Introducing Word Order within the LSA Framework

Simon Dennis
University of Adelaide

The importance of word order in determining meaning is a hotly debated issue. The success of Latent Semantic Analysis (LSA) across a broad range of tasks that require semantic analysis is a testament to the importance of word choice as compared to word order. Nonetheless at the sentence level, where LSA has been less successful, word order is sometimes the only way in which the role or sense in which a term is being used can be ascertained. Furthermore, a correct interpretation of a sentence often depends on combining words productively into phrasal or clausal units and LSA provides no insight into how this is achieved. In this chapter, we summarize attempts to capture the effects of word order with the Syntagmatic Paradigmatic model (SP, Dennis, in press, 2004). The SP model is an instance-based model that conforms to LSA principles; relying on large corpora and simple mathematical operations rather than formally specified linguistic rules or ad hoc heuristics. Unlike LSA, however, the SP model addresses both the propositional and syntactic levels of analysis.

The Importance of Word Order

Latent Semantic Analysis (LSA) is a ”bag of words” technique. A training corpus is represented as a word by document matrix. Weights are entered into the cells of the matrix based on the number of times a word appears in each document, but no attempt is made to capture the order in which the words appeared within the document. Furthermore, when constructing the meaning representation of a new document the vectors representing each unique word in the document are added, so again no attempt is made to capture word order. This insensitivity to word order has been raised as an important limitation of LSA.

However, Landauer, Laham, Rehder, and Schreiner (1997) have argued that the success of LSA in capturing human semantic judgments provides evidence that word choice rather than word order plays a more important role in assigning meaning. Landauer (2002) attempted to quantify the relative contributions of word order and word choice, based on the potential information available given human sized vocabularies and typical sentence lengths. He estimates that about 80% of the information content of a message could be contained in the word choice. In addition, many languages rely little on word order to convey meaning. For instance, the Warlpiri language of central Australia is almost completely free order (Kashket, 1986).

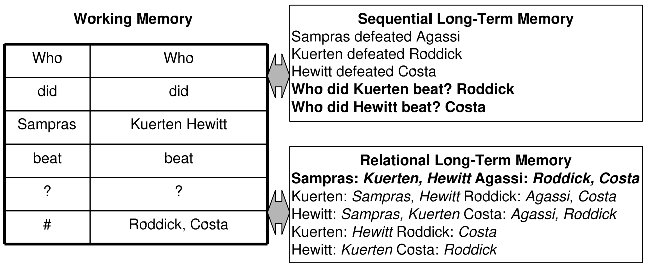
Nonetheless, there is still a significant contribution to meaning that is made by word order, particularly in English. Kintsch (2001) has shown how LSA can be extended to capture some of the effects of word order specifically in the areas of metaphor interpretation, causal inference, similarity rating and homophone discrimination. What remains to be explained, however, is how LSA style models can be made to capture the **propositional** information necessary to disambiguate who did what to whom, when and where. The sentence ”Mary loves John” does not express the same content as the sentence ”John loves Mary”. A complete understanding of the meaning of these sentences must make some distinction between ”John” as a lover and ”John” as a lovee and must explain how people are able to bind elements to these roles and extract them when necessary.

Furthermore, LSA representations focus on the lexical level and do not specify how words combine to form higher level units such as phrases and clauses. The evidence for such units is substantial (Radford, 1988) and role and sense assignments often apply to these constituents rather than individual words. Term extractions methods that chunk sequences of words so that they can be treated as units can be used as preprocessing stages to LSA. However, people are capable of forming constituents productively, so while term extraction may be practically useful it does not offer a complete explanation. That is, it does not provide an account of **syntactic** phenomena.

LSA can be thought of as a tool for analyzing language and a theory of meaning. In addition, however, LSA is an exemplar of a more general corpus-based approach to cognitive modeling. In this approach, simple statistical operations are applied to large naturally occuring corpora and no recourse is made to pre-existing knowledge (e.g. linguistic knowledge). So the question arises whether the limitations of LSA outlined above are just issues that have yet to be addressed versus stumbling blocks to the general framework.

The Syntagmatic Paradigmatic model (SP, Dennis, in press, 2004) attempts to answer this question by addressing both the propositional and syntactic levels of analysis using

I would like to acknowledge the many discussions that have influenced the current work. In particular, I would like to thank 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.



