On Statistical Methods in Natural Language Processing

Joakim NIVRE
School of Mathematics and Systems Engineering,
Växjö University, SE-351 95 Växjö, Sweden

Abstract What is a statistical method and how can it be used in natural language processing (NLP)? In this paper, we start from a definition of NLP as concerned with the design and implementation of effective natural language input and output components for computational systems. We distinguish three kinds of methods that are relevant to this enterprise: application methods, acquisition methods, and evaluation methods. Using examples from the current literature, we show that all three kinds of methods may be statistical in the sense that they involve the notion of probability or other concepts from statistical theory. Furthermore, we show that these statistical methods are often combined with traditional linguistic rules and representations. In view of these facts, we argue that the apparent dichotomy between "rule-based" and "statistical" methods is an over-simplification at best.

1 Introduction

In the current literature on natural language processing (NLP), a distinction is often made between "rule-based" and "statistical" methods for NLP. However, it is seldom made clear what the terms "rule-based" and "statistical" really refer to in this connection. Is it the knowledge of language embodied in the respective methods? Is it the way this knowledge is acquired? Or is it the way the knowledge is applied?

In this paper, we will try to throw some light on these issues by examining the different ways in which NLP methods deserve to be called "statistical", an exercise that will hopefully throw some light also on methods that do not deserve to be so called. We hope to show that statistics can play a role in all the major categories of NLP methods, that many of the "rule-based methods" actually involve statistics, and that many of the "statistical methods" employ quite traditional linguistic rules. We will therefore conclude that a more fruitful discussion of the methodology of natural language processing requires a more articulated conceptual framework, to which the present paper can be seen as a contribution.

2 NLP: Problems, Models and Methods

According to the recently published *Handbook of Natural Language Processing* [17, p. v], NLP is concerned with "the design and implementation of effective natural language input and output components for computational systems". The most important problems in NLP therefore have to do with natural language input and output. Here are a few typical and uncontroversial examples of such problems:

- Part-of-speech tagging: Annotating natural language sentences or texts with parts-of-speech.
- Natural language generation: Producing natural language sentences or texts from non-linguistic representations.
- Machine translation: Translating sentences or texts in a source language to sentences or texts in a target language.

In part-of-speech tagging we have natural language input, in generation we have natural language output, and in translation we have both input and output in natural language.

If our aim is to build effective components for computational systems, then we must develop *algorithms* for solving these problems. However, this is not always possible, simply because the problems are not well-defined enough. The way out of this dilemma is the same as in most other branches of science. Instead of attacking real world problems directly with all their messy details, we build mathematical models of reality and solve abstract problems within the models instead. Provided that the models are worth their salt, these solutions will provide adequate approximations for the real problems.

Formally, an abstract problem Q is a binary relation on a set I of problem *instances* and a set S of problem *solutions* [14]. The abstract problems that are relevant to NLP are those where either I or S (or both) are linguistic entities or representations of linguistic entities. More precisely, an NLP problem P can be modeled by an abstract problem Q if the instance set I is a subset of the set of permissible inputs to P and the solution set S is a subset of the set of possible solutions to P.[1]

2.1 Application Methods

A method for solving an NLP problem P typically consists of two elements:

1. A mathematical model M defining an abstract problem Q that can be used to model P.
2. An algorithm A that effectively computes Q.

We will say that M and A together constitutes an *application method* for problem P with Q as the *model problem*. For example, let G be a context-free grammar intended to model the syntax of a natural language L and let Q be the parsing problem for G. Then G together with, say, Earley's algorithm is an application method for syntactic analysis of L with Q as the model problem. In general, the relation between real problems, abstract problems, models and algorithms can be depicted as in Figure 1.[2]

For most application methods, the mathematical model M can be defined independently of the algorithm A. For example, a context-free grammar used in syntactic analysis is not dependent on any particular parsing algorithm, and there are many different parsing algorithms that can be used besides Earley's algorithm. Moreover, one and the same model can be used with different algorithms to compute different abstract problems, thus constituting application methods for different NLP problems. A case in point is a bidirectional grammar, which can be used with different algorithms to perform either parsing or generation (see, e.g., [1]). Other examples will be discussed below.

[1] In fact, it is sufficient that there exist effectively computable mappings from P inputs to I and from S to P solutions.

