2009 IEEE International Conference on Semantic Computing

Generalizing Latent Semantic Analysis

Andrew M. Olney
Institute for Intelligent Systems
University of Memphis
Memphis, USA
Email: aolney@memphis.edu

***Abstract*—Latent Semantic Analysis (LSA) is a vector space technique for representing word meaning. Traditionally, LSA consists of two steps, the formation of a word by document matrix followed by singular value decomposition of that matrix. However, the formation of the matrix according to the dimensions of words and documents is somewhat arbitrary. This paper attempts to reconceptualize LSA in more general terms, by characterizing the matrix as a feature by context matrix rather than a word by document matrix. Examples of generalized LSA utilizing n-grams and local context are presented and compared with traditional LSA on paraphrase comparison tasks.**

***Keywords*-latent semantic analysis; vector space; n-gram; paraphrase;**

I. Introduction

At the most basic level, models of semantics try to capture the meanings of words. This somewhat then begs the question of where the words themselves get their meanings. A common example of this problem can be seen in dictionary entries. Any entry defines the meaning of a word in terms of ...other words! Therefore there is a certain amount of circularity in defining the meanings of words that is inescapable when working only with other words.

A natural way of looking at word meanings is whether two words have the same or opposite meanings, i.e. are synonyms or antonyms [1]. Words with these relations have the property that they may occur in the same contexts:

> John felt *happy* vs. John felt *sad*
> Mary *walked* across the street vs. Mary *ambled* across the street

Therefore some aspects of word meaning can be defined by looking at the contexts in which words occur. Words with similar contexts may have meaning relationships like synonym or antonym. This is particularly relevant in information retrieval, where there may be multiple ways of expressing the same query.

The vector space model is a statistical technique that represents the similarity between collections of words as a cosine between vectors [2], [3]. As such it can be used as the basis for a computational approach to word meaning or to information retrieval. The process begins by collecting text into a corpus. A matrix is created from the corpus, having one row for each unique word in the corpus and one column for each "document," usually a paragraph. The cells c_{ij} of the matrix consist of a simple count of the number of times $word_i$ appeared in $document_j$. Since many words do not appear in any given document, the matrix is often sparse. Weightings are applied to the cells that take into account the frequency of $word_i$ in $document_j$ and the frequency of $word_i$ across all documents, such that distinctive words that appear infrequently are given the most weight.

Using the vector space model, two collections of words of arbitrary size are compared by creating two vectors. Each word is associated with a row vector in the matrix, and the vector of a collection is simply the sum of all the row vectors of words in that collection. Vectors are compared geometrically by the cosine of the angle between them. From this description it should be clear that the vector space model is both unsupervised and generative, meaning that new collections of words, i.e. sentences or documents, can be turned into vectors and then compared with any other collection of words.

Latent semantic analysis (LSA) [4]–[7] is an extension of the vector space model that uses singular value decomposition (SVD). SVD is a technique that creates an approximation of the original word by document matrix. After SVD, the original matrix is equal to the product of three matrices, word by singular value, singular value by singular value, and singular value by document. The size of each singular value corresponds to the amount of variance captured by a particular dimension of the matrix. Because the singular values are ordered in decreasing size, it is possible to remove the smaller dimensions and still account for most of the variance. The approximation to the original matrix is optimal, in the least squares sense, for any number of dimensions one would choose [8]. In addition, the removal of smaller dimensions introduces linear dependencies between words that are distinct only in dimensions that account for the least variance. Consequently, two words that were distant in the original space can be near in the compressed space, causing the inductive machine learning and knowledge acquisition effects reported in the literature [6]. This inductive property of LSA is what makes it useful for many applications, including approximating vocabulary acquisition in children [6], cohesion detection [9], grading essays [10], and understanding student contributions in tu-

978-0-7695-3800-6/09 $26.00 © 2009 IEEE
DOI 10.1109/ICSC.2009.89

torial dialogue [11], [12], entailment detection [13], and dialogue segmentation [14], amongst many others.

