazines would produce a working semantic space for the domain of baseball.

(Landauer et al., 1998a) used a psychology textbook in order to make a semantic space of psychology. An introductory textbook would in no way cover all of the terms and ideas contained within the study of psychology, but for the student first being introduced to the field it would suffice. The small semantic space could be used for grading homeworks, evaluating summaries, or comparing test scores with that of a student. It would be interesting to find out what size of a corpus would be necessary for various domains. It would also be interesting to see if a different number of dimensions are found to be optimal for different domains, as well as how dependent that number is on the specific corpus.

8.2 Analysis of Dimensions

Another area of interest that could be explored in the future is that of what the dimensions of the semantic space represent. By taking words that are thought to be strongly related and seeing if there is a particular dimension of which each has a large component, we might be able to find out how semantic space is representing our language. By extension this might offer some insight into our organization of thoughts with respect to language.

References

Michael Berry, Theresa Do, Gavin O'Brien, Vijay Krishna, and Sowmini Varadhan. 1993. SVDPACKC user's guide. University of Tennessee Computer Science Department Technical Report.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 92–96
Computer Science Department, Swarthmore College

S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407.

Christiane D. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Walter Kintsch and Anita Bowles. 2002. Metaphor comprehension: What makes a metaphor difficult to understand? *Metaphor and Symbol*, 17:249–262.

Kintsch, Steinhart, Stahl, and LSA research group. 2000. Developing summarization skills through the use of lsa-based feedback. *Interactive learning environments*, 8(2):7–109.

Walter Kintsch. 2000. Metaphor comprehension: A computational theory. *Psychonomic Bulletin & Review*, 7(2):257–299.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240.

Thomas Landauer, Peter Foltz, and Darrel Laham. 1998a. An introduction to latent semantic analysis. *Discourse Processes*, 25:259–284.

Thomas K. Landauer, Darrell Laham, and Peter Foltz. 1998b. Learning human-like knowledge by singular value decomposition: A progress report. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. The MIT Press.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 1997. *Numerical Recipies in C*. Cambridge University Press.

Peter Wiemer-Hastings. 2002. Adding syntactic information to LSA.

Appeared in: *Proceedings of the Class of 2003 Senior Conference*, pages 92–96
Computer Science Department, Swarthmore College