[2] Thanks to Mark Dougherty for designing this diagram.

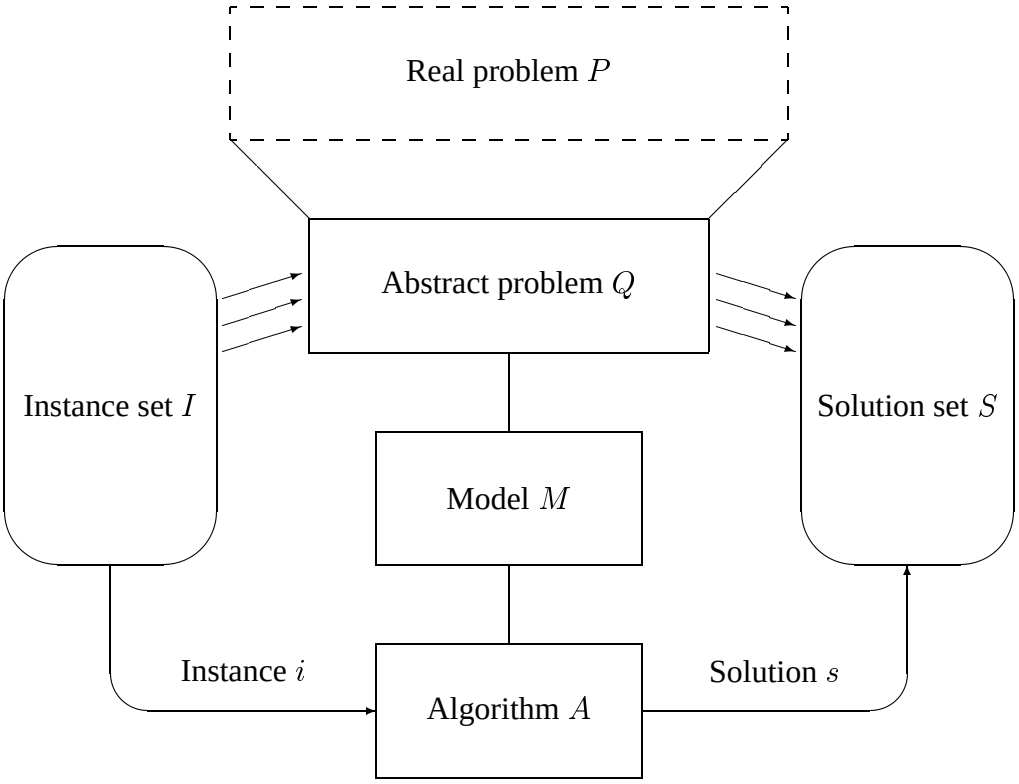

Figure 1: Real problems, abstract problems, models and algorithms

Example 1: Hidden Markov Models Let $M(S, K, \Pi, A, B)$ be a hidden Markov model with state set S, output alphabet K and probability distributions Π (initial state), A (state transitions) and B (symbol emissions) (see, e.g., [23]). Let Q_1 be the abstract problem of determining the optimal state sequence $X_1, X_2, \ldots X_T$ for a given observation sequence O of length T, and let Q_2 be the abstract problem of determining the probability of a given observation sequence O. The problem Q_1 can be computed in linear time using the Viterbi algorithm [31]. The problem Q_2 can be computed in linear time using one of several algorithms usually called the forward procedure, the backward procedure, and the forward-backward procedure [23].

If the states in S correspond to lexical categories, or parts-of-speech, and the symbols in K corresponds to word forms in a natural language L, the model M together with the Viterbi algorithm constitutes an application method for the part-of-speech tagging with Q_1 as the model problem. This is the standard method used in statistical part-of-speech tagging (see, e.g., [12, 15]). At the same time, however, the model M can be used together with the forward procedure to solve the language modeling problem in an automatic speech recognition system for L, with Q_2 as the model problem [9]. □

2.2 Acquisition Methods

So far, we have been concerned with methods for computing NLP problems, consisting of mathematical models with appropriate algorithms. However, these are not the only methods that are relevant within the field of NLP. We will use the term *acquisition method* to

refer to any procedure for constructing a mathematical model that can be used in an application method. For example, any procedure for developing a context-free grammar modeling a natural language or a hidden Markov model for part-of-speech tagging is an acquisition method in this sense. Compared to application methods, these methods form a rather heterogeneous class, ranging from rigorous algorithmic methods to the more informal problem-solving strategies typically employed by human beings.