Recently several attempts have been made to generalize semantic spaces and related distributional approaches. Although LSA is a kind of semantic space, these previous attempts do not specifically address matrix representation and construction. Weeds & Weir [15] present a framework for distributional similarity metrics, derived from the information retrieval metrics of precision and recall. Thus their paper focuses on calculating distributional similarity, rather than matrix representation. Padó & Lapata [16] describe a methodology for constructing dependency-based semantic spaces that extends a previous formulation [17]. This work is complementary to the work presented here, in that it attempts to extend beyond the descriptive formalism of [17] to a constructive/generative methodology. However, the differences between the present work and [16] are threefold: the present work does not focus solely on syntax but rather attempts to generalize beyond syntax, is strongly focused on matrix construction, and specifically addresses questions raised by the LSA community regarding generalization and word order [7].

II. Generalizing Latent Semantic Analysis

We argue that the traditional word by document matrix is somewhat arbitrary, and that other matrix forms should be considered. These matrix forms may differ on one or both dimensions, either word or document. Consider the word dimension (rows). In LSA, a "bag of words" approach, the sentences "John called Mary" and "Mary called John" are equivalent, i.e. they are the same vector. This is so because the individual word vectors from each sentence are added together to make the sentence vectors, and addition in vector spaces is commutative, i.e. $1 + 4 = 4 + 1$ [18]. However, the semantic equivalence of "John called Mary" and "Mary called John" is clearly undesirable.

One proposal for incorporating word order into LSA is to create an n-gram by document matrix instead of a word by document matrix. In this model, a bigram vector for "tall man" is atomic, rather than the combination of "tall" and "man." By incorporating word order at the atomic level of a vector, word order is included without violating the vector space property of LSA that requires commutative addition of vectors. This model has been used for unsupervised latent semantic grammar induction [19]. This example alone suggests that it may be fruitful to consider other dimensions besides words.

With respect to the document dimension (columns), consider the classic formulation of distributional analysis [20]. In this approach, two elements, here words, are considered equivalent if they occur in the same context, with context defined as their left and right environments, e.g. the word on the left and right. This approach has been successfully applied to both unsupervised part of speech induction [21]–[25] and unsupervised grammar induction [26]–[29]. LSA can be thought of as a kind of distributional analysis, but with a different definition of context. With LSA, context is defined in the most relaxed sense, because left and right environments are equivalent: all that counts is that the word appears in the document at all. Again, this example suggests that other dimensions besides documents might be productive for some tasks.

The examples of n-grams and distributional contexts as replacements for words and documents respectively are just a few out of countless possibilities. What is common amongst all these matrices is that they are constructed according to the following procedure. First, some feature of interest is selected for which a vector representation is desired. In LSA, this feature is a word. However, it could be n-grams, part of speech, etc. Second, a data stream is identified that contains the feature of interest, and a sampling protocol for the data stream is defined. In LSA, the data stream is a corpus and the sampling protocol is tokenization into sentence, paragraph, or multi-page units. However, the sampling protocol could be a moving window as in the distributional analysis literature mentioned above and also in Burgess et al. [30]. Finally, this sampling protocol must be mapped to a schema representing the columns of the matrix. In LSA the schema is transparent: a sample (document) is a column. However, in a distributional schema representing left and right neighbor words, the mapping is slightly less obvious. Let α be the position before the target word (row word) and β be the position following the target word. If there are n unique words in the corpus, then both α and β have size n. Then there are $2n$ columns in the matrix for both of these positions: the first n columns for all words that occur before the target word, and the second n columns for all the words that occur after the target word. For example, the frequency of a specific word ("likes") at a specific position (α) relative to a target word ("John"), across the entire corpus, is in the cell ("John",α_{likes}). Assuming a sorted order on the columns for α and β, the corresponding frequency for "likes" in the β position is in the cell ("John",$\beta_{likes} = \alpha_{likes}+n$). Therefore, rather than a "word by document matrix," an appropriate description for generalized LSA would be a "feature by context" matrix, where context is dependent both on the sampling protocol and the mapping schema.

