An Exemplar-based Approach to Unsupervised Parsing

Simon Dennis (Simon.Dennis@adelaide.edu.au)
Department of Psychology; University of Adelaide
Adelaide, SA 5005 Australia

Abstract

We present an approach to syntactic processing based on the Syntagmatic Paradignmatic model (Dennis, in press) that assumes that the parse of a sentence can be viewed as a set of alignments with exemplars from memory. Alignment is achieved using a span-based version of the normalized edit distance measure (Marzal & Vidal, 1993), which is more appropriate for linguistic tasks. Span similarities used in the algorithm are derived using a version of the topics model (Griffiths & Steyvers, 2002) in which part-of-speech sequences are generated from their preceeding and postceeding word context. Approximate nearest neighbour exemplars are chosen using Locality Sensitive Hashing (Indyk & Motwani, 1998; Gionis, Indyk, & Motwani, 1999). Parses generated by the model are compared against gold standard parses from the Penn Treebank. The method provides state of the art precision and recall on this task and suggests that an unsupervised approach to parsing is feasible. Furthermore, the model is more directly comparable to exemplar-based accounts in other areas of cognition such memory and categorization than recursion-based approaches to syntax.

Keywords: syntax, parsing, unsupervised, syntagmatic paradigmatic model, edit distance

Introduction

The nativist/empiricist debate on the origin on language has been one of the longest and most hotly contested in the history of cognitive science (Pinker, 1994; Elman, 1999). On the one hand, languages are clearly learned at some level with a great many variations that differ in quite subtle ways. Furthermore, the difficulty in creating an explanation of how the genes might influence language development suggests that it is unlikely that our biological endowment has a direct influence (Elman, 1999). However, the fact that humans have a much more complex system of language than other primates, that there are similarities across the world's languages and that language acquisition takes similar paths in different cultures suggest a strong innate component (Pinker, 1994).

One key, if unstated, plank in the nativist case is that to this point no statistical learning procedure capable of capturing the syntax of a complex natural language has been devised (see Dennis, submitted; Klein & Manning, 2001). While connectionist models have demonstrated an ability to solve restricted problems with toy corpora (Elman, 1991), issues such as systematicity and constituent formation and movement remain unresolved (Hadley, 1994) seriously undermining the empiricist position.

In addition, from a practical perspective the inability to create syntactic analyses in an unsupervised fashion makes the application of natural language processing systems in new domains tedious. Either one must hand specify appropriate rules or one must create annotated corpora on which to train systems. Both of these tasks are difficult and time consuming.

In this paper, we outline attempts to improve an exemplar-based model of unsupervised parsing proposed by Dennis (submitted) using span-based normalized edit distance (SNED). We start by outlining the exemplar-based approach to parsing. Then we define normalized edit distance and the span-based modification, which is used to align neighbours against the target sentence. Then, we discuss how one can calculate the span similarities necessary to apply the method to sentences. Next, we describe a version of Locality Sensitive Hashing (Indyk & Motwani, 1998; Gionis et al., 1999) adapted to work with part of speech strings that allows the rapid selection of near neighbours. Finally, we present constituent recall and precision data on sentences drawn from the Penn Treebank (Marcus et al., 1993).

Exemplar-based Parsing

The algorithm that we employ for parsing sentences is a version of the Syntagmatic Paradigmatic model (Dennis, in press, 2004, submitted). The model has been used to account for a number of phenomena including long term grammatical dependencies and systematicity (Dennis, in press), the extraction of statistical lexical information (syntactic, semantic and associative) from corpora (Dennis, 2003a), sentence priming (Harrington & Dennis, 2003), verbal categorization and property judgment tasks (Dennis, in press), serial recall (Dennis, 2003b), and relational extraction and inference (Dennis, in press, 2004).