In the following, we will concentrate almost exclusively on acquisition methods that make use of machine learning techniques in order to induce models (or model parameters) from empirical data, specifically *corpus* data. An empirical and algorithmic acquisition method typically consists of two elements:

1. A parameterized mathematical model M_Θ such that providing values for the parameters Θ will yield a mathematical model M that can be used in an application method for some NLP problem P.

2. An algorithm A that effectively computes values for the parameters Θ when given a sample of data from P.

If the data sample must contain both inputs and (correct) outputs from P, then A is said to be a *supervised* learning algorithm. If it is sufficient with a sample of inputs, we have an *unsupervised* learning algorithm.

Example 2: Hidden Markov Models (cont'd) Let $M_\Theta(S, K, \theta_\Pi, \theta_A, \theta_B)$ be a parameterized hidden Markov model with state set S and output alphabet K, but where probability distributions are unspecified. The acquisition problem in this case consists in finding suitable values for the distribution parameters θ_Π, θ_A and θ_B.

The Baum-Welch algorithm [4], sometimes called the forward-backward algorithm, is an unsupervised learning algorithm for solving this problem, given a sample of observation sequences with symbols drawn from K.

Thus, given a corpus C of texts in a natural language L such that the set of words occurring in C is (a subset of) K and S is a suitable tagset for L, then M_Θ together with the Baum-Welch algorithm constitutes an acquisition method for HMM-based part-of-speech tagging of L. □

2.3 Evaluation Methods

If acquisition and application methods were infallible, no other methods would be needed. In practice, however, we know that there are many factors which may cause an NLP system to perform less than optimally. For example, consider a situation where we first apply an acquisition method (M_Θ, A_1) to some corpus data C to construct a model M, and then use an application method (M, A_2) to solve an NLP problem P with the model problem Q. Then the following are some of the reasons why the performance on problem P may be suboptimal:

- The algorithm A_1 may fail to produce the best model given M_Θ and C.
- The algorithm A_2 may fail to compute the abstract problem Q.
- The abstract problem Q may be an inadequate model of P.

In this paper, we will use the term *evaluation method* to refer to any procedure for evaluating NLP systems. However, the discussion will focus on extrinsic evaluation of systems in terms of their accuracy. For example, let P be an NLP problem, and let (M_1, A_1) and (M_2, A_2) be two different application methods for P. A common way of evaluating and comparing the accuracy of these two methods is to apply them to a representative sample of inputs from P and measure the accuracy of the outputs produced by the respective methods. A special case of this evaluation scheme is the case where $A_1 = A_2$ and the models M_1 and M_2 are the results of applying two different acquisition methods to the same parameterized model M_Θ and training corpus C. In this case, it is primarily the acquisition methods that are evaluated. Moreover, the fact that this kind of evaluation is often integrated as a feedback loop into the actual acquisition method means that in practice the relationship between application methods, acquisition methods and evaluation methods can be quite complex. Still, from an analytical point of view, the three classes of methods are clearly distinguishable.

Example 3: Parsing Accuracy Let C be a corpus of parse trees for sentences in some natural language L, labeled with a set of category symbols V, and let S be a deterministic parsing system for L using the same set of category symbols. Using C as an empirical gold standard, we can evaluate the accuracy of S by running S on (the yields of trees in) C and comparing, for every sentence s in C, the parse tree $S(s)$ produced by S with the (presumably correct) parse tree $C(s)$ in C. We say that a constituent of a parse tree $S(s)$ is *correct* if the same constituent (with the same label) is found in $C(c)$. Two commonly used evaluation metrics are the following (see, e.g., [22]):

- Labeled recall:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\text{\# of correct constituents in } S(s_i)}{\text{\# of constituents in } C(s_i)}$$

- Labeled precision:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{\text{\# of correct constituents in } S(s_i)}{\text{\# of constituents in } S(s_i)}$$

When using these measures to compare the relative accuracy of several systems, we use standard techniques for assessing the statistical significance of any detected differences. □

3 Statistical Models and Methods

Having discussed in some detail what we mean by models and methods in NLP, we may now consider the question of what it means for a model or method to be *statistical*. According to [19], there are two broad classes of mathematical models: deterministic and stochastic. A mathematical model is said to be *deterministic* if it does not involve the concept of probability; otherwise it is said to be *stochastic*. Furthermore, a stochastic model is said to be *probabilistic* or *statistical*, if its representation is from the theories of probability or statistics, respectively.