We briefly contrast the above approach with the framework proposed by Lowe [17] and later expanded by Padó & Lapata [16] to include syntactic dependency information. Lowe defines a semantic space as a quadruple $\langle A, B, S, M\rangle$, where B is a set of basis elements (e.g. words, here described as features), A is a function mapping co-occurrence frequencies (e.g. an identity map, here split between the sampling protocol and the mapping schema), S is a similarity measure (e.g. cosine, not addressed here), and M is a

Table I
NEAREST NEIGHBORS FOR $Context_{local}$ AT 50 DIMENSIONS

Term	Neighbor 1	Neighbor 2	Neighbor 3
massive	brilliant	beautiful	battered
introduced	conducted	cashed	brought+up
flower	compass	candle	brown+powder
ran	hurried	dashed	brings+it
the	passing+the	our+little	a+neighbouring
in	but+in	and+in	about+in
because	and+because	40+acres	11+grains
his	brady+'s	although+my	above+his
rabbit	mouse	frog	cat
eat	chew	approach+her	apply+that

transformation between semantic spaces (e.g. dimensionality reduction through SVD, not addressed here). In addition to the general lack of one-to-one correspondences between Lowe's analysis and that presented above, it should also be noted that the present analysis focuses very much on the matrix generation problem and alternative ways of conceptualizing this problem. Lowe's analysis, while informative, is most useful as a basis for comparing existing semantic space approaches rather than formulating new ones.

III. A CASE STUDY OF GENERALIZED LSA

The remainder of the paper presents two examples of the generalized LSA approach, using n-grams as features and two levels of context, both LSA and distributional. The first context approach, LSA, will be referred to as $Context_{global}$ because the context for each word spans the entire document. The second context approach, distributional, will be referred to as $Context_{local}$ because the context for each word consists of the neighbors surrounding that word. The comparison task for all methods is paraphrase detection. Therefore it is expected that the additional structure provided by n-grams and $Context_{local}$ will increase the ability of a model to make meaning judgments relative to a less structured baseline. In this experiment, both the $Context_{local}$ and $Context_{global}$ n-gram variations were compared to a standard LSA baseline.

A. Materials

The corpus used to generate the models (matrices and vector spaces) in the experiment was the concatenation of the TASA corpus and the WSJ10 corpus. The Touchstone Applied Science Associates (TASA) corpus [31] was used to provide a sufficiently large sample of text so that sufficient statistical information may be acquired. The TASA corpus consists of approximately 70MB of text, containing roughly 11 million word tokens. The TASA corpus was designed to represent random samples of texts read by students between the first grade to college in the United States [31]. Thus it represents a general corpus whose topics cover a wide range of areas. Second, a subset of the Wall Street Journal section of the Penn Treebank (WSJ10) [29], [32] was used to slightly enlarge the scope of topics covered by TASA. This corpus contains approximately 52,000 words drawn from Wall Street Journal articles in 1989, where each sentence is 10 words or less after punctuation has been removed.

The first evaluation dataset is the Microsoft Research Paraphrase Corpus [33]. This corpus contains approximately 220,000 words in 5,801 sentence pairs that have been judged to be either semantically equivalent or semantically non-equivalent. Three human raters produced judgments, with an average inter-rater reliability of 83%. Approximately 67% of the sentence pairs were found to be equivalent. Since the non-equivalent pairs overlap in both information content and wording, discrimination is a difficult task. All sentence pairs were extracted from online news sources over a period of several months.

