Lexical Attraction Models of Language

Deniz Yuret
MIT Artificial Intelligence Laboratory
545 Technology Square, NE43-815
Cambridge, MA 02139
deniz@ai.mit.edu

Abstract

Abstract ID: A229
This paper presents lexical attraction models of language, in which the only explicitly represented linguistic knowledge is the likelihood of pairwise relations between words. This is in contrast with models that represent linguistic knowledge in terms of a lexicon, which assigns categories to each word, and a grammar, which expresses possible combinations in terms of these categories. The word-based nature and the simplicity of lexical attraction models make them good candidates for experiments in language learning. I introduce an unsupervised learning algorithm that uses lexical attraction and gives accuracy results comparable to supervised learning.

Content Areas: Natural Language Processing # Techniques or Algorithms # statistical or corpus based methods, Machine Learning and Discovery # Tasks or Problems # unsupervised learning

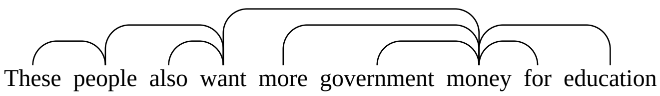
Introduction

The information in a sentence is contained partly in its words and partly in the relationships between the words. The main task of natural language processing is to identify the relationships between the words. When deciding whether two words are related we typically use two types of information. First, there are grammatical constraints, e.g. each determiner must modify a noun or transitive verbs must take objects. However, grammatical constraints are typically not restrictive enough to uniquely identify the correct relations. Second, there are selectional restrictions, e.g. given the verb *eat*, *cake* would be a more likely object than *train*. Lexical attraction models encode selectional restrictions. Ordinarily, one would need both types of information to identify the correct structure of a sentence. I will show how far we can go using lexical attraction alone.

The next section presents examples that demonstrate when grammatical constraints are not sufficient and lexical attraction is. The third section formalizes the model and expresses the meaning of my opening sentence in the language of information theory. The next section gives some basic results on dependency structures. The section on unsupervised learning gives the learning algorithm and evaluation results. The paper ends with a discussion of related work and future prospects.

Why lexical attraction?

Lexical attraction by itself can be sufficient to identify the correct relationships between the words in a sentence. The following example was taken from a training run after the unsupervised learning algorithm had processed a million words of news material:

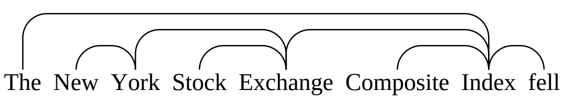


The links connect the word pairs that the program deemed most likely to be related (with certain constraints on the linkage to be explained later). Almost all the links correspond to what linguists would call head-modifier relationships for this sentence.

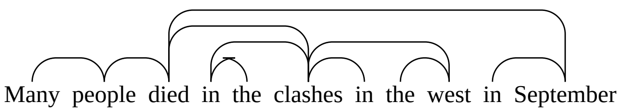
However, this perfect behavior is not typical. Experiments with the Penn Treebank corpus show that the best possible lexical attraction model can identify 77.4% of the links correctly.

Nevertheless, lexical attraction models can perform well in situations where grammatical constraints do not help. The following examples were taken at various stages of unsupervised learning with ten million words of news material.

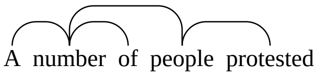
- Internal structure of a complex noun phrase:

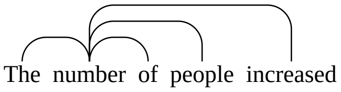


- Prepositional phrase attachment:

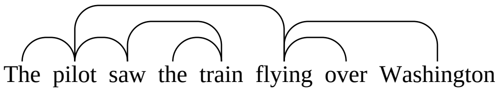


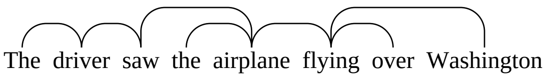
- Syntactic ambiguity:





- Another syntactic ambiguity:





The model

Lexical attraction is best described within the framework of information theory. Shannon defines the entropy of a discrete random variable as $H = -\sum p_i \log p_i$ where i ranges over the possible values of the random variable and p_i is the probability of value i (Shannon 1948; Cover & Thomas 1991). Consider a sequence of tokens drawn independently from a discrete distribution. In order to construct the shortest description of this sequence, each token i must be encoded using $-\log_2 p_i$ bits on average. $-\log_2 p_i$ can be defined as the information content of token i. Entropy can then be interpreted as the average information per token. Following is an English sentence with the information content of each word given below, assuming words are independently selected. The word probabilities were estimated using a large corpus of news material. Note that the information content is lower for the more frequently occurring words.

