Latent Semantic Grammar Induction: Context, Projectivity, and Prior Distributions

Andrew M. Olney
Institute for Intelligent Systems
University of Memphis
Memphis, TN 38152
aolney@memphis.edu

Abstract

This paper presents latent semantic grammars for the unsupervised induction of English grammar. Latent semantic grammars were induced by applying singular value decomposition to n-gram by context-feature matrices. Parsing was used to evaluate performance. Experiments with context, projectivity, and prior distributions show the relative performance effects of these kinds of prior knowledge. Results show that prior distributions, projectivity, and part of speech information are not necessary to beat the right branching baseline.

1 Introduction

Unsupervised grammar induction (UGI) generates a grammar from raw text. It is an interesting problem both theoretically and practically. Theoretically, it connects to the linguistics debate on innate knowledge (Chomsky, 1957). Practically, it has the potential to supersede techniques requiring structured text, like treebanks. Finding structure in text with little or no prior knowledge is therefore a fundamental issue in the study of language.

However, UGI is still a largely unsolved problem. Recent work (Klein and Manning, 2002; Klein and Manning, 2004) has renewed interest by using a UGI model to parse sentences from the Wall Street Journal section of the Penn Treebank (WSJ). These parsing results are exciting because they demonstrate real-world applicability to English UGI. While other contemporary research in this area is promising, the case for real-world English UGI has not been as convincingly made (van Zaanen, 2000; Solan et al., 2005).

This paper weaves together two threads of inquiry. The first thread is latent semantics, which have not been previously used in UGI. The second thread is dependency-based UGI, used by Klein and Manning (2004), which nicely dovetails with our semantic approach. The combination of these threads allows some exploration of what characteristics are sufficient for UGI and what characteristics are necessary.

2 Latent semantics

Previous work has focused on syntax to the exclusion of semantics (Brill and Marcus, 1992; van Zaanen, 2000; Klein and Manning, 2002; Paskin, 2001; Klein and Manning, 2004; Solan et al., 2005). However, results from the speech recognition community show that the inclusion of latent semantic information can enhance the performance of their models (Coccaro and Jurafsky, 1998; Bellegarda, 2000; Deng and Khudanpur, 2003). Using latent semantic information to improve UGI is therefore both novel and relevant.

The latent semantic information used by the speech recognition community above is produced by latent semantic analysis (LSA), also known as latent semantic indexing (Deerwester et al., 1990; Landauer et al., 1998). LSA creates a semantic representation of both words and collections of words in a vector space, using a two part process. First,

TextGraphs-2: Graph-Based Algorithms for Natural Language Processing, pages 45–52,
Rochester, April 2007

a term by document matrix is created in which the frequency of word w_i in document d_j is the value of cell c_{ij}. Filters may be applied during this process which eliminate undesired terms, e.g. common words. Weighting may also be applied to decrease the contributions of frequent words (Dumais, 1991). Secondly, singular value decomposition (SVD) is applied to the term by document matrix. The resulting matrix decomposition has the property that the removal of higher-order dimensions creates an optimal reduced representation of the original matrix in the least squares sense (Berry et al., 1995). Therefore, SVD performs a kind of dimensionality reduction such that words appearing in different documents can acquire similar row vector representations (Landauer and Dumais, 1997). Words can be compared by taking the cosine of their corresponding row vectors. Collections of words can likewise be compared by first adding the corresponding row vectors in each collection, then taking the cosine between the two collection vectors.

A stumbling block to incorporating LSA into UGI is that grammars are inherently ordered but LSA is not. LSA is unordered because the sum of vectors is the same regardless of the order in which they were added. The incorporation of word order into LSA has never been successfully carried out before, although there have been attempts to apply word order post-hoc to LSA (Wiemer-Hastings and Zipitria, 2001). A straightforward notion of incorporating word order into LSA is to use n-grams instead of individual words. In this way a unigram, bigram, and trigram would each have an atomic vector representation and be directly comparable.