The second evaluation dataset is the "20,000 Leagues Under the Sea" Corpus [34]. This corpus is derived from parallel translations of "20,000 Leagues Under the Sea". Aligned passages in the corpus are paraphrases of each other, and misaligned passages are not paraphrases. Thus, this corpus is concerned with the same general task as the Microsoft Research Paraphrase Corpus. This corpus contains approximately 370,000 words in 4,389 passage pairs. Approximately 50% of the passages are semantically equivalent. The non-equivalent passage pairs are "off-by-one," so that non-equivalent pairs have as much overlap in content and wording as do two subsequent passages from the corpus. Whereas the Microsoft Research Paraphrase Corpus has been designed to be difficult, the "20,000 Leagues Under the Sea" Corpus is a "natural" task.

B. Procedure

The first step is building the vector spaces. This was accomplished by first creating n-gram by context matrices, where the context varies as per Section III. The $Context_{global}$ approach uses an n-gram by document matrix. In this experiment, the matrix was constructed using all unigrams and bigrams that occur in more than one document. This restriction reduced the matrix size and eliminated terms that are unlikely to develop good representations after SVD; it is a heuristic employed by the BellCore LSI tools. The matrix had dimensions 756741 by 759561 and was generated using software written for this experiment.

The $Context_{local}$ approach uses an n-gram by local window matrix. The window used was 1 word to either side of the target word. This matrix was also constructed using unigrams and bigrams that occur more than once in the corpus. The local window was restricted to the word immediately before and immediately after the term. The matrix had dimensions 759353 by 268734 and was generated using the same software.

Table II
NEAREST NEIGHBORS FOR $Context_{global}$ AT 50 DIMENSIONS

Term	Neighbor 1	Neighbor 2	Neighbor 3
massive	by+conflict	a+massive	3.5+cents
introduced	first+used	first+discovered	especially+strong
flower	a+flower	a+brown	a+blue
ran	and+ran	and+hurried	and+gave
the	above+the	3,800.00+equal	11,600.00
in	and+in	again+in	activity+stock
because	and+because	although	accounting+knowledge
his	behind+his	and+his	ahab+chasing
rabbit	hypocrite	go+said	blanket+said
eat	buy	breathe	also+eat

The standard LSA approach, used as a baseline for comparison, uses a unigram by document matrix. This matrix was constructed using all unigrams that occur in more than one document in the matrix. This filtering kept the LSA approach consistent with the other two approaches. The matrix had dimensions 75640 by 759561. The matrix was also generated using the same software.

After the matrices have been constructed, each was transformed using SVD. A non-orthogonal SVD procedure was used due to the large sizes of the matrices: non-orthogonal SVD methods require less volatile memory than traditional orthogonal methods employed by the Bellcore LSI tools [35], [36]. Because this SVD procedure requires a number of Lanczos steps approximately proportional to the square of the number of dimensions desired, the number of target dimensions was limited to 100. This kept running time and storage requirements within reasonable limits, approximately 4 days to create and 70 gigabytes of disk storage for the $Context_{global}$ matrix alone. To obtain 300 dimensions for the same matrix would have required an estimated 36 days and 630 gigabytes of storage. Although 100 dimensions converged for the $Context_{global}$ matrix, only 83 dimensions converged for the $Context_{local}$ matrix.

After SVD, the left vectors were matched with their respective unigrams and bigrams. This matching is demonstrated by a selection of nearest neighbors for $Context_{local}$ and $Context_{global}$ in Table I and Table II. In both tables, there is semantic association between a word and its neighbors, although this association can become less clear with more distant neighbors, e.g. "because" and "11+grains". Since both tables present the neighbors of the same terms, the properties of the neighbors themselves may be compared. For example, the neighbors from $Context_{local}$ often share the same syntactic category as well as being similar semantically. The neighbors from $Context_{global}$, on the other hand, seem largely semantic. Though anecdotal, these tables serve to illustrate the semantic spaces created by this procedure.

For each sentence pair in the evaluation corpora, a cosine was calculated between the paired sentences. The method of generating this cosine varies according to the three approaches. For the standard LSA approach, the cosine was calculated in three steps. First, the vector for the first sentence in the pair was found by summing the vectors for each word in that sentence. Secondly, the vector for the second sentence in the pair was found by summing the vectors for each word in that sentence. Thirdly, the cosine between these two sentence vectors was calculated.