The	IRA	is	fighting	British	rule	in	Northern	Ireland
4.20	15.85	7.33	13.27	12.38	13.20	5.80	12.60	14.65

Maximum likelihood

The total information in this sentence is the sum of the information of each word, which is 99.28 bits. This is mathematically equivalent to the statement that the probability of seeing this sentence is the product of the probabilities of seeing each word, which is $2^{-99.28}$. Therefore there is an equivalence between the entropy and the probability assigned to the input. A probabilistic language model assigns a probability distribution over all possible sentences of the language. The maximum likelihood principle states that the parameters of a model should be estimated so as to maximize the probability assigned to the observed data. Therefore the most likely language model is also the one that achieves the lowest entropy.

Mutual information

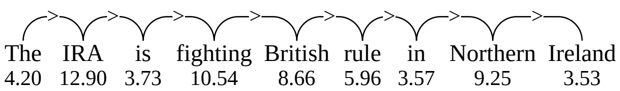
Language models achieve lower entropy by taking into account the relations between the words in the sentence. Consider the phrase *Northern Ireland*. Even though the independent probability of *Northern* is $2^{-12.6}$, it is seen before *Ireland* 36% of the time. Another way of saying this is that although *Northern* carries 12.6 bits of information by itself, it adds only 1.48 bits of new information to *Ireland*.

With this dependency, *Northern* and *Ireland* can be encoded using $1.48 + 14.65 = 16.13$ bits instead of $12.60 + 14.65 = 27.25$ bits. The 11.12 bit gain from the correlation of these two words is called mutual information. We measure lexical attraction with mutual information. The basic assumption of this work is that words with high lexical attraction are likely to be syntactically related.

Conditional independence

The *Northern Ireland* example shows that the information content of a word depends on other related words, i.e. its context. The context of a word in turn is determined by the language model used. The choice of context by a language model implies certain conditional independence assumptions, i.e. given the context of a word, its distribution is independent of the rest of the text.

For example, an n-gram model defines the context of a word as the $n-1$ words immediately preceding it. The following diagram gives the information content of the words in our example according to a bigram model. The information content of each word is computed based on its conditional probability given the previous word. As a result, the encoding of the sentence is reduced from 99.28 bits to 62.34 bits.



Every model has to make certain independence assumptions, otherwise the number of parameters would be prohibitively large to learn. Unfortunately, two words in a sentence are almost never completely independent. In fact, Beeferman et al. report that words can continue to show selectional influence for a window of several hundred words (Beeferman, Berger, & Lafferty 1997). However the degree of the dependency falls exponentially with distance. That justifies the choice of the n-gram models to relate dependency to proximity.

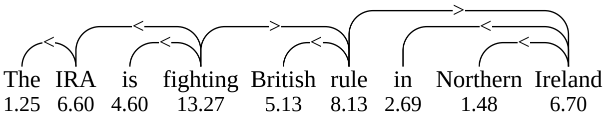
Linguistic context

Using the previous $n-1$ words as context is against our linguistic intuition. In a sentence like *"The man with the dog spoke"*, the selection of *spoke* is influenced by *man* and is independent of the previous word *dog*. It follows that the context of a word would be better determined by its linguistic relations rather than according to a fixed pattern.

The assumption in lexical attraction models is that each word depends on one other word in the sentence, but not necessarily an adjacent word as in n-gram models. Lexical attraction models make it possible to define the context of the word in terms of its syntactic relations.

Words in direct syntactic relation have strong dependencies. Chomsky defines such dependencies as *selectional relations* (Chomsky 1965). Subject and verb, for example, have a selectional relation, and so do verb and object. Subject and object, on the other hand, are assumed to be chosen independently of one another. It should be noted that this independence is only an approximation. The sentences *"The doctor examined the patient"* and *"The lawyer examined the witness"* show that the subject can have a strong influence on the choice of the object.