Working Memory

Who	Who
did	did
Sampras	Kuerten Hewitt
beat	beat
?	?
#	Roddick, Costa

Sequential Long-Term Memory
Sampras defeated Agassi
Kuerten defeated Roddick
Hewitt defeated Costa
Who did Kuerten beat? Roddick
Who did Hewitt beat? Costa

Relational Long-Term Memory
Sampras: *Kuerten, Hewitt* Agassi: *Roddick, Costa*
Kuerten: *Sampras, Hewitt* Roddick: *Agassi, Costa*
Hewitt: *Sampras, Kuerten* Costa: *Agassi, Roddick*
Kuerten: *Hewitt* Roddick: *Costa*
Hewitt: *Kuerten* Costa: *Roddick*

Figure 1. The Syntagmatic Paradigmatic (SP) architecture. The ”#” symbol indicates an empty slot. Ultimately, it will contain the answer to the question.

the general approach of LSA. In this chapter, the model will be described and work demonstrating its ability to capture propositional information will be summarized. The model will then be modified to operate over the starts and ends of words, rather than over words themselves. This modification allows the model to have a genuine sense of constituent and to directly address syntactic structure. The modified model was used to predict the phrase structure trees from preparsed sentences drawn from the Wall Street Journal corpus in the Penn Treebank.

The Syntagmatic Paradigmatic Model

The SP model was designed as a model of verbal cognition. It 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 section, we give a brief overview of the SP model. More complete descriptions, including the mathematical foundations, are provided by Dennis (in press, 2004).

In the SP model, sentence processing is characterized as the retrieval of associative constraints from sequential and relational long-term memory and the resolution of these constraints in working memory. Sequential long-term memory contains the sentences from a corpus. Relational long-term memory contains the extensional representations of the same sentences (see Figure 1).

Creating an interpretation of a sentence/utterance involves the following steps:

Sequential Retrieval: The current sequence of input words is used to probe sequential memory for traces containing similar sequences of words. In the example, traces four and five; ”Who did Kuerten beat? Roddick” and ”Who did Hewitt beat? Costa”; are the closest matches to the target sentence ”Who did Sampras beat? #” and are assigned high probabilities.

Sequential Resolution: The retrieved sequences are then aligned with the target sentence to determine the appropriate set of substitutions for each word. Note that the slot adjacent to the ”#” symbol aligns with the pattern Costa, Roddick. This pattern represents the role that the answer to the question must fill (i.e. the answer is the loser).

Relational Retrieval: The bindings of input words to their corresponding role vectors (the relational representation of the target sentence) are then used to probe relational long-term memory. In this case, trace one is favored as it involves similar role filler bindings. That is, it contains a binding of Sampras onto the Kuerten, Hewitt pattern and it also contains the Roddick, Costa pattern. Despite the fact that ”Sampras defeated Agassi” has a different surface form than ”Who did Sampras beat ? #” it contains similar relational information and consequently has a high retrieval probability.

Relational Resolution: Finally, the paradigmatic associations in the retrieved relational traces are used to update working memory. In the relational trace for ”Sampras defeated Agassi”, ”Agassi” is bound to the Roddick, Costa pattern. Consequently, there is a strong probability that ”Agassi” should align with the ”#” symbol which as a consequence of sequential retrieval is also aligned with the Roddick, Costa pattern. Note that the model has now answered the question - it was Agassi who was beaten by Sampras.

That completes the description of the basic model. An outstanding question, however, is how one decides how similar two strings of words are during sequential retrieval and how they should align during sequential resolution. Fortunately, there is a significant literature on this problem known as String Edit Theory (SET). In the next section, a brief overview of SET is provided.

String Edit Theory

When similar sentences are of the same length they can be aligned in a one to one fashion. If we are comparing John loves Mary and Bert loves Ellen, then John aligns with Bert, by virtue of the fact that John and Bert both fill the first slot of their respective sentences, and Mary aligns with Ellen because they both fill slot three.

However, in general this will not be the case. It is typical in natural languages for structure to be embedded. If we add a single adjective to our example, so that we are now comparing Little John loves Mary and Bert loves Ellen it becomes unclear how the sentences should align. What we require is a model of the alignment process.