The n-gram approaches $Context_{local}$ and $Context_{global}$ have parallel steps for computing a cosine, for unigram and bigram vectors respectively. For each sentence in the pair, a unigram vector is calculated in the same way as for standard LSA. However, an additional bigram vector is calculated for each sentence in the pair as well. The bigram vector for a sentence is the sum of bigram vectors in that sentence. For example, if there are n words in the sentence, then there are $n-1$ bigrams. However, not all bigrams in the sentence may have occurred in the corpus input to SVD. In that case, the bigram is given a zero vector. Therefore a cosine is calculated using the two unigram vectors, and another cosine is calculated using the two bigram vectors.

After a value was generated for each sentence pair, according to the approaches outlined above, the sentence pairs were split into training and testing sets. For the 20K corpus, the training set consisted of 2,194 randomly selected pairs. The testing set consisted of 2,193 randomly selected pairs. For the MRPC, the training set consisted of 4,076 randomly selected pairs. The testing set consisted of 1,725 randomly selected pairs. Binary logistic regression models were fit using the cosine data generated by each approach as predictors (unigram or unigram/bigram cosines). This process was repeated for each set of training data, yielding a total of three models for each data set. These models were then used to predict the semantic equivalence of the sentence pairs in the testing set.

C. Results

Performance for each model was tabulated in 2 x 2 contingency tables. From these tables, precision, recall, and F-measure were calculated [2]. Results for testing on the MRPC are presented in the left columns of Table III. All methods performed equally well. However, it is significant that none of the methods outperformed a simple baseline. The baseline capitalizes on the greater frequency of semantically equivalent pairs in the testing corpus by always predicting semantic equivalence. Since all other methods cluster tightly around this baseline, it is clear that the semantic information available to each method is insufficient to solve the problem. In other words, the results in Table III suggest that the MRPC is too difficult to reveal any potential differences that might exist between the three methods used in this experiment.

Results for the 20,000 Leagues corpus are presented in the right columns of Table III. These results are distinct from the

Table III
RESULTS FOR THE MRPC AND 20,000 LEAGUES CORPORA

	MRPC			20K Leagues		
Method	Precision	Recall	F-measure	Precision	Recall	F-measure
Baseline	.66	1.00	.80	.49	1.00	.66
LSA	.69	.91	.78	.90	.87	.88
$Context_{global}$	.69	.91	.79	.90	.87	.88
$Context_{local}$	.69	.92	.79	.87	.86	.86

results for the MRPC in that the different methods are now differentiated from the baseline, but not from each other. LSA and the linear combinations of unigram and bigram vectors had indistinguishable scores, regardless of context. This is somewhat surprising given that in training, bigram vectors were significant predictors of semantic equivalence. On the training set for the $Context_{global}$ method, each unit increase in bigram vector cosine made semantic equivalence 2.43 times more likely. On the training set for the $Context_{local}$ method, the effect of bigram vector cosine is even higher: each unit increase in bigram vector cosine made semantic equivalence 3.98 times more likely.

D. Discussion

These results suggest two counterintuitive conclusions. The first of these is that additional semantic information about bigrams does not improve performance on comparative meaning tasks. This is somewhat counterintuitive, since the sentences "John eats candy" and "Candy eats John" have identical unigram cosines (1.0) but differing bigram cosines (0) in the $Context_{global}$ space. Clearly if an evaluation dataset consisted entirely of sentences like these, or of sentences where words have been arbitrarily scrambled, e.g. "John candy eats," then the bigram vectors would aid discrimination. However, it seems to be the case that for the MRPC and the 20K corpus, the information carried by the bigram vectors is largely redundant with the information carried by the unigram vectors. It is tempting, but unsupported by the data, to extrapolate beyond these two corpora to language in general. The claim is not being made that semantic bigram information is not useful for language in general. However, it is clear that for current comparative meaning tasks, as represented by these two corpora, bigram vectors are not useful.