In this model, sentence parsing involves aligning near neighbour exemplar sentences from memory with the target sentence. For instance, suppose we wish to parse the sentence "His dog was big." (see

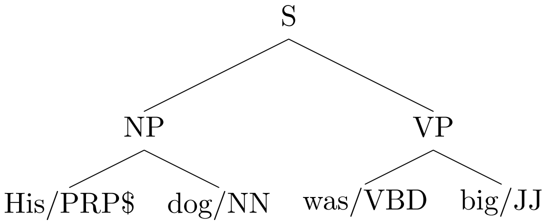


Figure 1. The correct parse of the sentence “His dog was big”. PRP$ = personal pronoun, NN = Noun, VBD = Verb, past tense, JJ = Adjective.

```
0.0000-|PRP$-NN-VBD-JJ|
       |PRP$-NN-VBD-JJ|
0.0011-|PRP$|NN|--VBD-JJ-|
       |PRP$|NN|MD-VB-VBN|
0.0017-|PRP$|NN|-VBD-JJ|
       |PRP$|NN|VBD-VBN|
0.0018-|PRP$-NN-|VBD|JJ|
       |DT-NN-NN|VBD|JJ|
```

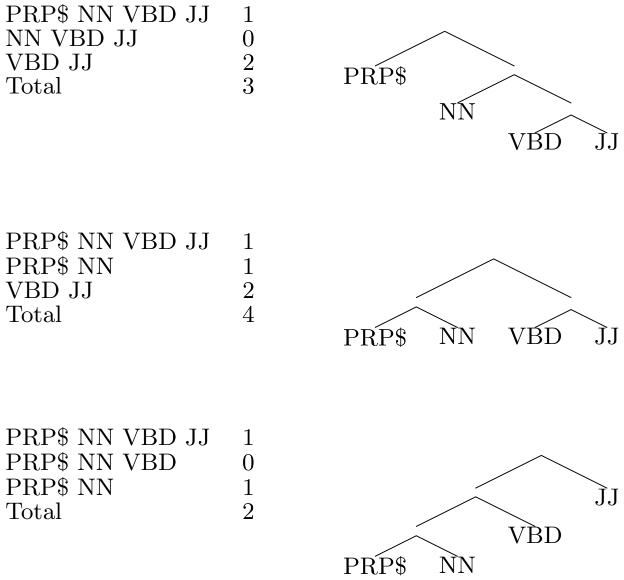
Figure 2. The minimum cost alignments and the corresponding costs for the four exemplars most similar to “His/PRP$ dog/NN was/VBD big/JJ”. Note aligning multiple exemplars against a target sentence can approximate a traditional parse. MD = Modal verb, VB = Verb, VBN = Verb, past participle, DT = Determiner.

Figure 1).

We start by converting the sentence to a part of speech (POS) sequence - “PRP$ NN VBD JJ”, where PRP$ = possessive pronoun, NN = noun, VBD = past tense verb and JJ = adjective. Next we identify near neighbour POS sequences from a large corpus and align each of these with the sentence (see Figure 2). In this case, we are using the 34,000 POS sequences that appeared at least twice in the first 350,000 sentences from the TASA corpus[1]. The number to the left of each alignment is the corresponding span-based edit distance (defined below). Note that these alignments induce constituent structure. In this case, for example, we would propose that PRP$-NN should constitute one constituent and VBD-JJ another.

While not constrained to be tree-like this structure may tend to correspond to a tree for many structurally unambiguous cases. However, for the purposes of testing against gold standard parses from the treebank, we induce a tree by determining the number of times each span of POS tags was identified by the model as a constituent. The binary parse with the highest total constituent count is then chosen using the obvious dynamic programming algorithm. In the example, the nonsingleton spans have the following counts:

PRP$ NN VBD JJ	1
PRP$ NN VBD	0
NN VBD JJ	0
PRP$ NN	1
PRP$ VBD	0
VBD JJ	2



PRP$ NN VBD JJ	1
NN VBD JJ	0
VBD JJ	2
Total	3

PRP$ NN VBD JJ	1
PRP$ NN	1
VBD JJ	2
Total	4

PRP$ NN VBD JJ	1
PRP$ NN VBD	0
PRP$ NN	1
Total	2

Figure 3. Possible binary parses of “His dog was big.” and their associated counts. In this case, parse number two would be chosen.

Figure 3 shows the three possible binary parses of the example sentence and the counts of the associated spans. In this case, the second parse would be chosen as it has the highest total span count.