The following diagram gives the information content of the words in the example sentence based on direct syntactic relations:



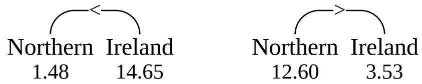
The arrows represent the head-modifier relations between words. The information content of each word is computed based on its conditional probability given its head. I marked the verb as governing the auxiliary and the noun governing the preposition which may look controversial to linguists. From an information theory perspective, the mutual information between content words is higher than that of function words. Therefore the model does not favor function word heads.

The probabilities were estimated by counting the occurrences of each pair in the same relative position. The linguistic dependencies reduce the encoding of the words in this sentence to 49.85 bits compared to the 62.34 bits of the bigram model. This number excludes the encoding of the dependency structure.

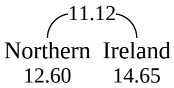
Symmetry of lexical attraction

Lexical attraction between two words is symmetric. The mutual information is the same no matter which direction the dependency goes. This directly follows from Bayes' rule. What is less obvious is that the choice of the head word and the corresponding dependency directions it imposes do not effect the joint probability of the sentence. The joint probability is determined only by the choice of the pairs of words to be linked.

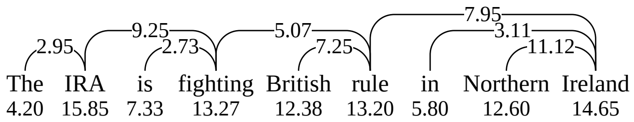
Consider the *Northern Ireland* example:



In the first case, I used the conditional probability of *Northern* given that the next word is *Ireland*. In the second case, I used the conditional probability of *Ireland* given that the previous word is *Northern*. In both cases the encoding of the two words is 16.13 bits, which is in fact $-\log_2 p$ of the joint probability of *Northern Ireland*. Thus a more natural representation would be the following, where the link has no direction and its label shows the number of bits gained, mutual information:



I generalize this result below and use the same representation for the whole sentence:



Theorem: The probability of a sentence with a given dependency structure does not depend on the choice of the head word.

Proof: Consider a sentence S where:

$$\begin{aligned} W &= \{w_0, w_1, \ldots, w_n\} \\ L &= \{(w_{i1}, w_{j1}), (w_{i2}, w_{j2}), \ldots\} \end{aligned}$$

denote words and links respectively. Let $P(L)$ denote the probability of a sentence having the dependency structure given by L. Assume that w_0 is the head word and every word probabilistically depends only on its governor. Then the joint probability of the sentence is given by the following expression:

$$\begin{aligned} P(S) &= P(L)P(w_0) \prod_{(w_i,w_j)\in L} P(w_j \mid w_i) \\ &= P(L)P(w_0) \prod_{(w_i,w_j)\in L} \frac{P(w_i,w_j)}{P(w_i)} \\ &= P(L) \prod_{w_i \in W} P(w_i) \prod_{(w_i,w_j)\in L} \frac{P(w_i,w_j)}{P(w_i)P(w_j)} \end{aligned}$$

In the final expression $P(w_0)$ plays no special role, i.e. starting from any other head word, I would have arrived at the same result. Therefore the choice of the head and the corresponding directions imposed on the links are immaterial for the probability of the sentence.

Therefore dependency structures can be formalized as Markov networks. A Markov network is an undirected graph representing the joint probability distribution for a set of variables (as opposed to Bayesian networks which are directed) (Pearl 1988). Each vertex corresponds to a random variable, a word in our case. The structure of the graph represents a set of conditional independence properties of the distribution: each variable is probabilistically independent of its non-neighbors in the graph given the state of its neighbors.

Information in a sentence

I started the paper saying that the information in a sentence is contained partly in its words and partly in the relationships between the words. The above proof contains the mathematical expression of this statement:

$$P(L) \prod_{w_i \in W} P(w_i) \prod_{(w_i, w_j) \in L} \frac{P(w_i, w_j)}{P(w_i) P(w_j)}$$

In the language of information theory, this is equivalent to saying:

information in a sentence =
information in the dependency structure
+ information in the words
– mutual information captured in syntactic relations.

The average information of an isolated word is independent of the language model. In a lexical attraction model the probability of possible dependency structures are equal. Therefore the first two terms in the expression have a constant contribution and the entropy of the model is completely determined by the mutual information captured in syntactic relations.