This first result, though counterintuitive, has been previously predicted. Landauer [37] reports an experiment designed to show the unimportance of word order in LSA without explicitly including word order in the LSA model. The experiment is an essay grading task in which LSA ranked the essays, two human judges ranked the essays, and all three rankings were compared using mutual information. Human-human mutual information was .90, while the average human-LSA mutual information was .81, leading Landauer to conclude that 90% of the information was shared between LSA and the judges, leaving only 10% unaccounted for, perhaps due to word order. The equivalence between bigram and unigram results reported in the current paper is the first direct support of Landauer's hypothesis.

Another counterintuitive result is that $Context_{local}$ and $Context_{global}$ appear to be largely equivalent with respect to these two tasks. Recall that $Context_{global}$ defines context to be anywhere in a sentence, whereas $Context_{local}$ defines context as the word on the left and the word on the right (Section III). Previous research has tended to associate semantic features with $Context_{global}$ [31] and syntactic features more with $Context_{local}$ [38]. Therefore, it is curious that these two ways of defining context would lead to equivalent results on a comparative meaning task. The major difference between $Context_{local}$ as applied here and local context in previous approaches is the subsequent use of SVD. It is possible that through dimensionality reduction, SVD has distilled the latent semantic information in $Context_{local}$. This interpretation is supported by the findings of Burgess et al. [30]. Using a method very similar to $Context_{local}$ but without SVD, they found that sentence vectors fared poorly for making semantic judgments. This previous result is in stark contrast to the result obtained here, which indicates that $Context_{local}$ and $Context_{global}$ are approximately equal at making semantic judgments. Again, the most salient difference between $Context_{local}$ and the method of [30] is the application of SVD. The result obtained in this experiment has some practical relevance, since $Context_{local}$ has equivalent performance to LSA, which is patented [39].

IV. CONCLUSION

There are several theoretical and practical implications for generalized LSA given the case study presented in Section III. One theoretical implication is that the n-gram methods evaluated in this case study are sensitive to word order. This is the first time that a latent semantic method has been sensitive to word order, which has previously been cited as a weakness of LSA [40]. By being sensitive to word order, it is possible that the n-gram methods described in this experiment will extend latent semantic approaches to new problem areas and widen the research effort behind latent semantic approaches. This result supports the notion

that generalizing LSA is a productive and useful strategy worthy of further study.

However, a related implication is that sensitivity to word order may not be as useful a property as was previously believed. Indeed the results from this case study suggest that bigram vectors contribute very little new information with respect to what is already conveyed by unigram vectors. Thus for n-gram models explored, it may be that the range of applications is restricted to domains where the same set of words is often reordered to express different ideas. An example of such an application might be causality, where the order of words is more critical to the meaning of the sentence. As this paper presents the first comparison of regular LSA with LSA having word order (via bigrams) it is the first work to directly address this concern in the LSA community [37], but much more work needs to be done to clarify the importance of word order in semantic spaces.

In broader terms, feature selection is a problem for the generalized LSA approach discussed in this paper. By themselves, the vector space approach and SVD do not optimize features for a task. They merely create a subspace approximation that optimally approximates the original matrix, in a least squares sense. Thus they beg the question of whether the original matrix was based on irrelevant features or contexts for the task at hand. As demonstrated in the case study in Section III, features of presumable utility like n-grams can turn out to not be useful. Another way of framing this problem with respect to feature selection is whether the goal is to optimize reconstruction of the data, an unsupervised task, or make the most accurate predictions, a supervised task [41]. Clearly generalized LSA approaches address only the former and not the latter.

Perhaps the two most productive ways of selecting features for building generalized LSA models of other domains are utilizing domain knowledge and using standard feature selection techniques (e.g. Guyon & Elisseeff [41]). In both cases, it may be wise to attempt to validate the efficacy of the features once the matrix has been constructed but before the SVD step. By avoiding the SVD step, which is time-intensive to compute, many more sets of features can be evaluated within a set period of time. After the features have been selected, the SVD step will likely improve performance even more by reducing the noise in the data.