It may seem counterintuitive that such an n-gram scheme has never been used in conjunction with LSA. Simple as this scheme may be, it quickly falls prey to memory limitations of modern day computers for computing the SVD. The standard for computing the SVD in the NLP sphere is Berry (1992)'s SVDPACK, whose single vector Lanczos recursion method with re-orthogonalization was incorporated into the BellCore LSI tools. Subsequently, either SVDPACK or the LSI tools were used by the majority of researchers in this area (Schütze, 1995; Landauer and Dumais, 1997; Landauer et al., 1998; Coccaro and Jurafsky, 1998; Foltz et al., 1998; Bellegarda, 2000; Deng and Khudanpur, 2003). Using

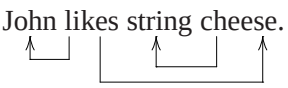

Figure 1: A Dependency Graph

the equation reported in Larsen (1998), a standard orthogonal SVD of a unigram/bigram by sentence matrix of the LSA Touchstone Applied Science Associates Corpus (Landauer et al., 1998) requires over **60 gigabytes** of random access memory. This estimate is prohibitive for all but current supercomputers.

However, it is possible to use a non-orthogonal SVD approach with significant memory savings (Cullum and Willoughby, 2002). A non-orthogonal approach creates the same matrix decomposition as traditional approaches, but the resulting memory savings allow dramatically larger matrix decompositions. Thus a non-orthongonal SVD approach is key to the inclusion of ordered latent semantics into our UGI model.

3 Dependency grammars

Dependency structures are an ideal grammar representation for evaluating UGI. Because dependency structures have no higher order nodes, e.g. *NP*, their evaluation is simple: one may compare with a reference parse and count the proportion of correct dependencies. For example, Figure 1 has three dependencies {(John, likes), (cheese, likes), (string, cheese) }, so the trial parse {(John, likes), (string, likes), (cheese, string)} has $1/3$ directed dependencies correct and $2/3$ undirected dependencies correct. This metric avoids the biases created by bracketing, where over-generation or undergeneration of brackets may cloud actual performance (Carroll et al., 2003). Dependencies are equivalent with lexicalized trees (see Figures 1 and 2) so long as the dependencies are projective. Dependencies are projective when all heads and their dependents are a contiguous sequence.

Dependencies have been used for UGI before with mixed success (Paskin, 2001; Klein and Manning, 2004). Paskin (2001) created a projective model using words, and he evaluated on WSJ. Although he reported beating the random baseline for that task, both Klein and Manning (2004) and we have repli-

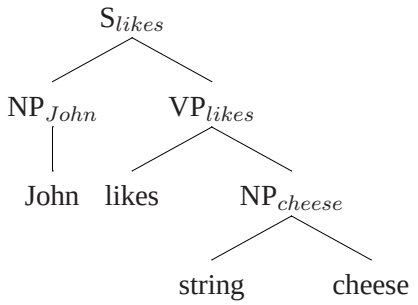

Figure 2: A Lexicalized Tree

cated the random baseline above Paskin's results. Klein and Manning (2004), on the other hand, have handily beaten a random baseline using a projective model over part of speech tags and evaluating on a subset of WSJ, WSJ10.

4 Unanswered questions

There are several unanswered questions in dependency-based English UGI. Some of these may be motivated from the Klein and Manning (2004) model, while others may be motivated from research efforts outside the UGI community. Altogether, these questions address what kinds of prior knowledge are, or are not necessary for successful UGI.

4.1 Parts of speech

Klein and Manning (2004) used part of speech tags as basic elements instead of words. Although this move can be motivated on data sparsity grounds, it is somewhat at odds with the lexicalized nature of dependency grammars. Since Paskin (2001)'s previous attempt using words as basic elements was unsuccessful, it is not clear whether parts of speech are necessary prior knowledge in this context.

4.2 Projectivity

Projectivity is an additional constraint that may not be necessary for successful UGI. English is a projective language, but other languages, such as Bulgarian, are not (Pericliev and Ilarionov, 1986). Nonprojective UGI has not previously been studied, and it is not clear how important projectivity assumptions are to English UGI. Figure 3 gives an example of a nonprojective construction: not all heads and their dependents are a contiguous sequence.

