Journal of Artificial Intelligence Research 17 (2002) 289-308 Submitted 6/02; published 10/02

A Unified Model of Structural Organization in Language and Music

Rens Bod RENS@ILLC.UVA.NL

Institute for Logic, Language and Computation
University of Amsterdam, Nieuwe Achtergracht 166
1018 WV Amsterdam, THE NETHERLANDS, and
School of Computing, University of Leeds
LS2 9JT Leeds, UK

Abstract

Is there a general model that can predict the perceived phrase structure in language and music? While it is usually assumed that humans have separate faculties for language and music, this work focuses on the commonalities rather than on the differences between these modalities, aiming at finding a deeper "faculty". Our key idea is that the perceptual system strives for the simplest structure (the "simplicity principle"), but in doing so it is biased by the likelihood of previous structures (the "likelihood principle"). We present a series of data-oriented parsing (DOP) models that combine these two principles and that are tested on the Penn Treebank and the Essen Folksong Collection. Our experiments show that (1) a combination of the two principles outperforms the use of either of them, and (2) exactly the same model with the same parameter setting achieves maximum accuracy for both language and music. We argue that our results suggest an interesting parallel between linguistic and musical structuring.

1. Introduction: The Problem of Structural Organization

It is widely accepted that the human cognitive system tends to organize perceptual information into hierarchical descriptions that can be conveniently represented by tree structures. Tree structures have been used to describe linguistic perception (e.g. Wundt, 1901; Chomsky, 1965), musical perception (e.g. Longuet-Higgins, 1976; Lerdahl & Jackendoff, 1983) and visual perception (e.g. Palmer, 1977; Marr, 1982). Yet, little attention has been paid to the commonalities between these different forms of perception and to the question whether there exists a general, underlying mechanism that governs all perceptual organization. This paper studies exactly that question: acknowledging the differences between the perceptual modalities, is there a general model that can predict the perceived tree structure for sensory input? In studying this question, we will use an empirical methodology: any model that we might hypothesize will be tested against manually analyzed benchmarks such as the linguistically annotated Penn Treebank (Marcus et al. 1993) and the musically annotated Essen Folksong Collection (Schaffrath, 1995). While we will argue for a general model of structural organization in language, music and vision, we will carry out experiments only with linguistic and musical benchmarks, since no benchmark of visual tree structures is currently available, to the best of our knowledge.

Figure 1 gives three simple examples of linguistic, musical and visual information with their corresponding tree structures printed below (these examples are resp. taken from Martin et al. 1987, Lerdahl & Jackendoff, 1983, and Dastani, 1998).

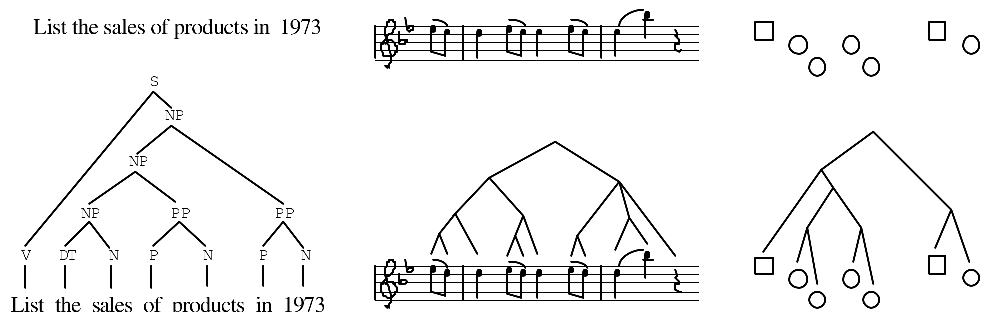

Figure 1: Examples of linguistic, musical and visual input with their tree structures

Thus, a tree structure describes how parts of the input combine into constituents and how these constituents combine into a representation for the whole input. Note that the linguistic tree structure is labeled with syntactic categories, whereas the musical and visual tree structures are unlabeled. This is because in language there are syntactic constraints on how words can be combined into larger constituents (e.g. in English a determiner can be combined with a noun only if it precedes that noun, which is expressed by the rule NP $\rightarrow$ DT N), while in music (and to a lesser extent in vision) there are no such restrictions: in principle any note may be combined with any other note.