ACKNOWLEDGMENT

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080594 and by the National Science Foundation, through Grant BCS-0826825, to the University of Memphis. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education or the National Science Foundation.

REFERENCES

[1] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Introduction to WordNet: An on-line lexical database*," *International Journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[2] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 1999.

[3] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[4] S. C. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. A. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990. [Online]. Available: citeseer.ist.psu.edu/deerwester90indexing.html

[5] S. Dumais, "Improving the retrieval of information from external sources," *Behavior Research Methods, Instruments and Computers*, vol. 23, no. 2, pp. 229–236, 1991.

[6] T. K. Landauer and S. T. Dumais, "A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge," *Psychological Review*, vol. 104, pp. 211–240, 1997.

[7] T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, Eds., *Handbook of latent semantic analysis*. Lawrence Erlbaum, 2007.

[8] M. W. Berry, Z. Drmac, and E. R. Jessup, "Matrices, vector spaces, and information retrieval," *SIAM Rev.*, vol. 41, no. 2, pp. 335–362, 1999.

[9] P. W. Foltz, W. Kintsch, and T. K. Landauer, "The measurement of textual coherence with latent semantic analysis," *Discourse Processes*, vol. 25, no. 2&3, pp. 285–308, 1998.

[10] P. Foltz, S. Gilliam, and S. Kendall, "Supporting Content-Based Feedback in On-Line Writing Evaluation with LSA," *Interactive Learning Environments*, vol. 8, no. 2, pp. 111–127, 2000.

[11] A. Graesser, P. Wiemer-Hastings, K. Wiemer-Hastings, D. Harter, T. Tutoring Research Group, and N. Person, "Using Latent Semantic Analysis to Evaluate the Contributions of Students in AutoTutor," *Interactive Learning Environments*, vol. 8, no. 2, pp. 129–147, 2000.

[12] B. A. Olde, D. Franceschetti, A. Karnavat, and A. C. Graesser, "The right stuff: Do you need to sanitize your corpus when using Latent Semantic Analysis?" in *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*. Mahwah, NJ: Erlbaum, 2002, pp. 708–713.

[13] A. Olney and Z. Cai, "An orthonormal basis for entailment," in *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference*. Menlo Park, CA: AAAI Press, May 15-17 2005, pp. 554–559.

[14] ——, "An orthonormal basis for topic segmentation in tutorial dialogue," in *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Philadelphia: Association for Computational Linguistics, October 6-8 2005, pp. 971–978.

[15] J. Weeds and D. Weir, "Co-occurrence retrieval: A flexible framework for lexical distributional similarity," *Computational Linguistics*, vol. 31, no. 4, pp. 439–475, 2005.

[16] S. Pado and M. Lapata, "Dependency-based construction of semantic space models," *Computational Linguistics*, vol. 33, no. 2, pp. 161–199, 2007.

[17] W. Lowe, "Towards a theory of semantic space," in *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society*, 2001, pp. 576–581.

[18] O. Pretzel, *Error-Correcting Codes and Finite Fields*. Oxford: Clarendon Press, 1998.

[19] A. M. Olney, "Latent semantic grammar induction: Context, projectivity, and prior distributions," in *Proceedings of TextGraphs-2: Graph-Based Algorithms for Natural Language Processing*. Rochester, NY: Association for Computational Linguistics, April 2007, pp. 45–52.

[20] Z. Harris, "Distributional structure," *Word*, vol. 10, pp. 140–162, 1954.

[21] G. R. Kiss, "Grammatical word classes: A learning process and its motivation," in *The Psychology of Learning and Motivation: Advances in Research and Theory*, G. H. Bower, Ed. New York: Academic Press, 1973, ch. 1, pp. 1–41.