Definitions of Edit Distances

The above algorithm relies on selecting alignments between POS tag sequences. In the following sections, we introduce the notion of span-based normalized edit distance (SNED) to play this role.

Edit Distance

Following the notation of Marzal and Vidal (1993), let Σ be a finite alphabet and Σ^* be the set of all finite-length strings over Σ. Let $X = X_1 X_2 ... X_n$ be a string of Σ^*, where X_i is the ith symbol of X. We denote by $X_{i...j}$ the substring of X that includes the symbols from X_i to $X_j, 1 \leq i, j \leq n$. The length of such a string is $|X_{i...j}| = j - i + 1$. If $i > j, X_{i...j}$ is the null string $\lambda, |\lambda| = 0$.

An elementary edit operation is a pair $(a, b) \neq (\lambda, \lambda)$, where a and b are strings of length 0 or 1. The edit operations are termed insertions (λ, b), substitutions (a, b) and deletions (a, λ). An edit

[1] We thank the late Stephen Ivens and Touchstone Applied Science Associates (TASA) of Brewster, New York for providing this valuable resource.

transformation of X into Y is a sequence S of elementary operations that transforms X into Y. Typically, edit operations have associated costs $\gamma(a,b)$. The function γ can be extended to edit transformations $S = S_1 S_2 ... S_l$ by letting $\gamma(S) = \sum_{i=1}^{l} \gamma(S_i)$.

Given $X, Y \in \Sigma^*$ and S^*_{XY} the set of all edit transformations of X into Y, then the edit distance is defined as:

$$\delta(X,Y) = min\{\gamma(S) | S \in S^*_{XY}\} \tag{1}$$

Note that the triangle inequality is a consequence of this definition, so provided $\gamma(a,a) = 0, \gamma(a,b) > 0$, if $a \neq b$, and $\gamma(a,b) = \gamma(b,a) \forall a,b \in \Sigma \cup \{\lambda\}$, δ is a metric.

Dynamic programming algorithms of complexity $O(mn)$, where n is the length of X and m is the length of Y, exist to calculate edit distance and to retrieve minimal edit transformations (Wagner & Fischer, 1974).

Normalized Edit Distance

Let $L(S)$ be the length of a given edit transformation. Then the *normalized edit distance* defined by Marzal and Vidal (1993) is:

$$d(X,Y) = min\{\gamma(S)/L(S) | S \in S^*_{XY}\} \tag{2}$$

Note that normalized edit distance is not a metric. It can, however, be calculated in $O(nm^2)$ time using an algorithm provided by Marzal and Vidal (1993).

Marzal and Vidal (1993) also show that NED does not produce the same answer as postnormalizing, by finding the minimum path and dividing by its length. Furthermore, for a handwritten character recognition task, normalized edit distance produced better performance than either standard edit distance or post normalized edit distance.

Span-based Normalized Edit Distance (SNED)

Any viable theory of sentence processing must account for the way in which people form constituents from series of words in sentences. The evidence for the phrasal structure of sentences is extensive and is of multiple types including phonological, morphonological, semantic and syntactic (see Radford, 1988, for a summary). Consequently, when analyzing sentence structure we would prefer a version of the normalized edit distance algorithm that aligns spans of symbols rather than individual symbols. Providing a definition of *span-based edit distance* involves relaxing the restriction in the standard algorithm, so that the strings a and b are drawn from Σ^*. So, the edit operations become $(a,b) = (X_{i...j}, Y_{k...l})$ for $0 \leq i \leq j \leq n$, $0 \leq k \leq l \leq m$. Similarly, one can define *span-based normalized edit distance* in an analogous way[2].

Spans	Contexts
PRP$ NN VBD JJ	SS:EE
PRP$ NN VBD	SS:big
NN VBD JJ	his:EE
PRP$ NN	SS:was
NN VBD	his:big
VBD JJ	dog:EE
PRP$	SS:dog
NN	his:was
VBD	dog:big
JJ	was:EE

Figure 4. Spans and associated contexts for the string His/PRP$ dog/NN was/VBD big/JJ. Note SS and EE are tags indicating the start and end of the sentence, respectively.