Although Edmundson applies the terms *stochastic*, *probabilistic* and *statistical* only to *models*, it is obvious that they can be used about *methods* as well. First of all, we have defined

both application methods and acquisition methods in such a way that they crucially involve a (possibly parameterized) model. If this model is stochastic, then it is reasonable to call the whole method stochastic. Secondly, we shall see that also the algorithmic parts of application and acquisition methods can contain stochastic elements. Finally, it seems uncontroversial to apply the term *statistical* to evaluation methods that make use of descriptive and/or inferential statistics.

In the taxonomy proposed by Edmundson, the most general concept is that of a *stochastic* model, with probabilistic and statistical models as special cases. Although this may be the mathematically correct way of using these terms, it does not seem to reflect current usage in the NLP community, where especially the term *statistical* is used in a wider sense more or less synonymous with *stochastic* in Edmundson's sense. We will continue to follow current usage in this respect.

Thus, for the purpose of this paper, we will say that a model or method is *statistical* (or *stochastic*) if it involves the concept of probability (or related notions such as entropy and mutual information) or if it uses concepts of statistical theory (such as statistical estimation and hypothesis testing).

4 Statistical Methods in NLP

In the remainder of this paper, we will discuss different ways in which statistical (or stochastic) models and methods can be used in NLP, using concrete examples from the literature to illustrate our points.

4.1 Application Methods

Most examples of statistical application methods in the literature are methods that make use of a stochastic model, but where the algorithm applied to this model is entirely deterministic. Typically, the abstract model problem computed by the algorithm is an *optimization problem* which consists in maximizing the probability of the output given the input. Here are some examples:

- Language modeling for automatic speech recognition using smoothed n-grams to find the most probable string of words $w_1, \ldots, w_n$ out of a set of candidate strings compatible with the acoustic data [21, 2].
- Part-of-speech tagging using hidden Markov models to find the most probable tag sequence $t_1, \ldots t_n$ given a word sequence $w_1, \ldots w_n$ [12, 15, 24].
- Syntactic parsing using probabilistic grammars to find the most probable parse tree T given a word sequence $w_1, \ldots, w_n$ (or tag sequence $t_1, \ldots, t_n$) [5, 30, 11].
- Word sense disambiguation using Bayesian classifiers to find the most probable sense s for word w in context C [20, 32].
- Machine translation using probabilistic models to find the most probable target language sentence t for a given source language sentence s [8, 10].

Many of the application methods listed above involve models that can be seen as instances of Shannon's *noisy channel model* [29], which represents a Bayesian modeling approach. The essential components of this model are the following:

- The problem is to predict a hidden variable H from an observed variable O, where O can be seen as the result of transmitting H over a noisy channel.
- The solution is to find that value h of H which maximizes the conditional probability $P(h \mid o)$, for the observed value o of O.
- The conditional probability $P(h \mid o)$ is often difficult to estimate directly, because this requires control over the variable o whose value is probabilistically dependent on the noisy channel.
- Therefore, instead of maximizing $P(h \mid o)$, we maximize the product $P(h)P(o \mid h)$, where the factors can be estimated independently, given representative samples of H and (H, O), respectively.

Within the field of NLP, the noisy channel model was first applied with great success to the problem of speech recognition [21, 2]. As pointed out by [13], this inspired NLP researchers to apply the same basic model to a wide range of other NLP problems, where the original channel metaphor can sometimes be extremely far-fetched.

It should be noted that there is no conflict in principle between the use of stochastic models and the notion of linguistic *rules*. For example, probabilistic parsing often makes use of exactly the same kind of rules as traditional grammar-based parsing and produces exactly the same kind of parse trees. Thus, a stochastic context-free grammar is an ordinary context-free grammar, where each production rule is associated with a probability (in such a way that probabilities sum to 1 for all rules with the same left-hand side); cf. also [5, 30, 11].