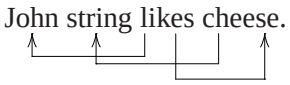

Figure 3: A Nonprojective Dependency Graph

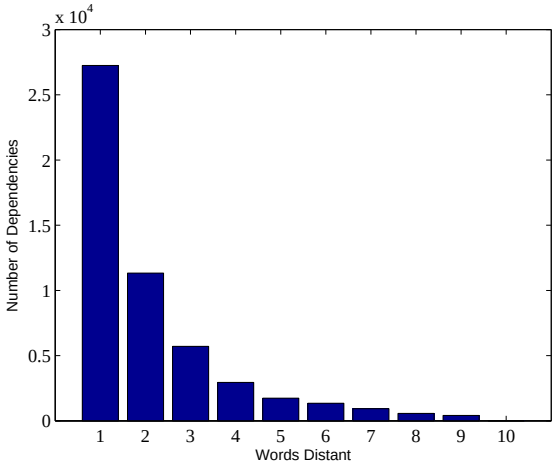

Figure 4: Distance Between Dependents in WSJ10

4.3 Context

The core of several UGI approaches is distributional analysis (Brill and Marcus, 1992; van Zaanen, 2000; Klein and Manning, 2002; Paskin, 2001; Klein and Manning, 2004; Solan et al., 2005). The key idea in such distributional analysis is that the function of a word may be known if it can be substituted for another word (Harris, 1954). If so, both words have the same function. Substitutability must be defined over a context. In UGI, this context has typically been the preceding and following words of the target word. However, this notion of context has an implicit assumption of word order. This assumption is true for English, but is not true for other languages such as Latin. Therefore, it is not clear how dependent English UGI is on local linear context, e.g. preceding and following words, or whether an unordered notion of context would also be effective.

4.4 Prior distributions

Klein and Manning (2004) point their model in the right direction by initializing the probability of dependencies inversely proportional to the distance between the head and the dependent. This is a very good initialization: Figure 4 shows the actual distances for the dataset used, WSJ10.

Klein (2005) states that, "It should be emphasized that this initialization was important in getting reasonable patterns out of this model." (p. 89). However, it is not clear that this is necessarily true for all UGI models.

4.5 Semantics

Semantics have not been included in previous UGI models, despite successful application in the speech recognition community (see Section 2). However, there have been some related efforts in unsupervised part of speech induction (Schütze, 1995). These efforts have used SVD as a dimensionality reduction step between distributional analysis and clustering. Although not labelled as "semantic" this work has produced the best unsupervised part of speech induction results. Thus our last question is whether SVD can be applied to a UGI model to improve results.

5 Method

5.1 Materials

The WSJ10 dataset was used for evaluation to be comparable to previous results (Klein and Manning, 2004). WSJ10 is a subset of the Wall Street Journal section of the Penn Treebank, containing only those sentences of 10 words or less after punctuation has been removed. WSJ10 contains 7422 sentences. To counteract the data sparsity encountered by using ngrams instead of parts of speech, we used the entire WSJ and year 1994 of the North American News Text Corpus. These corpora were formatted according to the same rules as the WSJ10, split into sentences (as documents) and concatenated. The combined corpus contained roughly 10 million words and 460,000 sentences.

Dependencies, rather than the original bracketing, were used as the gold standard for parsing performance. Since the Penn Treebank does not label dependencies, it was necessary to apply rules to extract dependencies from WSJ10 (Collins, 1999).

5.2 Procedure

The first step is unsupervised latent semantic grammar induction. This was accomplished by first creating n-gram by context feature matrices, where the feature varies as per Section 4.3. The $Context_{global}$ approach uses a bigram by document matrix such that word order is eliminated. Therefore the value of $cell_{ij}$ is the number of times $ngram_i$ occurred in $document_j$. The matrix had approximate dimensions 2.2 million by 460,000.

The $Context_{local}$ approach uses a bigram by local window matrix. If there are n distinct unigrams in the corpus, the first n columns contain the counts of the words preceding a target word, and the last n columns contain the counts of the words following a target word. For example, the value of at $cell_{ij}$ is the number of times $unigram_j$ occurred before the target $ngram_i$. The value of $cell_{i(j+n)}$ is the number of times $unigram_j$ occurred after the target $ngram_i$. The matrix had approximate dimensions 2.2 million by 280,000.

After the matrices were constructed, each was transformed using SVD. Because the non-orthogonal SVD procedure requires a number of Lanczos steps approximately proportional to the square of the number of dimensions desired, the number of dimensions was limited to 100. This kept running time and storage requirements within reasonable limits, approximately 4 days and 120 gigabytes of disk storage to create each.

Next, a parsing table was constructed. For each bigram, the closest unigram neighbor, in terms of cosine, was found, cf. Brill and Marcus (1992). The neighbor, cosine to that neighbor, and cosines of the bigram's constituents to that neighbor were stored. The constituent with the highest cosine to the neighbor was considered the likely head, based on classic head test arguments (Hudson, 1987). This data was stored in a lookup table so that for each bigram the associated information may be found in constant time.