One candidate for such a model is string edit theory. String edit theory was popularized in a book by Sankoff and Kruskal (1983) entitled, Time warps, string edits and macromolecules and has been developed in both the fields of computer science and molecular biology (Allison, Wallace, & Yee, 1992; Levenshtein, 1965; Needleman & Wunsch, 1970; Sellers, 1974). As the name suggests, the purpose of string edit theory is to describe how one string, which could be composed of words, letters, amino acids etc., can be edited to form a second string. That is, what components must be inserted, deleted or changed to turn one string into another.

As an example, suppose we are trying to align the sentences John loves Mary and Bert loves Ellen. The most obvious alignment is that which maps the two sentences to each other in a one to one fashion as suggested above:

```
John    loves   Mary
|       |       |
Bert    loves   Ellen
```

In this alignment, we have three edit operations. There is a change of John for Bert, a match of loves and a change of Mary for ”Ellen”.

Now if we add Little to the first sentence, we can use a deletion to describe one way in which the sentences could be aligned:

```
Little  John    loves   Mary
|       |       |       |
-       Bert    loves   Ellen
```

The - symbol is used to fill the slot left by a deletion (or an insertion) and can be thought of as the empty word. While these alignments may be the most obvious ones, there are many other options.

For instance, in aligning John loves Mary and Bert loves Ellen, we could start by deleting John:

```
John    loves   Mary    -
|       |       |       |
-       Bert    loves   Ellen
```

Note that Ellen is now inserted at the end of the alignment. Alternatively, we could have deleted John, and then inserted Bert to give:

```
John    -       loves   Mary
|       |       |       |
-       Bert    loves   Ellen
```

In total there are 63 ways in which John loves Mary can be aligned with Bert loves Ellen. Intuitively, the last alignment seems better than the one before because the word loves is matched. However, this alignment still seems worse than the first alignment because it requires John to be deleted and Bert to be inserted. A mechanism that produces alignments of sentences should favor those that have many matches and should penalize those that require many insertions and deletions. In probabilistic versions of SET probabilities are assigned to the edit operations. Probabilities of alignments are then determined by multiplying the probabilities of the edit operations of which they are composed. Matches are assigned higher probabilities and so alignments that contain many matches are considered more likely. Strings that have high probability alignments with the input sentence become active during sequential retrieval. Similarly, the probabilities of alignments are used to determine which words should align during sequential resolution.

Only a cursory explanation of SET has been possible in this chapter. Interested readers are referred to Sankoff and Kruskal (1983) and Allison et al. (1992). In addition, Dennis (in press) provides a more complete mathematical treatment in the context of the SP model including an explanation of how the edit model can be trained so that it does not rely on direct word overlap as is the case in the examples above.

Propositional Information

To illustrate the SP models ability to extract propositional information and answer simple questions from natural texts, Dennis (2004) exposed the model to sixty nine articles taken from the Association of Tennis Professionals (ATP) website at http://www.atptennis.com/ and then had the model answer questions of the form ”Who won the match between Sampras and Agassi?”.

The tennis news domain was chosen primarily because choosing the winner of a tennis match cannot be solved by appealing to simple type heuristics. Relevant source sentences often contain the names of both the winner and the loser so that the correct answer must be selected from items of the same type. Consequently, successful completion of this task requires a propositional analysis.

The model was trained on the ATP corpus and then each question was presented with the final answer slot vacant (e.g. ”Who won the match between Sampras and Agassi? #”). The SP model was invoked to complete the pattern. The word with the highest probability in the # slot was assumed to be the answer returned by the model. Following relational processing, on 67% of occasions the model correctly returned the winner of the match. 26% of the time it incorrectly produced the loser of the match. 5% of the time it responded with a player other than either the winner or loser of the match and on 3% of occasions it committed a type error, responding with a word or punctuation symbol that was not a players name. While this performance is far from perfect, it does demonstrate that the model is able to extract relational information from natural text at rates well above chance.