All of the examples discussed so far involve a stochastic model in combination with a deterministic algorithm. However, there are also application methods where not only the model but also the algorithm is stochastic in nature. A good example is the use of a Monte Carlo algorithm for parsing with the DOP model [6]. This is motivated by the fact that the abstract model problem, in this case the parsing problem for the DOP model, is intractable in principle and can only be solved efficiently by approximation.

4.2 Acquisition Methods

Statistical acquisition methods are methods that rely on *statistical inference* to induce models (or model parameters) from empirical data, in particular corpus data, using either supervised or unsupervised learning algorithms (cf. section 2.2). The model induced may or may not be a stochastic model, which means that there are as many variations in this area as there are different NLP models. We will therefore limit ourselves to a few representative examples and observations, starting with acquisition methods for stochastic models.

Supervised learning of stochastic models is often based on maximum-likelihood estimation (MLE) using relative frequencies. Given a parameterized model M_Θ with parameter Θ and a sample of data C, a maximum likelihood estimation of Θ is an estimate that maximizes the *likelihood function* is $P(C \mid \Theta)$. For example, if we want to estimate the category probabilities of a discrete variable X with a finite number of possible values $x_1, \ldots, x_n$ given a

sample C, then the MLE is obtained by letting $\hat{P}(x_i) = f_C(x_i)(1 \leq i \leq n)$, where $f_C(x_i)$ is the relative frequency of x_i in C.

In actual practice, pure MLE is seldom satsifactory because of the so-called sparse data problem, which makes it necessary to *smoothe* the probability distributions obtained by MLE. For example, hidden Markov models for part-of-speech tagging are often based on smoothed relative frequency estimates derived from a tagged corpus (see, e.g., [24, 25]; cf. also section 2.2 above).

Unsupervised learning of stochastic models requires a method for estimating model parameters from unanalyzed data, such as the Expectation-Maximization algorithm [18]. Let M_Θ be a parameterized model with parameter Θ, let H be the hidden (analysis) variable, and let C be a data sample from the observable variable O. Then, as observed in [23], the EM algorithm can be seen as an iterative solution to the following circular statements:

- **Estimate:** If we knew the value of Θ, then we could compute the expected distribution of H in C.
- **Maximize:** If we knew the distribution of H in C, then we could compute the MLE of Θ.

The circularity is broken by starting with a guess for Θ and iterating back and forth between an *expectation step* and a *maximization step* until the process converges, which means that a local maximum for the likelihood function has been found. This general idea is instantiated in a number of different algorithms that provide acquisition methods for different stochastic models. Here are some examples, taken from [23]:

- The Baum-Welch or forward-backward algorithm for hidden Markov models [4].
- The inside-outside algorithm for inducing stochastic context-free grammars [3].
- The unsupervised word sense disambiguation algorithm of [28].

It is important to note that, although statistical acquisition methods may be more prominent in relation to stochastic models, they can in principle be used to induce any kind of model from empirical data, given suitable constraints on the model itself. In particular, statistical methods can be used to induce models involving linguistic rules of various kinds, such as rewrite rules for part-of-speech tagging [7] or constraint grammar rules [27].

Finally, we note that the use of stochastic or randomized algorithms can be found in acquisition methods as well as application methods. Thus, in [26] a Monte-Carlo algorithm is used to improve the efficiency of transformation-based learning [7] when applied to dialogue act tagging.

4.3 Evaluation Methods

As noted earlier, evaluation of NLP systems can have different purposes and consider many different dimensions of a system. Consequently, there are a wide variety of methods that can be used for evaluation. Many of these methods involve empirical experiments or quasi-experiments in which the system is applied to a representative sample of data in order to provide quantitative measures of aspects such as efficiency, accuracy and robustness. These evaluation methods can make use of statistics in at least three different ways:

- Descriptive statistics

- Estimation
- Hypothesis testing

Before exemplifying the use of descriptive statistics, estimation and hypothesis testing in natural language processing, it is worth pointing out that these methods can be applied to any kind of NLP system, regardless of whether the system itself makes use of statistical methods. It is also worth remembering that evaluation methods are used not only to evaluate complete systems but also to provide iterative feedback during acquisition (cf. section 2.3).