Non-lexical language models, such as probabilistic context free grammars, fail to represent this aspect of language. The non-terminal rules in a PCFG of the form $A \rightarrow BC$ represent the probability distribution over labeled trees, corresponding to $P(L)$ in the above expression. The terminal rules of the form $A \rightarrow x$ represent individual word probabilities within a single category, approximately corresponding to $\prod P(w_i)$. There is nothing in the PCFG model that represents information gained from relating words to each other. To use an ordinary PCFG for disambiguation is equivalent to saying that the correct parse of a sentence is the one that has the most likely compatible tree, more or less independent of the words in the sentence. Lexicalized PCFG's patch this weakness by pairing non-terminals with head words. Lexical attraction models do away with the tree probabilities completely and choose a parse based on the likelihood of individual words being related.

Dependency structure

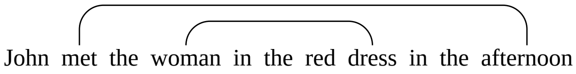
The linguistic formalism that takes syntactic relations between words as basic primitives is known as the dependency formalism. Mel'čuk discusses important properties of syntactic relations in his book on dependency syntax (Mel'čuk 1988). Sleator and Temperley have a large scale implementation of English syntax based on a similar formalism they call link grammars (Sleator & Temperley 1991). This section presents basic properties of linguistic dependency structures.

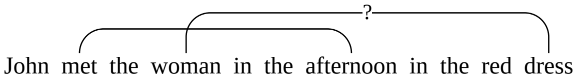
Basic properties

The probabilistic analysis given in the previous section assumes that the dependency structure is acyclic. It is generally the case that the syntactic relations in a sentence form an acyclic graph, i.e. a tree. Linguistically, each word in a sentence has a unique governor, except for the head word, which governs the whole sentence[1].

[1] See (Mel'čuk 1988, p. 25) for a discussion.

Most sentences in natural languages also have the property that syntactic relation links drawn over words do not cross. This property is called *planarity* (Sleator & Temperley 1993), *projectivity* (Mel'čuk 1988), or *adjacency* (Hudson 1984) by various researchers. The examples below illustrate the planarity of English. In the first sentence, it is easily seen that the woman was in the red dress and the meeting was in the afternoon. However, in the second sentence, the same interpretation is not possible. In fact, it seems more plausible for John to be in the red dress.





Gaifman gave the first formal analysis of dependency structures that satisfy the planarity condition(Gaifman 1965). His paper gives a natural correspondence between dependency systems and phrase-structure systems and shows that the dependency model characterized by planarity is context-free. Sleator and Temperley show that their planar model is also context-free even though it allows cycles (Sleator & Temperley 1991).

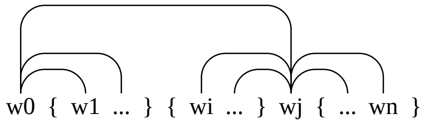
Linear Encoding

Lexical attraction models assume a uniform probability distribution over the possible dependency structures for a sentence. Thus, it is important to know the number of possible dependency structures to estimate the cost of their encoding. Without the planarity condition, the number of possible dependency structures for an n word sentence would be given by Cayley's formula: n^{n-2} (Harary 1969). The encoding of the dependency structure would then take $O(n \log n)$ bits. However, the encoding of planar dependency structures is linear in the number of words as the following theorem shows.

Theorem: Let $f(n)$ be the number of possible dependency structures for an $n+1$ word sentence. We have:

$$f(n) = \frac{1}{2n+1} \binom{3n}{n}$$

Proof: Consider a sentence with $n+1$ words. The leftmost word w_0 must be connected to the rest of the sentence via one or more links. Even though the links are undirected, I will impose a direction taking w_0 as the head to make the argument simpler. Let w_j be the rightmost child of w_0. We can split the rest of the sentence into three groups: left descendants of w_j span w_i^{j-1}, right descendants of w_j span w_{j+1}^n, and the remaining descendants of w_0 span w_1^{i-1}. Each of these three groups can be empty. The notation w_i^j denotes the span of words $w_i \dots w_j$.