Another important aspect of the model is that it is often able to answer questions on the basis of sentences that imply the result, but do not directly state it; a property termed ”inference by coincidence”. To assess the contribution of this kind of inference, the sentence with maximal relational retrieval strength for each query was classified as either a literal statement of the result, a statement from which the result could be inferred or some other statement that did not entail the result. Of the 270 correct answers produced by the model, 79 fell into the literal category, 113 into the inference category and 78 into the other category. So for those traces for which a categorization could be made (i.e. the literal and inference categories) 59% were in the inference category. Given that in each case a literal statement of the results existed in the corpus it is significant that inference by coincidence seems to be playing such an important role in the performance of the model.

In summary, the ability of the SP model to isolate the combatants from arbitrary sentences and to successfully separate winners from losers demonstrates that it is capable of extracting propositional information from text. Unlike existing work in this domain (Blaheta & Charniak, 2000; Gildea & Jurafsky, 2002; Palmer, Rosenzweig, & Cotton, 2001), it need make no a priori commitment to particular grammars, heuristics or sets of semantic roles and it does not require an annotated corpus on which to train. In this way, the model conforms to the general LSA framework. Furthermore, the large number of occasions on which the most probable relational trace was a sentence from which the result could be inferred, but not directly derived, suggests that ”inference by coincidence” is a useful byproduct of extracting propositional information in this way.

Syntactic Information

In the work on proposition extraction described above each question required only a single word answer. More generally, however, answers will be constituents - Pete Sampras, the President of the United States of America, the dolly with the hat on etc. The edit model described above deals at the word level and is not well suited to capturing constituent information. To illustrate the problem, consider aligning the target sentence ”Bert who knows Ralph loves Ellen” against ”John loves Mary”.

```
Bert who knows Ralph loves Ellen
John --------------- loves Mary        A1
-------------- John  loves Mary        A2
```

Alignments A1 and A2 are equally probable as they involve the same numbers of each type of edit operation, that is, three deletions, two changes and one match. However, alignment A1 is preferable in terms of the propositional content being extracted. Furthermore, one might argue that it isn’t just any Bert who loves Ellen, but rather it is the Bert who knows Ralph who loves Ellen. That is, the alignment could be considered to be:

```
|Bert who knows Ralph|loves|Ellen|
|-------John---------|loves|Ellen|
```

Furthermore, if one were able to construct constituent alignments it becomes possible to see how an exemplar-based approach of this kind could approximate the more familiar tree analyses of formal linguistics. Figure 2 shows the alignment of several possible exemplars against the target sentnece ”Bert who knows Ralph loves Ellen”. The fact that John, Joe, Sofie, Al and Peter align with ”Bert who knows Ralph” could be taken to indicate that this span is a constituent and perhaps a noun phrase given that these words are nouns. Likewise the fact that flew, ran and cried align with ”loves Ellen” might indicate that this span is a verb phrase. Because there are mutiple exemplars all of which align (and possibly multiple alignments of a given exemplar) structure is induced. While not constrained to be tree-like this structure may tend to correpsond to a tree for many structurally unambiguous cases.

```
 Bert who knows Ralph loves Ellen
|------John----------|---flew----| NP VP
|------Joe-----------|---ran-----| NP VP
|------Sofie---------|---cried---| NP VP
|------Al------------|likes|Joan-| NP V N
|------Peter---------|knows|Barb-| NP V N
|Mike|who|--believes-|grows|corn-| N who VP V N
|Tom-|who|sees-|Lynn-|heard|Libby| N who V N V N
|Sam-|who|helps|Mum--|dates|Bill-| N who V N V N
```

Figure 2. Aligning multiple exemplars against a target sentence can approximate a traditional parse. N = Noun, V = Verb, NP = Noun Phrase and VP = Verb Phrase.