Descriptive statistics is often used to provide the quantitative measurements of a particular quality such as accuracy or robustness, as exemplified in the following list:

- Word error rate, usually defined as the number of deletions, insertions and substitutions divided by the number of words in the test sample, is the standard measure of accuracy for automatic speech recognition systems (see, e.g., [22]).
- Accuracy rate (or percent correct), defined as the number of correct cases divided by the total number of cases, is commonly used as a measure of accuracy for part-of-speech tagging and word sense disambiguation (see, e.g., [22]).
- Recall and precision, often defined as the number of true positives divided by, respectively, the sum of true positives and false negatives (recall) and the sum of true positives and false positives (precision), are used as measures of accuracy for a wide range of applications including part-of-speech tagging, syntactic parsing and information retrieval (see, e.g, [22]).

Statistical estimation becomes relevant when we want to generalize the experimental results obtained for a particular test sample. For example, suppose that a particular system s obtains accuracy rate r when applied to a particular test corpus. How much confidence should we place on r as an estimate of the true accuracy rate ρ of system s? According to statistical theory, the answer depends on a number of factors such as the amount of variation and the size of the test sample. The standard method for dealing with this problem is to compute a *confidence interval* i, which allows us to say that the real accuracy rate ρ lies in the interval $[r - i/2, r + i/2]$ with probability p. Commonly used values of p are 0.95 and 0.99.

Statistical hypothesis testing is crucial when we want to compare the experimental results of different systems applied to the same test sample. For example, suppose that two systems s_1 and s_2 obtain an error rate of r_1 and r_2 when measured with respect to a particular test corpus, and suppose furthermore that $r_1 < r_2$. Can we draw the conclusion that s_1 has higher accuracy than s_2 in general? Again, statistical theory tells us that the answer depends on a number of factors including the size of the difference $r_2 - r_1$, the amount of variation, and the size of the test sample. And again, there are standard tests available for testing whether a difference is statistically significant, i.e. whether the probability p that there is no difference between ρ_1 and ρ_2 is smaller than a particular threshold α. Standard tests of statistical significance for this kind of situation include the paired t-test, Wilcoxon's signed ranks test, and McNemar's test. Commonly used values of α are 0.05 and 0.01.

5 Conclusion

In this paper, we have discussed three different kinds of methods that are relevant in natural language processing:

- An *application method* is used to solve an NLP problem P, usually by applying an algorithm A to a mathematical model M in order to solve an abstract problem Q approximating P.
- An *acquisition method* for an NLP problem P is used to construct a model M that can be used in an application method for P. Of special interest here are empirical and algorithmic acquisition methods that allow us to construct M from a parameterized model M_Θ by applying an algorithm A to a representative sample of P.
- An *evaluation method* for an NLP problem P is used to evaluate application methods for P. Of special interest here are experimental (or empirical) evaluation methods that allow us to evaluate application methods by applying them to a representative sample of P.

We have argued that statistics, in the wide sense including both stochastic models and statistical theory, can play a role in all three kinds of methods and we have supplied numerous examples to substantiate this claim. We have also tried to show that there are many ways in which statistical methods can be combined with traditional linguistic rules and representation, both in application methods and in acquisition methods. In conclusion, we believe that methodological discussions in NLP can benefit from a more articulated conceptual framework and we hope that the ideas presented in this paper can make some contribution to such a framework.

References

[1] Appelt, D. E. (1987) Bidirectional Grammars and the Design of Natural Language Generation Systems. In Wilks, Y. (ed) *Theoretical Issues in Natural Language Processing 3*, pp. 185–191. Hillsdale, NJ: Lawrence Erlbaum.

[2] Bahl, L. R., Jelinek, F. and Mercer, R. L. (1983) A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5(2), 179–190.

[3] Trainable Grammars for Speech Recognition. In Klatt, D. H. and Wolf, J. J. (eds) *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pp. 547–550.

[4] Baum, L. E. and Petrie, T. (1966) Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Annals of Mathematical Statistics* 37, 1559–1563.