Apart from these differences, there is also a fundamental commonality: the perceptual input undergoes a process of hierarchical structuring which is not found in the input itself. The main problem is thus: how can we derive the perceived tree structure for a given input? That this problem is not trivial may be illustrated by the fact that the inputs above can also be assigned the following, alternative tree structures:

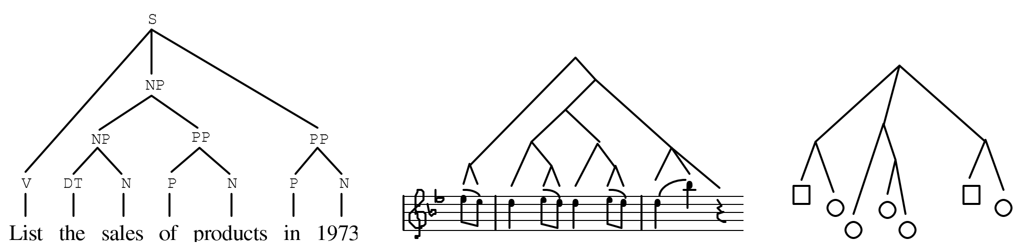

Figure 2: Alternative tree structures for the inputs in Figure 1

These alternative structures are possible in that they *can* be perceived. The linguistic tree structure in Figure 1 corresponds to a meaning which is different from the tree in Figure 2. The two musical tree structures correspond to different groupings into motifs. And the two visual structures correspond to different visual Gestalts. But while the alternative tree structures are all possible, they are not plausible: they do not correspond to the structures that are actually perceived by the human cognitive system.

The phenomenon that the same input may be assigned different structural organizations is known as the *ambiguity problem*. This problem is one of the hardest problems in modeling human perception. Even in language, where a phrase-structure grammar may specify which words can be combined into constituents, the ambiguity problem is notoriously hard (cf. Manning & Schütze, 1999). Charniak (1997: 37) argues that many sentences from the Wall Street Journal have more than one million different parse trees. The ambiguity problem for musical input is even harder, since there are virtually no constraints on how notes may be combined into constituents. Talking about rhythm perception in music, Longuet-Higgins and Lee (1987) note that "Any given sequence of note values is in principle infinitely ambiguous, but this ambiguity is seldom apparent to the listener.".

In the following Section, we will discuss two principles that have traditionally been proposed to solve ambiguity: the likelihood principle and the simplicity principle. In Section 3, we will argue for a new integration of the two principles within the data-oriented parsing framework. Our hypothesis is that the human cognitive system strives for the simplest structure generated by the shortest derivation, but that in doing so it is biased by the frequency of previously perceived structures. In Section 4, we go into the computational aspects of our model. In Section 5, we discuss the linguistic and musical test domains. Section 6 presents an empirical investigation and comparison of our model. Finally, in Section 7, we give a discussion of our approach and go into other combinations of simplicity and likelihood that have been proposed in the literature.

2. Two principles: Likelihood and Simplicity

How can we predict from the set of all possible tree structures the tree that is actually perceived by the human cognitive system? In the field of visual perception, two competing principles have traditionally been proposed to govern structural organization. The first, initiated by Helmholtz (1910), advocates the *likelihood principle*: perceptual input will be organized into the most probable structure. The second, initiated by Wertheimer (1923) and developed by other Gestalt psychologists, advocates the *simplicity principle*: the perceptual system is viewed as finding the simplest rather than the most probable structure (see Chater, 1999, for an overview). These two principles have also been used in linguistic and musical structuring. In the following, we briefly review these principles for each modality.

2.1 Likelihood

The likelihood principle has been particularly influential in the field of natural language processing (see Manning and Schütze, 1999, for a review). In this field, the most appropriate tree structure of a sentence is assumed to be its most likely structure. The likelihood of a tree is computed from the probabilities of its parts (e.g. phrase-structure rules) which are in turn estimated from a large manually analyzed language corpus, i.e. a *treebank*. State-of-the-art probabilistic parsers such as Collins (2000), Charniak (2000) and Bod (2001a) obtain around 90% precision and recall on the Penn Wall Street Journal treebank (Marcus et al. 1993).

