Latent semantic analysis

+11 Recommend this on Goog

Thomas K Landauer and Susan Dumais (2008), Scholarpedia, 3(11):4356. doi:10.4249/scholarpedia.4356 revision #91420 [link to/cite this article]

- **Dr. Thomas K Landauer**, University of Colorado at Boulder and Pearson Knowledge Technologies
- **Dr. Susan Dumais**, Microsoft Inc., One Microsoft Way, Redmond WA

Latent semantic analysis (LSA) is mathematical method for computer modeling and simulation of the meaning of words and passages by analysis of representative corpora of natural text. LSA closely approximates many aspects of human language learning and understanding. It supports a variety of applications in information retrieval, educational technology and other pattern recognition problems where complex wholes can be treated as additive functions of component parts.

Contents

Overview of purpose and method

Latent Semantic Analysis (also called LSI, for Latent Semantic Indexing) models the contribution to natural language attributable to combination of words into coherent passages. It uses a long-known matrix-algebra method, Singular Value Decomposition (SVD), which became practical for application to such complex phenomena only after the advent of powerful digital computing machines and algorithms to exploit them in the late 1980s. To construct a *semantic space* for a language, LSA first casts a large representative text corpus into a rectangular matrix of words by coherent passages, each cell containing a transform of the number of times that a given word appears in a given passage. The matrix is then decomposed in such a way that every passage is represented as a vector whose value is the sum of vectors standing for its component words. Similarities between words and words, passages and words, and of passages to passages are then computed as dot products, cosines or other vector-algebraic metrics. (For *word* and *passage* in the above, any objects that can be considered parts that add to form a larger object may be substituted.)

LSA as a theory and model of language

The language-theoretical interpretation of the result of the analysis is that LSA vectors approximate the meaning of a word as its average effect on the meaning of passages in which it occurs, and reciprocally approximates the meaning of passages as the average of the meaning of their words. The derived relation between individual words should not be confused with surface *co-occurrence*, the frequency or likelihood that words appear in the same passages: it is correctly interpreted as the similarity of the effects that the words have on passages in which they occur. That this kind of mutual constraint can be realized in other ways than SVD, for example with neural network models, recommends it as a potential theory of corresponding biological mechanisms in language and thought. The use of a large and representative language corpus supports representation of the meanings of new passages by statistical induction, thus comparison of the meaning of new passages to each other whether containing literal words in common or not, and thence to a wide range of practical applications. For example, the sentences *Cardiac surgeries are quite safe these days* and *Nowadays, it is not at all risky to operate on the heart* have very high similarity (cos =.76, p<< 0001). In LSA, there is no notion of multiple discrete senses or disambiguation prior to passage meaning formation. A word-form type has the same effect on every passage in which it occurs, and that in turn is the average of the vectors for all of the passages in which it occurs. Thus, a word vector represents a mixture of all its *senses*, in proportion to the sum of their contextual usages.

Technical description of LSA

Term-Passage Matrix

A large collection of text statistically representative of human language experience is first divided into passages with coherent meanings, typically paragraphs or documents. The collection is then represented as a term-passage matrix. Rows stand for individual terms and columns stand for passages or documents (or other units of analysis of interest.) Individual cell entries contain the frequency with which each term occurs in a document.

Transformed Term-Passage Matrix

The entries in the term-document matrix are often transformed to weight them by their estimated importance in order to better mimic the human comprehension process. For language simulation, the best performance is observed when frequencies are cumulated in a sublinear fashion within cells (typically $log(freq_{ij} + 1)$, where $freq_{ij}$ is the frequency of term i in document j), and inversely with the overall occurrence of the term in the collection (typically using inverse document frequency or entropy measures).

Stop-listing and stemming

These are very rarely used. In keeping with the underlying theory and model, neither stemming nor stop-listing is appropriate or usually effective. As in natural language, the meaning of passages cannot be accurately reconstructed or understood without all of its words. However, when LSA is used to compare word strings shorter than normal text paragraphs, e.g. short sentences, zero weighting of function words is often pragmatically useful.

Dimension reduction

A reduced-rank singular value decomposition (SVD) is performed on the matrix, in which the k largest singular values are retained, and the remainder set to 0. The resulting representation is the best k-dimensional approximation to the original matrix in the least-squares sense. Each passage and term is now represented as a k-dimensional vector in the space derived by the SVD. In most applications k the dimensionality is much smaller than the number of terms in the term-passage matrix. For most language simulation k , >50 and <1,000 dimensions are optimal, with 300 +/- 50 most often best, although there is neither theory nor method to predict the optimum. It has been conjectured that in many cases, such as language simulation, that the optimal dimensionality is intrinsic to the domain being simulated and thus must be empirically determined. The dimension reduction step performs a powerful induction: a different value for the similarity of every word to every other whether or not they have ever occurred in a common context.