[5] Black, E., Jelinek, F., Lafferty, J. D., Magerman, D. M., Mercer, R. L. and Roukos, S. (1992) Towards History-Based Grammars: Using Richer Models for Probabilistic Parsing. In *Proceedings DARPA Speech and Natural Language Workshop*, Harriman, New York, pp. 134–139. Los Altos, CA: Morgan Kaufman.

[6] Bod, R. (1999) *Beyond Grammar: An Experience-Based Theory of Language*. Cambridge University Press.

[7] Brill, E. (1995) Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21(4), 543–566.

[8] Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Lafferty, J., Mercer, R. and Rossin, P. (1990) A Statistical Approach to Machine Translation. *Computational Linguistics* 16(2), 79–85.

[9] Brown, P. F., Della Pietra, V. J., deSouza, P. V. and Mercer, R. L. (1990) Class-Based N-Gram Models of Natural Language. In *Proceedings of the IBM Natural Language ITL*, pp. 283–298. Paris, France.

[10] Brown, P., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. (1993) The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19(2), 263–311.

[11] Charniak, E. (1997) Statistical Parsing with a Context-Free Grammar and Word Statistics. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI-97).* Menlo Park: AAAI Press.

[12] Church, K. (1988) A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. Second Conference on Applied Natural Language Processing, ACL.

[13] Church, K. W. and Mercer, R. L. (1993) Introduction to the Special Issue on Computational Linguistics Using Large Corpora. *Computational Linguistics* 19, 1–24.

[14] Cormen, T. H., Leiserson, C. E. and Rivest, R. L. (1990) *Introduction to Algorithms*. MIT Press.

[15] Cutting, D., Kupiec, J., Pedersen, J. and Sibun, P. (1992). A Practical Part-of-speech Tagger. In *Third Conference on Applied Natural Language Processing*, ACL, 133–140.

[16] Dale, R. (2000) Symbolic Approaches to Natural Language Processing. In [17], pp. 1–9.

[17] Dale, R., Moisl, H. and Somers, H. (eds.) (2000) *Handbook of Natural Language Processing*. Marcel Dekker.

[18] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39, 1–38.

[19] Edmundson, H. P. (1968) Mathematical Models in Linguistics and Language Processing. In Borko, H. (ed.) *Automated Language Processing*. John Wiley and Sons.

[20] Gale, W. A., Church, K. W. and Yarowsky, D. (1992) A Method for Disambiguating Word Senses in a Large Corpus. *Computers and the Humanities* 26, 415–439.

[21] Jelinek, F. (1976) Continuous Speech Recognition by Statistical Methods. *Proceedings of the IEEE* 64(4), 532–557.

[22] Jurafsky, D. and Martin, J. H. (2000) *Speech and Language Processing*. Upper Saddle River, NJ: Prentice-Hall.

[23] Manning, C. D. and Schütze, H. (1999) *Foundations of Statistical Natural Language Processing.* MIT Press.

[24] Merialdo, B. (1994) Tagging English Text with a Probabilistic Model. *Computational Linguistics* 20(2), 155–172.

[25] Nivre, J. (2000) Sparse Data and Smoothing in Statistical Part-of-Speech Tagging. *Journal of Quantitative Linguistics* 7(1), 1–18.

[26] Samuel, K., Carberry, S. and Vijay-Shanker, K. (1998) Dialogue Act Tagging with Transformation-Based Learning. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING-14)*, pp. 1150–1156.

[27] Samuelsson, C., Tapanainen, P. and Voutilainen, A. (1996) Inducing Constraint Grammars. In Miclet, L. and de la Higuera, C. (eds) *Grammatical Inference: Learning Syntax from Sentences*, Lecture Notes in Artificial Intelligence 1147, pp. 146–155. Springer.

[28] Schütze, H. (1998) Automatic Word Sense Discrimination. *Computational Linguistics* 24, 97–237.

[29] Shannon, C. E. (1948) A Mathematical Theory of Communication. *Bell System Technical Journal* 27, 379–423, 623–656.

[30] Stolcke, A. (1995) An Efficient Probabilistic Context-Free Parsing Algorithm That Computes Prefix Probabilities. *Computational Linguistics* 21(2), 165–202.

[31] Viterbi, A. J. (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory* 13, 260–269.

[32] Yarowsky, D. (1992) Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-14)*, pp. 454–460.