The likelihood principle has also been applied to musical perception, e.g. by Raphael (1999) and Bod (2001b/c). As in probabilistic natural language processing, the most probable musical tree structure can be computed from the probabilities of rules or fragments taken from a large annotated musical corpus. A musical benchmark which has been used by some models is the Essen Folksong Collection (Schaffrath, 1995).

Also in vision science, there is a huge interest in probabilistic models (e.g. Hoffman, 1998; Kersten, 1999). Mumford (1999) has even seen fit to declare the Dawning of Stochasticity. Unfortunately, no visual treebanks are currently available.

2.2 Simplicity

The simplicity principle has a long tradition in the field of visual perception psychology (e.g. Restle, 1970; Leeuwenberg, 1971; Simon, 1972; Buffart et al. 1983; van der Helm, 2000). In this field, a visual pattern is formalized as a constituent structure by means of a visual coding language based on primitive elements such as line segments and angles. Perception is described as the process of selecting the simplest structure corresponding to the "shortest encoding" of a visual pattern.

The notion of simplicity has also been applied to music perception. Collard et al. (1981) use the coding language of Leeuwenberg (1971) to predict the metrical structure for four preludes from Bach's *Well-Tempered Clavier*. More well-known in music perception is the theory proposed by Lerdahl and Jackendoff (1983). Their theory contains two kinds of rules: "well-formedness rules" and "preference rules". The role of well-formedness rules is to define the kinds of formal objects (grouping structures) the theory employs. What grouping structures a listener actually hears, is then described by the preference rules which describe Gestalt-preferences of the kind identified by Wertheimer (1923), and which can therefore also be seen as an embodiment of the simplicity principle.

Notions of simplicity also exist in language processing. For example, Frazier (1978) can be viewed as arguing that the parser prefers the simplest structure containing minimal attachments. Bod (2000a) defines the simplest tree structure of a sentence as the structure generated by the smallest number of subtrees from a given treebank.

3. Combining Likelihood and Simplicity

The key idea of the current paper is that both principles play a role in perceptual organization, albeit rather different ones: the simplicity principle as a general cognitive preference for economy, and the likelihood principle as a probabilistic bias due to previous perceptual experiences. Informally stated, our working hypothesis is that the human cognitive system strives for the simplest structure generated by the shortest derivation, but that in doing so it is biased by the frequency of previously perceived structures (some other combinations of simplicity and likelihood will be discussed in Section 7). To formally instantiate our working hypothesis, we first need a model that defines the set of *possible* structures of an input. In this paper, we have chosen for a model that defines the set of phrase-structures for an input on the basis of a treebank of previously analyzed input, and which is known as the Data-Oriented Parsing or DOP model (see Bod, 1998; Collins & Duffy, 2002). DOP learns a grammar by extracting subtrees from a given treebank and combines these subtrees to analyze fresh input. We have chosen DOP because (1) it uses subtrees of arbitrary size, thereby capturing non-local dependencies, and (2) it has obtained very competitive results on various benchmarks (Bod, 2001a/b; Collins & Duffy, 2002). In the following, we first review the DOP model and discuss the use of the likelihood and simplicity principles by this approach. Next, we show how these two principles can be combined to instantiate our working hypothesis.

3.1 Data-Oriented Parsing

In this Section, we illustrate the DOP model with a linguistic example (for a rigorous definition of DOP, the reader is referred to Bod, 1998). We will come back to some musical examples in Section 5. Suppose we are given the following extremely small linguistic treebank of two trees for resp. *she wanted the dress on the rack* and *she saw the dog with the telescope* (actual treebanks contain tens of thousands of trees, cf. Marcus et al. 1993):