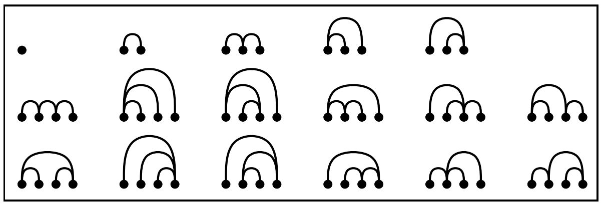


The problem of counting the number of dependency structures for the n words headed by w_0 can be split into three smaller versions of the same problem: count the number of structures for w_1^{i-1} headed by w_0, w_i^{j-1} headed by w_j, and w_{j+1}^n headed by w_j. Therefore $f(n)$ can be decomposed with the following recurrence relation:

$$f(n) = \sum_{p+q+r=n-1} f(p)f(q)f(r) \qquad p, q, r \geq 0$$

Here the numbers p, q, and r represent the number of words in w_1^{i-1}, w_i^{j-1}, and w_{j+1}^n respectively. This is a recurrence with 3-fold convolution. The general expression for a recurrence with m-fold convolution is $C(mn, n)/(mn - n + 1)$ where C is the binomial coefficient (Graham, Knuth, & Patashnik 1994). Therefore $f(n) = C(3n, n)/(2n + 1)$.

The first few values of $f(n)$ are: 1, 1, 3, 12, 55, 273, 1428. The following figure shows the possible dependency structures with up to four words.



An upper bound on the number of dependency structures can be obtained using the following inequality:

$$\binom{3n}{n} \leq \frac{(3n)^{3n}}{(n)^n(2n)^{2n}} = \frac{3^{3n}}{2^{2n}}$$

Taking the logarithm of this value and dividing it by n, we can show that the encoding of a planar dependency structure takes less than $3\log_2 3 - 2 \approx 2.75$ bits per word.

Unsupervised learning

This section presents an on-line unsupervised learning algorithm and the results of training lexical attraction models. The algorithm interdigitates learning and processing for every input sentence in the following manner:

- Find the most likely structure for an input sentence given the current state of the model.
- Update the model assuming the structure found is the correct one.

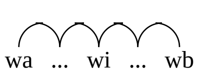
Parser

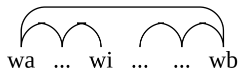
Given a lexical attraction model, the most likely dependency structure for a sentence can be found in time $O(n^3)$ using dynamic programming as first shown in (Eisner 1996). Here, I will present a simple derivation for a similar algorithm.

Let $P(a, b)$ be the probability of w_a^b given w_a and the most likely dependency structure. P can be defined recursively as follows:

$$\begin{aligned} P_1 &= \max_i P(a, i)P(i, b) \\ P_2 &= \max_i P(a, i)P(w_b|w_a)P(b, i + 1) \\ P(a, b) &= \max(P_1, P_2) \end{aligned}$$

The computations of P_1 and P_2 enumerate the uncovered and covered structures respectively:





The above recursion implemented as a memoizing procedure directly gives an $O(n^3)$ algorithm. Each recursive call can record the best i found for its argument for recovery of the dependency structure later.

Experiments

In the following experiments, test results were obtained using one million words of annotated Wall Street Journal text from the Penn Treebank corpus (Marcus & others 1994). The phrase structure annotations were converted into dependency structure links and the results give the percentage of links guessed correctly in the experiment. For unsupervised training, out of sample plain WSJ text was used. Experiments 1–3 were run for benchmarking purposes.

Experiment 1 In this experiment, the sentences were parsed using a random model. The model used a random number generator to generate the lexical attraction values. The resulting accuracy was 25.2%.

Experiment 2 The annotated test data was used for supervised training. The resulting model was tested on the same data to get an upper bound on the performance of a lexical attraction model which resulted in 77.4% accuracy.

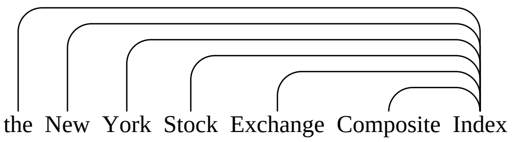
Experiment 3 One thousand sentences were held out from the test data and the remaining was used for supervised training. Testing on the held out data led to 53.4% accuracy.

Experiment 4 In this experiment, I started with an empty model, giving 0 frequency for any word pair. I used the same 20 million words of training data as the previous experiment. The parser was run on every sentence using the existing model. The model was updated after every sentence by incrementing the frequencies of the linked pairs. The resulting accuracy was 35.5%.