The mathematics

Consider a rectangular t x p matrix of terms and passages, X . Any rectangular matrix can be decomposed into the product of three other matrices using the singular value decomposition. Thus,

$$X = T * S * P^T \tag{1}$$

is the SVD of a matrix X where T is a t x r matrix with orthonormal columns, P is a p x r matrix with orthonormal columns, and S is an r x r diagonal matrix with the entries sorted in decreasing order. The entries of the S matrix are the singular values (eigenvalue$^{.5}$), and the T and P matrices are the left and right singular vectors, corresponding to term and passage vectors. This is simply a re-representation of the X matrix using orthogonal indexing dimensions. LSA uses a truncated SVD, keeping only the k largest singular values and their associated vectors, so

$$X = T_k * S_k * P_k^T \tag{2}$$

is the reduced-dimension SVD, as used in LSA. This is the best (in a least squares sense) approximation to X with k parameters, and is what LSA uses for its semantic space. The rows in T_k are the term vectors in LSA space and the rows in P_k are the passage vectors.

Similarity in the reduced space

Since both passages and terms are represented as vectors, it is straightforward to compute the similarity between passage-passage, term-term, and term-passage. In addition, terms and/or passages can be combined to create new vectors in the space. The process by which new vectors can be added to an existing LSA space is called folding-in. The cosine distance between vectors is used as the measure of their similarity for many applications because of its relation to the dot-product criterion and has been found effective in practice, although other metrics, such as Euclidean or city-block distances are sometimes used. To accurately update the SVD and thereby take into account new term frequencies and/or new terms requires considerable computation; minor perturbations to the original term-by-document matrix X can produce different term and passage vectors (and therefore affect precision and recall for query matching).

Applying LSA

In applying LSA successfully to language simulations, the most important factor is to have an appropriate and large enough text (or other) corpus. As a very rough rule of thumb, corpora supplying less than ~ 20K word types in less than ~ 20K passages are likely to yield faulty results. Vector precision in the results of two bytes is usually sufficient; 300 dimensions is almost always near optimal, ~ 200-2,000 usually within a useful range.

Application guidelines

Stemming

Every derivative form of a word-form type will have a different vector that is appropriately similar or dissimilar to others, modulo training success. Stemming often confabulates meanings, e.g. *flower* becomes *flow'. By contrast, LSA assigns a different meaning to each variant and a continuous similarity value between each pair: for example, flower-flow have cos = -.01, dish-dishes cos = .68. The meanings associated with words by LSA can be intuitively appreciated by examining cosine similarities with other words in a typical LSA semantic space, for example:* Flower: petals .93, gymnosperms 0.47 Flow: flows .84, opens 0.46 Dish: sauce 0.70, bowl 0.63 Dishes: kitchen 0.75, cup 0.57

Recomputing, folding in, computational limitations

Once a semantic space has been created from a sufficiently large and representative sample, it is usually unnecessary to re-compute it in order to add new terms or passages. For new terms, it is usually sufficient to compute the vector value of a new term as the average of the vectors of all (~ >10) passages in which it has occurred. Again very roughly, unless a corpus of over ~ 20K passages has changed by about 20% or more, recomputing the SVD will have insignificant effects on old vectors. Because new passage vectors are computed as the sum of their word vectors, the same rule of thumb applies to their addition to a semantic space. These observations along with the vast increases in computing power and the advent of highly parallel

SVD algorithms have answered early concerns about the practicality of LSA based on the assumption that frequent retraining would be necessary. The same advances have made training on far larger corpora possible; for example a 500 million word corpus in 2007 took less than a day on a modest-sized cluster.

SVD and LSA packages

Free SVD and LSA packages include SVDPACK/SVDPACKC which use iterative methods to compute the SVD of large sparse matrices. Some have special features for particular applications and thus might be worth considering, especially for non-language applications. As of 2008, there were many useful tools and semantic spaces available at http://lsa.colorado.edu/, but continuance is not guaranteed. LSI++ is a commonly used toolkit for information retrieval applications. Computational complexity of the sparse iterative methods is $O(z + k)$, where z is the number of non-zero entries per passage and k is the number of dimensions in the reduced space.

Theory, experience and experiments all indicate that semantically and topically coherent passages of ~ 50-100 words, for example paragraphs from newspapers or text books, are optimal units for training, but that substantial variation is well tolerated.

Typical language simulation applications

LSA has been used most widely for small database IR and educational technology applications. In IR test collections when all other features (e.g. stemming, stop-listing, and term-weighting) of comparison methods are held constant, LSA gives combined precision and recall results around 30% better than others. Its strength is in recall because of its independence of literal word overlap. Its lack of wider use in IR appears to be due to widely over-estimated training and retraining requirements. LSA's best-known educational applications are as the primary component in automatic essay grading systems that equal human readers in accuracy and in summary writing and other computer tutors. It has been the basis of technologies to improve indexing, to assess the coherence and content sequencing of books, diagnose psychological disorders, match jobs and applicants, monitor and enhance team communications and other applications. It has been used as the basis of a metric for the developmental status of words as a function of the amount of language encountered. It has been used as a tool for experiments and as a component of theories and applications in psychology, anthropology, sociology, psycholinguistics, data mining and machine learning.