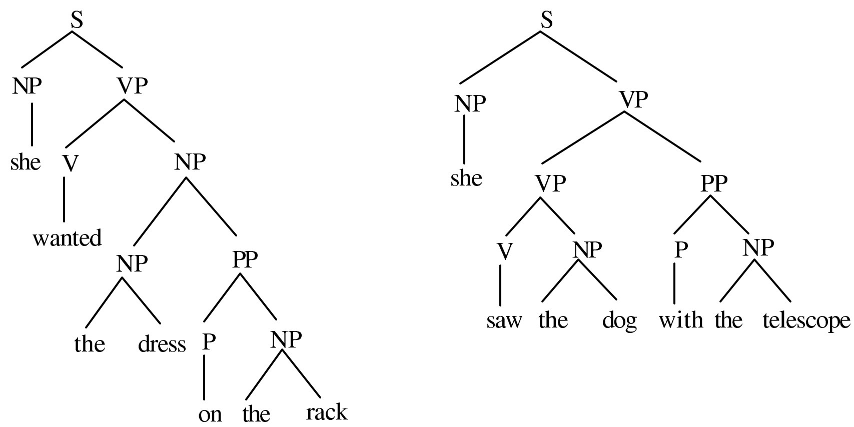

Figure 3: An example treebank

The DOP model can parse a new sentence, e.g. *She saw the dress with the telescope*, by combining subtrees from this treebank by means of a *substitution operation* (indicated as ∘):

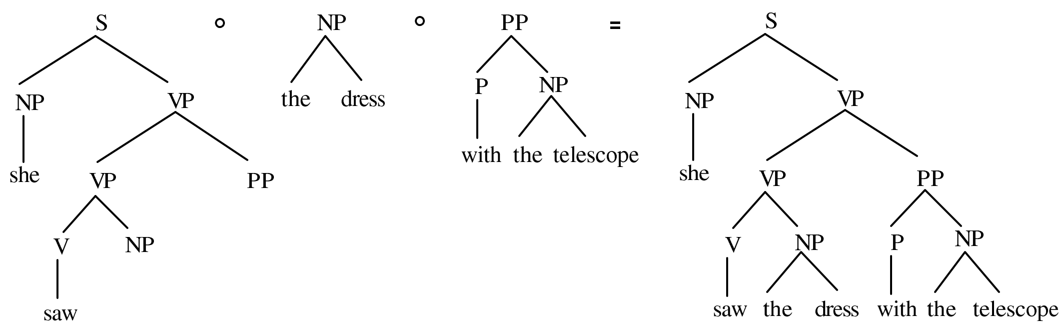

Figure 4: Parsing a sentence by combining subtrees from Figure 3

Thus the substitution operation combines two subtrees by substituting the second subtree on the leftmost nonlexical leaf node of the first subtree (the result of which may be combined with a third subtree, etc.). A combination of subtrees that results in a tree structure for the whole sentence is called a *derivation*. Since there are many different subtrees, of various sizes, there are typically also many different derivations that produce, however, the *same* tree; for instance:

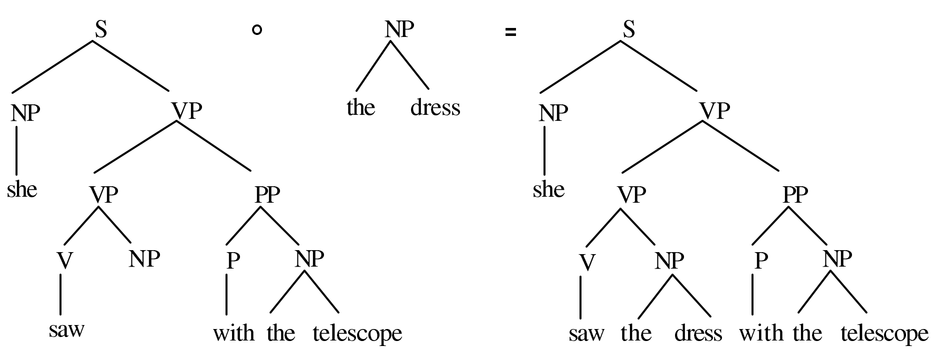

Figure 5: A different derivation which produces the same parse tree

The more interesting case occurs when there are different derivations that produce *different* parse trees. This happens when a sentence is ambiguous; for example, DOP also produces the following alternative parse tree for *She saw the dress with the telescope*:

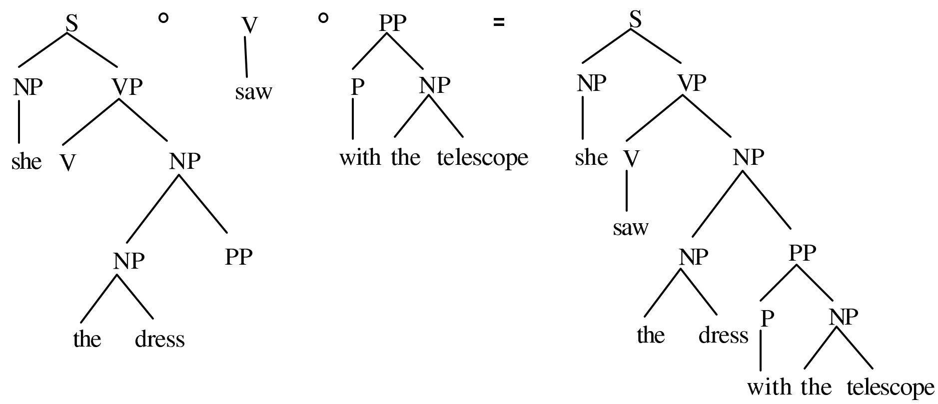

Figure 6: A different derivation which produces a different parse tree

3.2 Likelihood-DOP

In Bod (1993), DOP is enriched with the likelihood principle to predict the perceived tree structure from the set of possible structures. This model, which we will call *Likelihood-DOP*, computes the most probable tree of an input from the occurrence-frequencies of the subtrees. The probability of a subtree t, $P(t)$, is computed as the number of occurrences of t, $|\,t\,|$, divided by the total number of occurrences of treebank-subtrees that have the same root label as t. Let $r(t)$ return the root label of t. Then we may write:

$$P(t) \;=\; \frac{|\,t\,|}{\sum_{t':\; r(t')=r(t)} \;|\,t'\,|}$$

The probability of a derivation $t_1 \circ ... \circ t_n$ is computed by the product of the probabilities of its subtrees t_i:

$$P(t_1 \circ ... \circ t_n) \;=\; \prod_i P(t_i)$$

As we have seen, there may be different derivations that generate the same parse tree. The probability of a parse tree T is thus the sum of the probabilities of its distinct derivations. Let t_{id} be the i-th subtree in the derivation d that produces tree T, then the probability of T is given by

$$P(T) \;=\; \sum_d \prod_i P(t_{id})$$

In parsing a sentence s, we are only interested in the trees that can be assigned to s, which we denote by T_s. The best parse tree, T_{best}, according to Likelihood-DOP is then the tree which maximizes the probability of T_s:

$$T_{best} = \arg\max_{T_s} P(T_s)$$

Thus Likelihood-DOP computes the probability of a tree as a sum of products, where each product corresponds to the probability of a certain derivation generating the tree. This distinguishes Likelihood-DOP from most other statistical parsing models that identify exactly one derivation for each parse tree and thus compute the probability of a tree by only one product of probabilities (e.g. Charniak, 1997; Collins, 1999; Eisner, 1997). Likelihood-DOP's probability model allows for including counts of subtrees of a wide range of sizes: everything from counts of single-level rules to counts of entire trees.

Note that the subtree probabilities in Likelihood-DOP are directly estimated from their relative frequencies in the treebank-trees. While the relative-frequency estimator obtains competitive results on several domains (Bonnema et al. 1997; Bod, 2001a; De Pauw, 2000), it does not maximize the likelihood of the training data (Johnson, 2002). This is because there may be hidden derivations which the relative-frequency estimator cannot deal with.[1] There are estimation procedures that do take into account hidden derivations and that maximize the likelihood of the training data. For example, Bod (2000b) presents a Likelihood-DOP model which estimates the subtree probabilities by a maximum likelihood re-estimation procedure based on the expectation-maximization algorithm (Dempster et al. 1977). However, since the relative frequency estimator has so far not been outperformed by any other estimator (see Bod et al. 2002b), we will stick to the relative frequency estimator for the current paper.