One way in which the SP model can be modified to generate these sorts of alignments is by focusing on the gaps between the words (as characterized by the words on either side) rather than on the words themselves. That is, the sentence ”Bert who knows Ralph loves Ellen” becomes:

```
S|Bert Bert|who who|knows knows|Ralph Ralph|loves loves|Ellen Ellen|E
```

where S is a start of sentence symbol and E is an end of sentence symbol. Now if we align the gaps in the same way that we aligned words we get the following alignment:

```
S|Bert Bert|who who|knows knows|Ralph Ralph|loves loves|Ellen Ellen|E
S|John                                 John|loves loves|Mary   Mary|E
```

Note that in contrast to the case in which individual words were aligned, we now have an unambiguously preferred alignment. S—Bert should align with S—John and Ralph—loves should align with John—loves. Aligning the gaps in this way then generates the following alignment in the ovbious way.

```
|Bert who knows Ralph|loves|Ellen|
|-------John---------|loves|Ellen|
```

Employing gap alignment requires that one construct a model of bigram substitutability. To calculate bigram change probabilitites a version of Hofmann’s (2001) aspect model of dyadic data was trained using a Markov Chain Monte Carlo method.

Evaluating the modified model

The task of unsupervised parsing provides a useful testbed with which to evaluate the ability of the modified model to identify constituents in sentences. Figure 3 shows a parse of the sentence ”Bert who knows Ralph loves Ellen” which specifies four constituents:

- Bert who knows Ralph loves Ellen
- Bert who knows Ralph
- knows Ralph
- loves Ellen

The task of the model, then, is to identify these constituents and conversely to avoid identifying non constituent spans such as ”knows Ralph loves”.

So that the model could be compared against other unsupervised parsing methods (e.g. Klein & Manning, 2001) a binary parse was constructed by applying the modified model to exemplars taken from the Wall Street Journal section of the

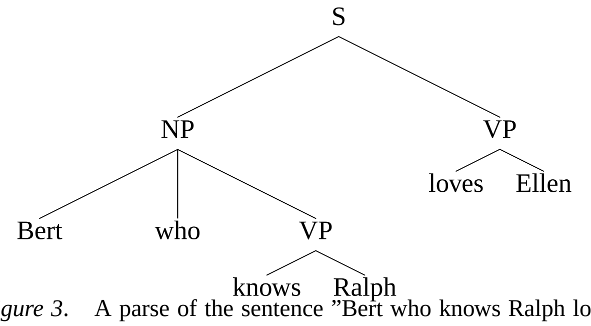


Figure 3. A parse of the sentence ”Bert who knows Ralph loves Ellen”.

Penn Treebank (Marcus et al., 1993). Only sentences that were shorter than the target sentence were chosen. The fifty most probable sentences for each length less than that of the target were used. The number of times each span of words was identified by the model as a constituent was determined and normalized to give probabilitites. For instance, the alignments presented in Figure 2 would give the following counts:

Span	Count	Probability
Bert who knows Ralph	5	.56
loves Ellen	3	.33
knows Ralph	1	.11

The most probable binary parse was then chosen using the obvious dynamic programming algorithm (c.f. Klein & Manning, 2001). This procedure was applied to all of the sentences from the Wall Street Journal section of the treebank that were of length 10 or less.

Results

To assess performance the parses produced by the model were compared against the gold standard parses provided by the treebank. Four 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.
- Non crossing boudnaries precision: The proporition of constituents proposed by the model that do not cross constituent boundaries in the gold standard.
- F_1: The harmonic mean of unlabelled recall and unlabelled precision.

Because the treebank provides parses that are not binary (in Chomsky normal form) but the procedure used makes this assumption it is not possible to achieve perfect performance. Klein and Manning (2001) calculated that best possible F_1 measure that can be achieved is 87%.

Figure 4 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 although all are still well below the theoretically achievable maximum of 87%.