Calculating POS Span Costs

In order to apply the SNED algorithm one requires a γ function that indicates the cost of substituting one string of POS tags for another. To calculate substitution costs we combined ideas from the Context Constituent Model (CCM, Klein & Manning, 2001) and the Topics model (Griffiths & Steyvers, 2002). The Topics model is a probabilistic generative model in which documents (i.e. contexts) are assumed to generate topics which in turn generate words. A document, then, is defined by its mixture distribution of topics and a topic is defined by its mixture distribution of words. The Topics model assumes that these distributions are Dirichlet (see also Latent Dirichlet Allocation, Blei, Ng, & Jordan, 2002) and employs are Markov Chain Monte Carlo method to estimate the required conditional probabilities from a corpus (see Griffiths & Steyvers, 2002, for a deeper coverge of the model). In our application, we employ the same mechanism but assume that word contexts (i.e. the words immediately before and after a given span) generate topics which in turn generate POS spans[3]. For instance, the example sentence, His/PRP$ dog/NN was/VBD big/JJ, would generate the spans and contexts shown in Figure 4.

Note that spans that tend to substitutable for each other will have similar sets of contexts (see Redington, Chater, & Finch, 1998; Dennis, 2003a, for similar insights at the lexical level). For instance, we might expect the pattern of contexts in which we find VBD JJ to be similar to the pattern in which we find MD VBD VBN as they are both verb phrases.

Each POS span is associated with a distribution

[2] For the current purposes, we assume that $a, b \neq \lambda$ although it would be useful to draw a and b from $\Sigma^* \cup \{\lambda\}$ as an alternative formulation. We employ the obvious dynamic programming algorithm, which has time complexity $O(m^3 n^2)$, where m and n are the lengths of the strings.

[3] In the Context Constituent model both spans and contexts are defined in terms of POS tags. We found, however, that using words for the contexts improved performance.

over topics (i.e. $P(t|s)$ where t is the topic and s the span). Calculating a similarity between POS spans, then, can be achieved by comparing these distributions. A number of measures are possible. We chose to take the dot product of the distributions, which is equivalent to the probability that independent topic samples from each of the distributions would be identical:

$$\gamma(s_1, s_2) = \sum_i P(t_i|s_1)P(t_i|s_2)$$

Figure 5 shows a hierarchical cluster solution for the vectors corresponding to the 60 most frequent spans. Note that there is clear similarity structure with spans representing sentences, verb phrases, $\overline{N}$ and $\overline{\overline{N}}$ structures well separated.

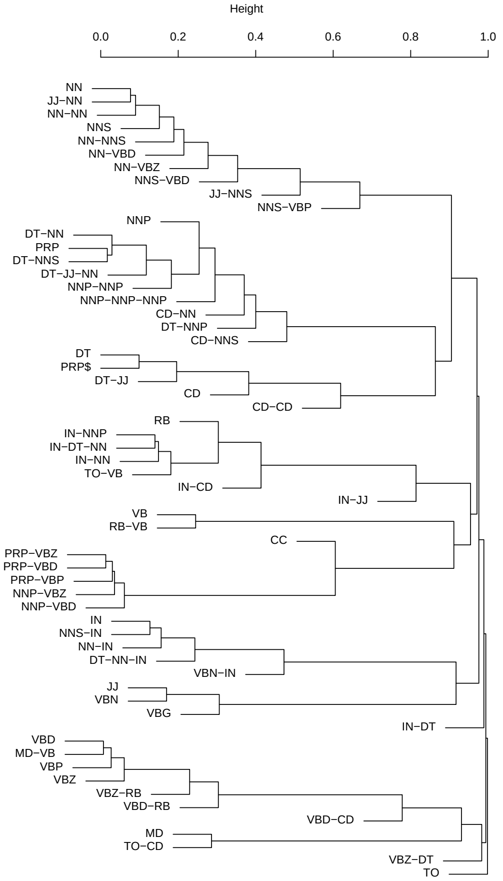


Figure 5. Hierarchiical clustering solution for the vectors corresponding to the 60 most frequent spans.