3.3 Simplicity-DOP

Likelihood-DOP does not do justice to the preference humans display for the simplest structure generated by the shortest derivation of an input. In Bod (2000a), the simplest tree structure of an input is defined as the tree that can be constructed by the smallest number of subtrees from a treebank. We will refer to this model as *Simplicity-DOP*. Instead of producing the most probable parse tree for an input, Simplicity-DOP thus produces the parse tree generated by the shortest derivation consisting of the fewest treebank-subtrees, *independent* of the probabilities of these subtrees. We define the *length* of a derivation d, $L(d)$, as the number of subtrees in d; thus if $d = t_1 \circ ... \circ t_n$ then $L(d) = n$. Let d_T be a derivation which results in parse tree T, then the best parse tree, T_{best}, according to Simplicity-DOP is the tree which is produced by a derivation of minimal length:

$$T_{best} = \arg\min_{T_s} L(d_{T_s})$$

As in Section 3.2, T_s is a parse tree of a sentence s. For example, given the treebank in Figure 3, the simplest parse tree for *She saw the dress with the telescope* is given in Figure 5, since

[1] Only if the subtrees are restricted to depth 1 does the relative frequency estimator coincide with the maximum likelihood estimator. Such a depth-1 DOP model corresponds to a stochastic context-free grammar. It is well-known that DOP models which allow subtrees of greater depth outperform depth-1 DOP models (Bod, 1998; Collins & Duffy, 2002).

that parse tree can be generated by a derivation of only two treebank-subtrees, while the parse tree in Figure 6 (and any other parse tree) needs at least three treebank-subtrees to be generated.[2]

The shortest derivation may not be unique: it can happen that different parse trees of a sentence are generated by the same minimal number of treebank-subtrees (also the most probable parse tree may not be unique, but this never happens in practice). In that case we will back off to a frequency ordering of the subtrees. That is, all subtrees of each root label are assigned a rank according to their frequency in the treebank: the most frequent subtree (or subtrees) of each root label gets rank 1, the second most frequent subtree gets rank 2, etc. Next, the rank of each (shortest) derivation is computed as the sum of the ranks of the subtrees involved. The derivation with the smallest sum, or highest rank, is taken as the final best derivation producing the final best parse tree in Simplicity-DOP (see Bod, 2000a).

We performed one little adjustment to the rank of a subtree. This adjustment averages the rank of a subtree by the ranks of its own sub-subtrees. That is, instead of simply taking the rank of a subtree, we compute the rank of a subtree as the (arithmetic) mean of the ranks of all its sub-subtrees (including the subtree itself). The effect of this technique is that it redresses a very low-ranked subtree if it contains high-ranked sub-subtrees.

While Simplicity-DOP and Likelihood-DOP obtain rather similar parse accuracy on the Wall Street Journal and the Essen Folksong Collection (in terms of precision/recall -- see Section 6), the best trees predicted by the two models do not quite match. This suggests that a combined model, which does justice to both simplicity and likelihood, may boost the accuracy.

3.4 Combining Likelihood-DOP and Simplicity-DOP: SL-DOP and LS-DOP

The underlying idea of combining likelihood and simplicity is that the human perceptual system searches for the simplest tree structure (generated by the shortest derivation) but in doing so it is biased by the likelihood of the tree structure. That is, instead of selecting the simplest tree *per se*, our combined model selects the simplest tree from among the n likeliest trees, where n is our free parameter. There are of course other ways to combine simplicity and likelihood within the DOP framework. A straightforward alternative would be to select the most probable tree from among the n simplest trees, suggesting that the perceptual system is searching for the most probable structure only from among the simplest ones. We will refer to the first combination of simplicity and likelihood (which selects the simplest among the n likeliest trees) as *Simplicity-Likelihood-DOP* or *SL-DOP*, and to the second combination (which selects the likeliest among the n simplest trees) as *Likelihood-Simplicity-DOP* or *LS-DOP*. Note that for n=1, Simplicity-Likelihood-DOP is equal to Likelihood-DOP, since there is only one most probable tree to select from, and Likelihood-Simplicity-DOP is equal to Simplicity-DOP, since there is only one simplest tree to select from. Moreover, if n gets large, SL-DOP converges to Simplicity-DOP while LS-DOP converges to Likelihood-DOP. By varying the parameter n, we will be able to compare Likelihood-DOP, Simplicity-DOP and several instantiations of SL-DOP and LS-DOP.