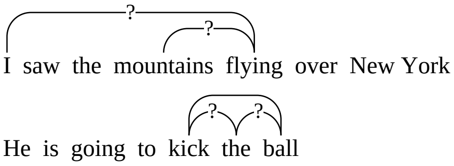
Experiment 5 Starting with an empty model and using the same 20 million words of training data, the parser was run on every sentence using the existing model as in the previous experiment. However, the model was updated after every sentence by incrementing the frequencies of all first and second degree neighbors. The resulting accuracy was 40.4%.

The difference between experiments 4 and 5 is significant. When the model is empty, giving 0 frequency to each pair, the parser ends up linking adjacent words. In experiment 4, positive mutual information between some linked pairs is discovered and the parser starts linking them even when they are not adjacent. However, related words that are never seen adjacent cannot be discovered. These include objects of verbs separated by determiners, as in "**kick** the **ball**", or prepositional phrases and adverbials modifying a verb that need to follow the object, such as "**kick** the ball **today**". Incrementing the second degree neighbors in experiment 5 solves this problem and lets the program discover all related words eventually.

The use of Penn Treebank data for evaluation of unsupervised language acquisition is debatable. The conversion from the phrase structure representation to dependency structure representation is bound to have errors. These errors are most pronounced in noun phrases, where the Penn Treebank gives no internal structure. The heuristics employed lead to the following linkage which results in an accuracy result of 50% for the correct linkage discovered by the program:



Another problem with the evaluation is the equal treatment of content and function words. For extraction of meaning, the mistakes in content-word links are significantly more important than the mistakes in function-word links. The following two sentences illustrate the difference. In the first sentence, a mistake would result in choosing the wrong subject for flying. In the second sentence, once the program has detected the relation between kick and ball, which way the word *the* links is less important.



The program used in experiment 5 discovers 45.6% of the content-word links correctly. With 180 million words of training, this accuracy goes up to 51.9%.

Discussion

Lexical attraction models make the following two simplifying assumptions about language:

- Each word is independent of the rest of the words in the text given its head.
- The possible dependency structures for a sentence are apriori equally likely.

The first assumption is violated when there are second order relations. For example in prepositional phrases the preposition and the noun determine the distribution of the phrase together. The second assumption is against the arity requirements of words. When a determiner links to a noun, in general it should not link to another noun no matter how strong the lexical attraction. Relaxing these assumptions while keeping the mathematical and linguistic sensibility of the theory may lead to powerful models.

(Eisner 1996) presents three possible models and achieves good parsing performance using supervised training. (Chelba & Jelinek 1998) give the best known results in language modeling for speech recognition using a similar framework.

Summary

I presented some basic results for lexical attraction models, in which the only explicity represented linguistic knowledge is the likelihood of pairwise relations between words. I showed that these models can be used for unsupervised language learning. With an accuracy of around 50% for relations between content-words, the trained models can be used in information retrieval and information extraction applications.

References

Beeferman, D.; Berger, A.; and Lafferty, J. 1997. A model of lexical attraction and repulsion. In *ACL/EACL '97*.

Chelba, C., and Jelinek, F. 1998. Exploiting syntactic structure for language modeling. In *COLING-ACL 98*.

Chomsky, N. 1965. *Aspects of the theory of syntax*. MIT Press.

Cover, T. M., and Thomas, J. A. 1991. *Elements of Information Theory*. John Wiley and Sons, Inc.

Eisner, J. M. 1996. Three new probabilistic models for dependency parsing. In *COLING-96*.

Gaifman, H. 1965. Dependency systems and phrase-structure systems. *Information and Control* 8:304–337.

Graham, R. L.; Knuth, D. E.; and Patashnik, O. 1994. *Concrete Mathematics*. Addison-Wesley, 2 edition.

Harary, F. 1969. *Graph Theory*. Addison-Wesley.

Hudson, R. A. 1984. *Word Grammar*. B. Blackwell.

Marcus, M. P., et al. 1994. The penn treebank: annotating predicate argument structure. In *ARPA Human Language Technology Workshop*.

Mel'čuk, I. A. 1988. *Dependency Syntax: Theory and Practice*. SUNY.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Shannon, C. E. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27.

Sleator, D., and Temperley, D. 1991. Parsing english with a link grammar. Technical Report CMU-CS-91-196, CMU.

Sleator, D., and Temperley, D. 1993. Parsing english with a link grammar. In *Third international workshop on parsing technologies*.