Finding Nearest Neighbors

A final issue to be resolved is how the algorithm selects nearest neighbour sequences to align. Given that there may be large numbers of potential sentences the performance of the nearest neighbour search will have a significant impact on the performance of the algorithm as a whole. In our case, it is sufficient to have a set of approximate nearest neighbours, so we use a version of Locality Sensitive Hashing (LSH, Indyk & Motwani, 1998; Gionis et al., 1999) adapted to work in Σ^* rather than in $\Re^d$ as is typical.

The basic idea of LSH is to create multiple hash functions each of which is designed so that similar sequences are likely to collide. Finding the nearest neighbours of a target string involves applying the hash functions to the new case and accumulating the strings that appear in the corresponding buckets.

To create the hash functions in Σ^* we create a set of rewrite rules that map one POS sequence to a simpler one. Different hash functions are created by permuting the rewrite rules. For example, suppose we have the exemplar sentence "Her little dolly felt sad.", which translates to PRP$ JJ NN VBD JJ, in our corpus and we wish to find the nearest neighbours of the target sentence "His dog was big." (PRP$ NN VBD JJ). Further, suppose that we have the following rewrite rules JJ NN → NN, PRP$ NN → NN, DT NN → NN. Let:

h1 = [JJ NN → NN, PRP$ NN → NN, DT NN → NN]
h2 = [PRP$ NN → NN, DT NN → NN, JJ NN → NN]

Now for the two strings we get the following keys:

Target

h1(PRP$ NN VBD JJ) = NN VBD JJ
h2(PRP$ NN VBD JJ) = NN VBD JJ

Exemplar

h1(PRP$ JJ NN VBD JJ) = NN VBD JJ
h2(PRP$ JJ NN VBD JJ) = PRP$ NN VBD NN

Because the two strings have a hash key in common, string two will be found when the system is queried with string one. In practice, locality sensitive hashing is fast and is not greatly affected by the size of the corpus. In our trials, we constructed a five hash system with hash functions containing 200 rewrite rules that were selected by taking the most similar POS span pairs that mapped a longer span into a shorter one.

Evaluating the model

The procedure outlined above was applied to all of the sentences from the Wall Street Journal section of the Penn treebank (Marcus et al., 1993) that were of length 10 or less. To assess performance the parses produced by the model were compared against the gold standard parses provided by the treebank. Three measures were calculated:

- Unlabelled Recall: The mean proportion of constituents in the gold standard that the model proposed.
- Unlabelled Precision: The mean proportion of constituents in the models answer that appear in the gold standard.
- F_1: The harmonic mean of unlabelled recall and unlabelled precision.

Because the treebank provides parses that are not binary, but the procedure used makes this assumption it is not possible to achieve perfect performance. Klein and Manning (2001) calculated that the best possible F_1 measure that can be achieved is 88.1%.

Figure 6 shows the performance of the model against chance selection of trees and against three versions of the Constituent Context Model (CCM) proposed by Klein and Manning (2001). Clearly, all of these models are performing well above chance with the performance of SNED close to other methods such as the context constituent model (Klein & Manning, 2001).

A key issue in the performance of the model is the number of nearest neighbours that are returned by the locality sensitive hashing algorithm. A significant number of sequences had no nearest neighbours and as a consequence performance on these examples is likely to be compromized. Figure 7 shows the impact of restricting the analysis to the items that return nearest neighbour sets of different sizes. If the SNED based algorithm is restricted to those sequences for which at least 30 neighbours are returned (i.e. SNED30) performance is close to the theoretically achievable maximum of 88.1%. Note, however, that one must interpret this figure carefully as it is also possible that there is a selection effect that is inflating these results.