Next, the WSJ10 was parsed using the parsing table described above and a minimum spanning tree algorithm for dependency parsing (McDonald et al., 2005). Each input sentence was tokenized on whitespace and lowercased. Moving from left to right, each word was paired with all remaining words on its right. If a pair existed in the parsing table, the associated information was retrieved. This information was used to populate the fully connected graph that served as input to the minimum spanning tree algorithm. Specifically, when a pair was retrieved from the parsing table, the arc from

the stored head to the dependent was given a weight equal to the cosine between the head and the nearest unigram neighbor for that bigram pair. Likewise the arc from the dependent to the head was given a weight equal to the cosine between the dependent and the nearest unigram neighbor for that bigram pair. Thus the weight on each arc was based on the degree of substitutability between that word and the nearest unigram neighbor for the bigram pair.

If a bigram was not in the parsing table, it was given maximum weight, making that dependency maximally unlikely. After all the words in the sentence had been processed, the average of all current weights was found, and this average was used as the weight from a dummy root node to all other nodes (the dummy ROOT is further motivated in Section 5.3). Therefore all words were given equal likelihood of being the root of the sentence. The end result of this graph construction process is an n by $n+1$ matrix, where n is the number of words and there is one dummy root node. Then this graph was input to the minimum spanning tree algorithm. The output of this algorithm is a non-projective dependency tree, which was directly compared to the gold standard dependency tree, as well as the respective baselines discussed in Section 5.3.

To gauge the differential effects of projectivity and prior knowledge, the above procedure was modified in additional evaluation trials. Projectivity was incorporated by using a bottom-up algorithm (Covington, 2001). The algorithm was applied in two stages. First, it was applied using the nonprojective parse as input. By comparing the output parse to the original nonprojective parse, it is possible to identify independent words that could not be incorporated into the projective parse. In the second stage, the projective algorithm was run again on the nonprojective input, except this time the independent words were allowed to link to any other words defined by the parsing table. In other words, the first stage identifies unattached words, and the second stage "repairs" the words by finding a projective attachment for them. This method of enforcing projectivity was chosen because it makes use of the same information as the nonprojective method, but it goes a step further to enforce projectivity.

Prior distributions of dependencies, as depicted in Figure 4, were incorporated by inversely weighting graph edges by the distance between words. This modification transparently applies to both the nonprojective case and the projective case.

Figure 5: Right Branching Baseline

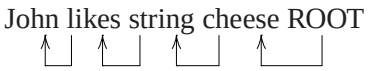

Figure 6: Left Branching Baseline

5.3 Scoring

Two performance baselines for dependency parsing were used in this experiment, the so-called right and left branching baselines. A right branching baseline predicts that the head of each word is the word to the left, forming a chain from left to right. An example is given in Figure 5. Conversely, a left branching baseline predicts that the head of each word is the word to the right, forming a chain from right to left. An example is given in Figure 6. Although perhaps not intuitively very powerful baselines, the right and left branching baselines can be very effective for the WSJ10. For WSJ10, most heads are close to their dependents, as shown in Figure 4. For example, the percentage of dependencies with a head either immediately to the right or left is 53%. Of these neighboring heads, 17% are right branching, and 36% are left branching.

By using the sign test, the statistical significance of parsing results can be determined. The sign test is perhaps the most basic non-parametric tests and so is useful for this task because it makes no assumptions regarding the underlying distribution of data.

Consider each sentence. Every word must have exactly one head. That means that for n words, there is a $1/n$ chance of selecting the correct head (excluding self-heads and including a dummy root head). If all dependencies in a sentence are independent, then a sentence's dependencies follow a binomial distribution, with n equal to the number of words, p equal to $1/n$, and k equal to the number of correct dependencies. From this it follows that the expected number of correct dependencies per sentence is np, or 1. Thus the random baseline for nonprojective depen-

dency parsing performance is one dependency per sentence.