Non-English and cross-language applications

LSA has been used successfully in a wide variety of languages. These include all the U.N. and European Union languages, Chinese and Japanese (in Chinese character representations where the sum of components assumption holds over different complexity of components), Swahili, Hindi, Arabic and Latvian. Highly inflected and word-compounding languages have been surprisingly amenable so long as sufficiently large and topic-covering training corpora are used. One demonstration of linguistic and anthropological/philosophical interest, as well as practical value, of LSA's multiple language capability comes from cross-language information retrieval. In this method, independent LSA spaces in two or more languages are first created from single language corpora in which several hundred passages are direct translations or topically close corresponding texts in the other languages. Then the different language spaces are rotated by the least squares Procrustes method so that the common passages are best aligned. Tested by similarity of one random passage to the other of translated pairs not used in the alignment, recall and precision are within normal ranges for single-language IR.

Linguistic and philosophical implications

Plato, Chomsky, Pinker and others have claimed that neither grammar nor semantics can be learned from exposure to language because there is too little information in experience, so must be primarily innate. LSA has shown that computational induction can extract much more information than previously supposed. The finding that words and passages of similar meaning as expressed in a wide variety of different languages can be mapped onto each other by a simple linear transform implies that the semantic structure of language may, in a sense, be universal—presumably because everywhere people must learn to talk mostly about the same things.

Shortfalls, objections, evidence and arguments

Potential limitations should be noted:

1. Training by text alone does not include oral language exposure, direct instruction by parents and teachers, and the association of language with perception and action. The amount of such loss for an exemplary application was as estimated as follows. In expert ratings of the quality of student essays, agreement between humans and a method using only LSA, the mutual information between LSA and humans was 90% as high as that between the humans.
2. LSA is blind to word order. A kind of upper bound on loss of information in this respect was estimated as follows The amount of information possible in sentences—where syntax and grammar have most of their influence—was divided into two parts. In a twenty-word sentence, the amount of information in the possible combinations of the 100,000 word types known by a literate adult is about nine times that in the possible permutations of the twenty words. Both these approaches suggest that LSA might be only about 10% inferior to humans, but they clearly do not go far enough. Humans also generate meaningful language, make and understand linguistically posed propositions, appreciate metaphors and analogies--among many other things.

Some commentators have also argued that LSA must be fundamentally wrong as theory because is not grounded in perception and intention. The strength of this objection is considerably reduced by the observation that language must veridically reflect these sources or it would be nearly useless, and by the human ability to generate, use and understand words as abstract and unrelated to perception as the word *abstract* itself, and by LSA's varied successes.

References

- Berry, M. W., Dumais, S. T. and O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. SIAM: Review, 37(4): 573-595.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41, 391-407.

- Foltz, P. W., Laham, D., and Landauer, T. K. (1999). The Intelligent Essay Assessor: Applications to educational technology. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning, 1(2). Online journal.
- Foltz, P. W. Kintsch, W. and Landauer T. K. (1998). The measurement of textual coherence with Latent Semantic Analysis. Discourse Processes, 25(2&3), 285-307.
- Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D.,Person, N., & the Tutoring Research Group. (2000). Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. Interactive Learning Environments, 129-148.
- Hofmann, T. (1999). Probabilistic latent semantic analysis. In Proceedings of 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 50-57.
- Kahana, M. J. (1996). Associative retrieval processes in free recall. Memory & Cognition,24, 103-109.
- Landauer, T. K., and Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240.
- Berry, M.W., and Browne, M. (2005). Understanding Search Engines: Mathematical Modeling and Text Retrieval, Second Edition, SIAM, Philadelphia.

Internal references

- Olaf Sporns (2007) Complexity. Scholarpedia, 2(10):1623.
- Tomasz Downarowicz (2007) Entropy. Scholarpedia, 2(11):3901.

Recommended reading

- Landauer, T. K., and Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240
- Berry, M. W., Dumais, S. T. and O'Brien, G. W. (1995). Using linear algebra for intelligent information retrieval. SIAM: Review, 37(4): 573-595.
- Landauer,T. K., McNamara, D.S., Dennis, S. and Kintsch W. (Eds). (2007) Handbook of Latent Semantic Analysis, Mahwah NJ: Lawrence Erlbaum Associates.

External links

- Thomas Landauer's Website (http://www.pearsonkt.com/bioLandauer.shtml)

See also

Sponsored by: Eugene M. Izhikevich, Editor-in-Chief of Scholarpedia, the peer-reviewed open-access encyclopedia

Sponsored by: Prof. Ke CHEN, School of Computer Science, The University of Manchester, U.K.

Reviewed by (http://scholarpedia.org/w/index.php?title=Latent_semantic_analysis&oldid=46314) : Anonymous

Accepted on: 2008-08-30 16:38:37 GMT (http://scholarpedia.org/w/index.php?title=Latent_semantic_analysis&oldid=46314)

Categories: Pattern Recognition | Artificial Intelligence | Multiple Curators