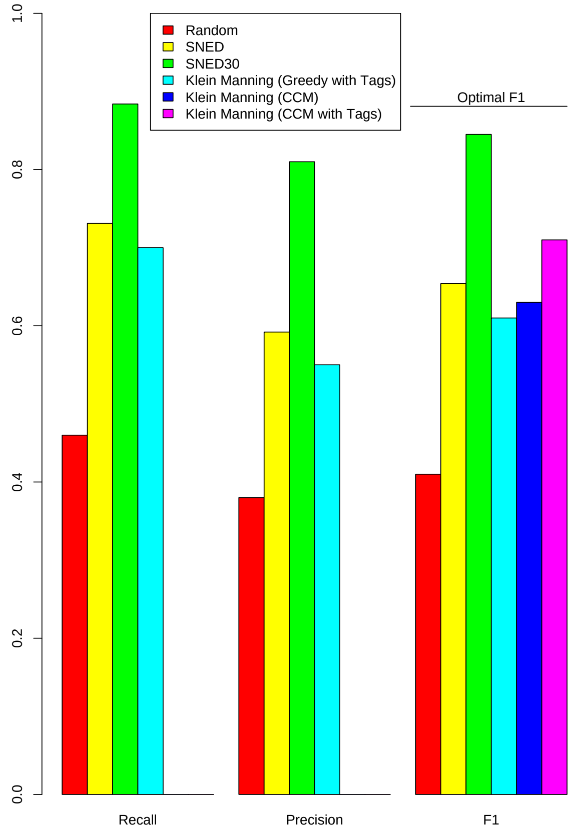


Figure 6. Results of Unsupervised Parsing Experiment. Note SNED 30 refers to performance calculated over those sequences for which there were at least 30 neighbours.

Conclusions

The version of the Syntagmatic Paradignmatic model (Dennis, in press, 2004) presented in this paper provides a demonstration of an exemplar-based approach to syntax. Many of the most influential models in memory (Shiffrin & Steyvers, 1997), learning (Logan, 1988), decision-making (Dougherty, 1999), phonology (Nakisa & Plunkett, 1998), lexical access (Goldinger, 1998), and categorization (Nosofsky, 1986) are exemplar-based, but models of this kind have played a much less significant role in cognitive models of syntax (although see Daelesmans, 1999, for examples in the computational lingusitics literature). Furthermore, the syntagmatic paradigmatic approach has been used to extract proposition-like information from corpora (Dennis, 2004, in press) and consequently seems to provide a useful unifying framework.

Perhaps more critically, however, the results presented in this paper demonstrate that significant grammaical structure can be extracted from a natural corpus, not just at the word level (Redington et al., 1998), but also at the word span level. This suggests that unsupervised parsing will be feasible thus withdrawing one of the key, if unstated, arguments in favour of the nativist account of language acquisition.

Acknowledgments

We would like to acknowledge the many discussions that have influenced the current work. In particular, we would like to thank Dan Jurafsky, Jim Martin, Walter Kintsch, Tom Landauer and Jose Quesada for helpful comments and suggestions. This research was supported by Australian Research Foundation Grant A00106012, U.S. National Science Foundation Grant EIA-0121201 and U.S. Department of Education Grant R305G020027.

References

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2002). Latent dirichlet allocation. In *Nueral information porcessing systems* (Vol. 14). Lawrence Erlbaum Associates.

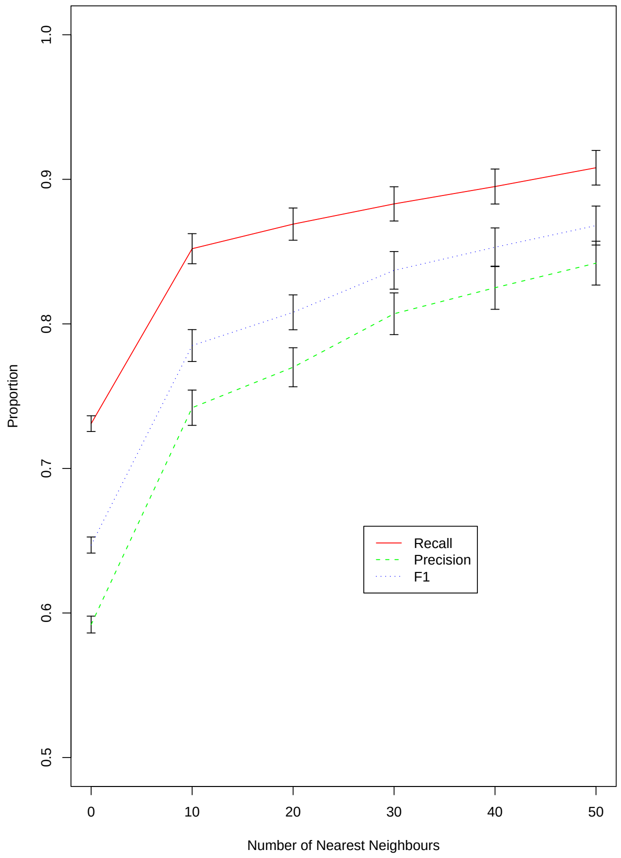


Figure 7. Performance as a function of the number of nearest neighbours. Bars indicate the 95% confidence intervals.