Using the gold standard of the WSJ10, the number of correct dependencies found by the latent semantic model can be established. The null hypothesis is that one randomly generated dependency should be correct per sentence. Suppose that r^+ sentences have more correct dependencies and r^- sentences have fewer correct dependencies (i.e. 0). Under the null hypothesis, half of the values should be above 1 and half below, so $p = 1/2$. Since signed difference is being considered, sentences with dependencies equal to 1 are excluded. The corresponding binomial distribution of the signs to calculate whether the model is better than chance is $b(n, p) = b(r^+ + r^-, 1/2)$. The corresponding p-value may be calculated using Equation 1.

$$1 - \sum_{k=0}^{r^+-1} \frac{n!}{k!(n-k)!} 1/2(1/2)^{n-k} \qquad (1)$$

This same method can be used for determining statistically significant improvement over right and left branching baselines. For each sentence, the difference between the number of correct dependencies in the candidate parse and the number of correct dependencies in the baseline may be calculated. The number of positive and negative signed differences are counted as r^+ and r^-, respectively, and the procedure for calculating statistically significant improvement is the same.

6 Results

Each model in Table 6 has significantly better performance than item above using statistical procedure described in Section 5.2. A number of observations can be drawn from this table. First, all the models outperform random and right branching baselines. This is the first time we are aware of that this has been shown with lexical items in dependency UGI. Secondly, local context outperforms global context. This is to be expected given the relatively fixed word order in English, but it is somewhat surprising that the differences between local and global are not greater. Thirdly, it is clear that the addition of prior knowledge, whether projectivity or prior distributions, improves performance. Fourthly,

Method Context/Projectivity/Prior	Dependencies Correct
Random/no/no	14.2%
Right branching	17.6%
Global/no/no	17.9%
Global/no/yes	21.0%
Global/yes/no	21.4%
Global/yes/yes	21.7%
Local/no/no	22.5%
Local/no/yes	25.7%
Local/yes/yes	26.3%
Local/yes/no	26.7%
Left branching	35.8%

Table 1: Parsing results on WSJ10

projectivity and prior distributions have little additive effect. Thus it appears that they bring to bear similar kinds of constraints.

7 Discussion

The results in Section 6 address the unanswered questions identified in Section 4, i.e. parts of speech, semantics, context, projectivity, and prior distributions.

The most salient result in Section 6 is successful UGI without part of speech tags. As far as we know, this is the first time dependency UGI has been successful without the hidden syntactic structure provided by part of speech tags. It is interesting to note that latent semantic grammars improve upon Paskin (2001), even though that model is projective. It appears that lexical semantics are the reason. Thus these results address two of the unanswered questions from Section 6 regarding parts of speech and semantics. Semantics improve dependency UGI. In fact, they improve dependency UGI so much so that parts of speech are not necessary to beat a right branching baseline.

Context has traditionally been defined locally, e.g. the preceding and following word(s). The results above indicate that a global definition of context is also effective, though not quite as highly performing as a local definition on the WSJ10. This suggests that English UGI is not dependent on local linear context, and it motivates future exploration of word-order free languages using global context. It is

also interesting to note that the differences between global and local contexts begin to disappear as projectivity and prior distributions are added. This suggests that there is a certain level of equivalence between a global context model that favors local attachments and a local context model that has no attachment bias.

Projectivity has been assumed in previous cases of English UGI (Klein and Manning, 2004; Paskin, 2001). As far as we know, this is the first time a nonprojective model has outperformed a random or right branching baseline. It is interesting that a nonprojective model can do so well when it assumes so little about the structure of a language. Even more interesting is that the addition of projectivity to the models above increases performance only slightly. It is tempting to speculate that projectivity may be something of a red herring for English dependency parsing, cf. McDonald et al. (2005).

Prior distributions have been previously assumed as well (Klein and Manning, 2004). The differential effect of prior distributions in previous work has not been clear. Our results indicate that a prior distribution will increase performance. However, as with projectivity, it is interesting how well the models perform without this prior knowledge and how slight an increase this prior knowledge gives. Overall, the prior distribution used in the evaluation is not necessary to beat the right branching baseline.

Projectivity and prior distributions have significant overlap when the prior distribution favors closer attachments. Projectivity, by forcing a head to govern a contiguous subsequence, also favors closer attachments. The results reported in Section 6 suggest that there is a great deal of overlap in the benefit provided by projectivity and the prior distribution used in the evaluation. Either one or the other produces significant benefits, but the combination is much less impressive.

It is worthwhile to reiterate the sparseness of prior knowledge contained in the basic model used in these evaluations. There are essentially four components of prior knowledge. First, the ability to create an ngram by context feature matrix. Secondly, the application of SVD to that matrix. Thirdly, the creation of a fully connected dependency graph from the post-SVD matrix. And finally, the extraction of a minimum spanning tree from this graph. Although we have not presented evaluation on word-order free languages, the basic model just described has no obvious bias against them. We expect that latent semantic grammars capture some of the universals of grammar induction. A fuller exploration and demonstration is the subject of future research.