[2] One might argue that a more straightforward metric of simplicity would return the parse tree with the smallest number of nodes (rather than the smallest number of treebank-subtrees). But such a metric is known to perform quite badly (see Manning & Schütze, 1999; Bod, 2000a).

4. Computational Issues

Bod (1993) showed how standard chart parsing techniques can be applied to Likelihood-DOP. Each treebank-subtree t is converted into a context-free rule r where the lefthand side of r corresponds to the root label of t and the righthand side of r corresponds to the frontier labels of t. Indices link the rules to the original subtrees so as to maintain the subtree's internal structure and probability. These rules are used to create a derivation forest for a sentence (using a chart parser -- see Charniak, 1993), and the most probable parse is computed by sampling a sufficiently large number of random derivations from the forest ("Monte Carlo disambiguation", see Bod, 1998). While this technique has been successfully applied to parsing the ATIS portion in the Penn Treebank (Marcus et al. 1993), it is extremely time consuming. This is mainly because the number of random derivations that should be sampled to reliably estimate the most probable parse increases exponentially with the sentence length (see Goodman, 2002). It is therefore questionable whether Bod's sampling technique can be scaled to larger domains such as the Wall Street Journal (WSJ) portion in the Penn Treebank.

Goodman (1996) showed how Likelihood-DOP can be reduced to a compact stochastic context-free grammar (SCFG) which contains exactly eight SCFG rules for each node in the training set trees. Although Goodman's method does still not allow for an efficient computation of the most probable parse (in fact, the problem of computing the most probable parse in Likelihood-DOP is NP-hard -- see Sima'an, 1996), his method does allow for an efficient computation of the "maximum constituents parse", i.e. the parse tree that is most likely to have the largest number of correct constituents. Unfortunately, Goodman's SCFG reduction method is only beneficial if indeed *all* subtrees are used, while maximum parse accuracy is usually obtained by restricting the subtrees. For example, Bod (2001a) shows that the "optimal" subtree set achieving highest parse accuracy on the WSJ is obtained by restricting the maximum number of words in each subtree to 12 and by restricting the maximum depth of unlexicalized subtrees to 6. Goodman (2002) shows that some subtree restrictions, such as subtree depth, may be incorporated by his reduction method, but we have found no reduction method for our optimal subtree set.

In this paper we will therefore use Bod's subtree-to-rule conversion method for Likelihood-DOP, but we will not use Bod's Monte Carlo sampling technique from derivation forests, as this turned out to be computationally prohibitive. Instead, we will use the well-known Viterbi optimization algorithm for chart parsing (cf. Charniak, 1993; Manning & Schütze, 1999) which allows for computing the k most probable derivations of an input in cubic time. Using this algorithm, we will estimate the most probable parse tree of an input from the 10,000 most probable derivations, summing up the probabilities of derivations that generate the same tree. Although this approach does not guarantee that the most probable parse tree is actually found, it is shown in Bod (2000a) to perform at least as well as the estimation of the most probable parse by Monte Carlo techniques on the ATIS corpus. Moreover, this approach is known to obtain significantly higher accuracy than selecting the parse tree generated by the single most probable derivation (Bod, 1998; Goodman, 2002), which we will therefore not consider in this paper.

For Simplicity-DOP, we also first convert the treebank-subtrees into rewrite rules just as with Likelihood-DOP. Next, the simplest tree, i.e. the shortest derivation, can be efficiently computed by Viterbi optimization in the same way as the most probable derivation, provided that we assign all rules equal probabilities, in which case the shortest derivation is equal to the most probable derivation. This can be seen as follows: if each rule has a probability p then the probability of a derivation involving n rules is equal to p^n, and since $0<p<1$ the derivation with the fewest rules has the greatest probability. In our experiments in Section 6, we give each rule a probability mass equal to $1/R$, where R is the number of distinct rules derived by Bod's method. As mentioned in 3.3, the shortest derivation may not be unique. In that case we compute all shortest derivations of an input and then apply our ranking scheme to these derivations. The ranks of the shortest derivations are computed by summing up the ranks of the subtrees they involve. The shortest derivation with the smallest sum of subtree ranks is taken to produce the best parse tree.