[22] S. Finch and N. Chater, "Bootstrapping syntactic categories," in *Proceedings of the 14th Annual Meeting of the Cognitive Science Society*. Mahwah, NJ: Lawrence Erlbaum Associates, July 29-August 1 1992, pp. 820–825.

[23] M. Redington, N. Chater, and S. Finch, "Distributional information and the acquisition of linguistic categories: A statistical approach," in *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum, June 18-21 1993, pp. 848–853. [Online]. Available: citeseer.ist.psu.edu/redington93distributional.html

[24] H. Schütze, "Part-of-speech induction from scratch," in *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*. Philadelphia: Association for Computational Linguistics, June 22-26 1993, pp. 251–258.

[25] ——, "Distributional part-of-speech tagging," in *Proceedings of the 7th European Association for Computational Linguistics Conference (EACL-95)*. Philadelphia: Association for Computational Linguistics, March 27-31 1995, pp. 141–149.

[26] E. Brill, "A simple rule-based part of speech tagger," in *Proceedings of the Third Conference on Applied Natural Language Processing*. ACL, 1992.

[27] A. Clark, "Unsupervised induction of stochastic context-free grammars using distributional clustering," in *Proceedings of the ACL 2001 Workshop on Computational Natural Language Learning*, W. Daelemans and R. Zajac, Eds. Philadelphia: Association for Computational Linguistics, July 6-7 2001, pp. 105–112.

[28] D. Klein and C. D. Manning, "A generative constituent-context model for improved grammar induction," in *Proceedings of the 40th Annual Meeting of the ACL*. Philadelphia: Association for Computational Linguistics, July 7-12 2002, pp. 128–135.

[29] ——, "Corpus-based induction of syntactic structure: Models of dependency and constituency," in *Proceedings of the 42nd Annual Meeting of the ACL*. Philadelphia: Association for Computational Linguistics, July 21-26 2004, pp. 478–485.

[30] C. Burgess, K. Livesay, and K. Lund, "Explorations in context space: Words, sentences, discourse," *Discourse Processes*, vol. 25, no. 2&3, pp. 211–257, 1998.

[31] T. Landauer, D. Laham, and P. Foltz, "Computer-based grading of the conceptual content of essays," 1998.

[32] D. Klein, K. Toutanova, H. T. Ilhan, S. D. Kamvar, and C. D. Manning, "Combining heterogeneous classifiers for word-sense disambiguation," in *WSD Workshop at ACL 40*, 2002.

[33] W. Dolan, C. Quirk, and C. Brockett., "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *Proceedings of the International Conference on Computational Linguistics*. Philadelphia: Association for Computational Linguistics, August 23-27 2004, pp. 350–356.

[34] T. Chung, "20,000 leagues under the sea paraphrase corpus," May 14 2006, http://www.isi.edu/natural-language/people/20k1.txt http://www.isi.edu/natural-language/people/20k2.txt http://www.isi.edu/natural-language/people/goldstandard.txt.

[35] A. M. Olney, "Unsupervised induction of latent semantic grammars with application to parsing," Ph.D. dissertation, University of Memphis, August 2006.

[36] J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Volume 1: Theory*. Philadelphia: Society for Industrial and Applied Mathematics, 2002.

[37] T. Landauer, "LSA as a Theory of Meaning," in *Handbook of latent semantic analysis*, T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, Eds. Lawrence Erlbaum, 2007, pp. 379–400.

[38] M. Redington, N. Chater, and S. Finch, "Distributional information: A powerful cue for acquiring syntactic categories." *Cognitive Science*, vol. 22, no. 4, pp. 425–469, 1998.

[39] S. C. Deerwester, S. T. Dumais, G. W. Furnas, R. A. Harshman, T. K. Landauer, K. E. Lochbaum, and L. A. Streeter, "Computer information retrieval using latent semantic structure," U. S. Patent No. 4,839,853, 1989.

[40] P. Wiemer-Hastings and I. Zipitria, "Rules for syntax, vectors for semantics," in *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Erlbaum, August 1-4 2001, pp. 1112–1117.

[41] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.