8 Conclusion

This paper presented latent semantic grammars for the unsupervised induction of English grammar. The creation of latent semantic grammars and their application to parsing were described. Experiments with context, projectivity, and prior distributions showed the relative performance effects of these kinds of prior knowledge. Results show that assumptions of prior distributions, projectivity, and part of speech information are not necessary for this task.

References

Jerome R. Bellegarda. 2000. Large vocabulary speech recognition with multispan statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1):76–84.

Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. 1995. Using linear algebra for intelligent information retrieval. *Society for Industrial and Applied Mathematics Review*, 37(4):573–595.

Michael W. Berry. 1992. Large scale singular value computations. *International Journal of Supercomputer Applications*, 6(1):13–49.

Eric Brill and Mitchell Marcus. 1992. Automatically acquiring phrase structure using distributional analysis. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York*, pages 155–160, Philadelphia, February 23-26. Association for Computational Linguistics.

John Carroll, Guido Minnen, and Ted Briscoe. 2003. Parser evaluation using a grammatical relation annotation scheme. In A. Abeill, editor, *Treebanks: Building and Using Syntactically Annotated Corpora*, chapter 17, pages 299–316. Kluwer, Dordrecht.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton, The Hague.

Noah Coccaro and Daniel Jurafsky. 1998. Towards better integration of semantic predictors in statistical language modeling. In *Proceedings of the International Conference on Spoken Language Processing*, pages 2403–2406, Piscataway, NJ, 30th November-4th December. IEEE.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Michael A. Covington. 2001. A fundamental algorithm for dependency parsing. In John A. Miller and Jeffery W. Smith, editors, *Proceedings of the 39th Annual Association for Computing Machinery Southeast Conference*, pages 95–102, Athens, Georgia.

Jane K. Cullum and Ralph A. Willoughby. 2002. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations, Volume 1: Theory*. Society for Industrial and Applied Mathematics, Philadelphia.

Scott C. Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

Yonggang Deng and Sanjeev Khudanpur. 2003. Latent semantic information in maximum entropy language models for conversational speech recognition. In *Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–63, Philadelphia, May 27-June 1. Association for Computational Linguistics.

Susan Dumais. 1991. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23(2):229–236.

Peter W. Foltz, Walter Kintsch, and Thomas K. Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25(2&3):285–308.

Zellig Harris. 1954. Distributional structure. *Word*, 10:140–162.

Richard A. Hudson. 1987. Zwicky on heads. *Journal of Linguistics*, 23:109–132.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, Philadelphia, July 7-12. Association for Computational Linguistics.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 478–485, Philadelphia, July 21-26. Association for Computational Linguistics.

Dan Klein. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.

Thomas K. Landauer and Susan T. Dumais. 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.

Thomas. K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. Introduction to latent semantic analysis. *Discourse Processes*, 25(2&3):259–284.

Rasmus M. Larsen. 1998. Lanczos bidiagonalization with partial reorthogonalization. Technical Report DAIMI PB-357, Department of Computer Science, Aarhus University.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530, Philadelphia, October 6-8. Association for Computational Linguistics.

Mark A. Paskin. 2001. Grammatical bigrams. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 91–97. MIT Press, Cambridge, MA.

Vladimir Pericliev and Ilarion Ilarionov. 1986. Testing the projectivity hypothesis. In *Proceedings of the 11th International Conference on Computational Linguistics*, pages 56–58, Morristown, NJ, USA. Association for Computational Linguistics.

Hinrich Schütze. 1995. Distributional part-of-speech tagging. In *Proceedings of the 7th European Association for Computational Linguistics Conference (EACL-95)*, pages 141–149, Philadelphia, March 27-31. Association for Computational Linguistics.

Zach Solan, David Horn, Eytan Ruppin, and Shimon Edelman. 2005. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, 102:11629–11634.

Menno M. van Zaanen. 2000. ABL: Alignment-based learning. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 961–967, Philadelphia, July 31-August 4. Association for Computational Linguistics.

Peter Wiemer-Hastings and Iraide Zipitria. 2001. Rules for syntax, vectors for semantics. In *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 1112–1117, Mahwah, NJ, August 1-4. Erlbaum.