Daelesmans, W. (1999). Introduction to the special issue on memory-based language processing. *Journal of experimental and theoretical artificial intelligence*, *11*, 369-390.

Dennis, S. (2003a). An alignment-based account of serial recall. In *Twenty fifth conference of the cognitive science society* (Vol. 25). Lawrence Erlbaum Associates.

Dennis, S. (2003b). A comparison of statistical models for the extraction of lexical information from text corpora. In *Twenty fifth conference of the cognitive science society* (Vol. 25). Lawrence Erlbaum Associates.

Dennis, S. (2004). An unsupervised method for the extraction of propositional information from text. *Proceedings of the National Academy of Sciences*, *101*, 5206-5213.

Dennis, S. (in press). A memory-based theory of verbal cognition. *Cognitive Science.*

Dennis, S. (submitted). Introducing word order in an LSA framework. In *Latent Semantic Analysis: A road to meaning.*

Dougherty, M. R. P. (1999). Minerva-dm: A memory processes model for judgements of likelihood. *Psychological Review*, *106*(1), 180-209.

Elman, J. L. (1991). Distributed representations, simple recurrent networks and grammatical structure. *Machine Learning*, *7*, 195-225.

Elman, J. L. (1999). The origins of language: A conspiracy theory. In B. Mc Whinney (Ed.), *The emergence of language.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Gionis, A., Indyk, P., & Motwani, R. (1999). Similarity search in high dimensions via hashing. In *The VLDB journal* (p. 518-529).

Goldinger, S. D. (1998). Echoes of echoes? an episodic theory of lexical access. *Psychological Review*, *105*(2), 251-279.

Griffiths, T. L., & Steyvers, M. (2002). Prediction and semantic association. In *Nips.*

Hadley, R. F. (1994). Systematicity in connectionist language learning. *Mind and language*, *9*(3), 247-272.

Harrington, M., & Dennis, S. (2003). Structural priming in sentence comprehension. In *Twenty fifth conference of the cognitive science society* (Vol. 25). Lawrence Erlbaum Associates.

Indyk, P., & Motwani, R. (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In (pp. 604–613).

Klein, D., & Manning, C. D. (2001). Distributional phrase structure induction. In W. Daelemans & R. Zajac (Eds.), *Connl-2001* (p. 113-120). Toulouse, France.

Logan, G. D. (1988). Towards an instance theory of automatization. *Psychological Review*, *95*, 492-527.

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., et al. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, *19*(2), 313-330.

Marzal, A., & Vidal, E. (1993). Computation of normalized edit distance and applications. *IEEE trans. on Pattern Analysis and Machine Intelligence*, *15*(9), 926–932.

Nakisa, R. C., & Plunkett, K. (1998). Evolution for rapidly learned representations for speech. *Language and cognitive processes*, *13*(2/3), 105-127.

Nosofsky, R. M. (1986). Attention, similarity and the idenitification-categorization relationship. *Journal of Experimental psychology: General*, *115*, 39-57.

Pinker, S. (1994). *The language instinct: How the mind creates language.* New York: William Morro.

Radford, A. (1988). *Transformational grammar: A first course.* Cambridge: Cambridge University Press.

Redington, M., Chater, N., & Finch, S. (1998). Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive Science*, *22*(4), 425-469.

Shiffrin, R. M., & Steyvers, M. (1997). Model for recognition memory: Rem - retrieving effectively from memory. *Psychonomic Bulletin & Review*, *4*(2), 145-166.

Wagner, R. A., & Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the ACM*, *21*(1), 168-173.