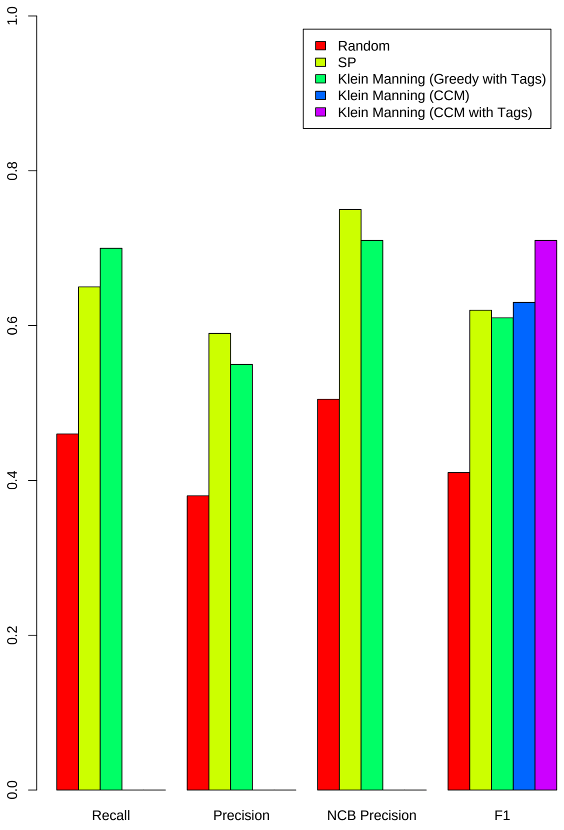


Figure 4. Results of Unsupervised Parsing Experiment.

Note that the ”greedy with tags” and ”CCM with tags” models both employed part of speech tags, which provides a boost to performance. The use of tag information means that these models are no longer strictly unsupervised. As the modified SP model was not provided with tag information the most appropriate comparison is against the straight CCM model. The performance of the modified SP model is approximately the same level as the CCM model with F_1 mesures around 62-63%. Note, however, that unlike the context constituent model the SP model provides insight into not only the problem of parsing, but also how propositional information can be extracted in an unsupervised manner.

General Conclusions

Latent Semantic Analysis (LSA) is a tool for analysizing text and a theory of how meaning is constructed. In addition, however, it exemplifies an approach to the modelling of cognitiion. In this approach, simple statistical operations are applied to large naturally occuring corpora. Structure is extracted from the environment rather than being specified a priori. In this chapter, I have atempted to illustrate using the Syntagmatic Paradigmatic model how this same approach can be applied to extracting propositional and syntactic information; domains that standard LSA cannot address because of its insensitivity to word order.

The ability of the model to answer questions about tennis matches that require a propositional analysis show that the model is able to extract this information. In addition, the model's reliance on ”inference by coincidence” in many cases suggests that it may prove more robust than existing inferential systems. Furthermore, a modifed version of the model can be used to parse naturally occurring sentences at levels that are close to the state of the art for unsupevised parsers. In both cases there remains significant room for improvement. Nonetheless, sucess on these tasks demonstrates that it is possible to extract both propositonal and syntactic information from corpora within the general constraints imposed by the LSA approach.

References

Allison, L., Wallace, C. S., & Yee, C. N. (1992). Finite-state models in the alignment of macromolecules. *Journal of Molecular Evolution*, *35*(1), 77-89.

Blaheta, D., & Charniak, E. (2000). Assigning function tags to parsed text. In *Proceedings of the 1st annual meeting the north american chapter of the acl (naacl)* (p. 234-240). Seattle, Washington.

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*.

Gildea, D., & Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, *28*(3), 245-288.

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

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, *42*(1-2), 177-196.

Kashket, M. (1986). Parsing a free-word order language: warlpiri. In *Proceedings of the 24th conference on association for computational linguistics.* Association for Computational Linguistics.

Kintsch, W. (2001). Predication. *Cognitive Science*, *25*, 173-202.

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

Landauer, T. K. (2002). On the computational basis of learning and cognition: Arguments from lsa. In N. Ross (Ed.), *The psychology of learning and motivation* (Vol. 41, p. 43 - 84). Academic Press.

Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E. (1997). How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In P. Langley (Ed.), *Proceedings of the 19th annual meeting of the cognitive science society* (p. 412-417). Mahwah, NJ: Erlbaum.

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions and reversals. *Dokl. Akad. Nauk. SSSR*, *163*, 845-848.

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.

Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, *48*, 443-453.

Palmer, M., Rosenzweig, J., & Cotton, S. (2001). Automatic predicate argument analysis of the penn treebank. In J. Allan (Ed.), *Proceedings of hlt 2001, first international conference on human language technology research.* San Francisco: Morgan Kaufmann.

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

Sankoff, D., & Kruskal, J. B. (1983). *Time warps, string edits and macromolecules: the theory and practise of sequence comparison*. Addison Wesley.

Sellers, P. H. (1974). An algorithm for the distance between two finite sequences. *Journal of Combinatorial Theory*, *16*, 253-258.