For SL-DOP and LS-DOP, we compute either n likeliest or n simplest trees by means of Viterbi optimization. Next, we either select the simplest tree among the n likeliest ones (for SL-DOP) or the likeliest tree among the n simplest ones (for LS-DOP). In our experiments, n will never be larger than 1,000.

5. The Test Domains

As our linguistic test domain we used the Wall Street Journal (WSJ) portion in the Penn Treebank (Marcus et al. 1993). This portion contains approx. 50,000 sentences that have been manually annotated with the perceived linguistic tree structures using a predefined set of lexico-syntactic labels. Since the WSJ has been extensively used and described in the literature (cf. Manning & Schütze, 1999; Charniak, 2000; Collins, 2000; Bod, 2001a), we will not go into it any further here.

As our musical test domain we used the European folksongs in the Essen Folksong Collection (Schaffrath, 1995; Huron, 1996), which correspond to approx. 6,200 folksongs that have been manually enriched with their perceived musical grouping structures. The Essen Folksong Collection has been previously used by Bod (2001b) and Temperley (2001) to test their musical parsers. The current paper presents the first experiments with Likelihood-DOP, Simplicity-DOP, SL-DOP and LS-DOP on this collection. The Essen folksongs are not represented by staff notation but are encoded by the Essen Associative Code (ESAC). The pitch encodings in ESAC resemble "solfege": scale degree numbers are used to replace the movable syllables "do", "re", "mi", etc. Thus 1 corresponds to "do", 2 corresponds to "re", etc. Chromatic alterations are represented by adding either a "#" or a "b" after the number. The plus ("+") and minus ("-") signs are added before the number if a note falls resp. above or below the principle octave (thus -1, 1 and +1 refer al to "do", but on different octaves). Duration is represented by adding a period or an underscore after the number. A period (".") increases duration by 50% and an underscore ("_") increases duration by 100%; more than one underscore may be added after each number. If a number has no duration indicator, its duration corresponds to the smallest value. Thus pitches in ESAC are encoded by integers from 1 to 7 possibly preceded or followed by symbols for octave, chromatic alteration and duration. Each pitch encoding is treated as an atomic symbol, which may be as simple as "1" or as complex as "+2#_.". A pause is represented by 0, possibly followed by duration indicators, and is also treated as an atomic symbol. No loudness or timbre indicators are used in ESAC.

Phrase boundaries are indicated by hard returns in ESAC. The phrases are unlabeled (cf. Section 1 of this paper). Yet to make the ESAC annotations readable for our DOP models, we added three basic labels to the phrase structures: the label "S" to each whole song, the label "P" to each phrase, and the label "N" to each atomic symbol. In this way, we obtained conventional tree structures that could directly be employed by our DOP models to parse new input. The use of the label "N" distinguishes our annotations from those in previous work (Bod, 2001b/c) where we only used labels for song and phrase ("S" and "P"). The addition of "N" enhances the productivity and robustness of the musical parsing model, although it also leads to a much larger number of subtrees.

As an example, assume a very simple melody consisting of two phrases, (1 2) (2 3), then its tree structure is given in Figure 7.

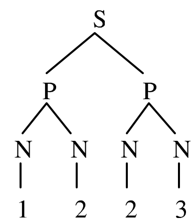

Figure 7: Example of a musical tree structure consisting of two phrases

Subtrees that can be extracted from this tree structure include the following:

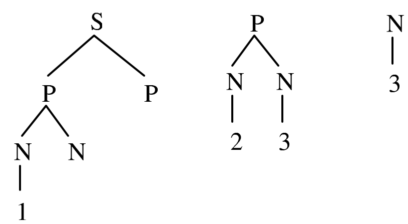

Figure 8: Some subtrees that can be extracted from the tree in figure 7

Thus the first subtree indicates a phrase starting with a note 1, followed by exactly one other (unspecified) note, with the phrase itself followed by exactly one other (unspecified) phrase. Such subtrees can be used to parse new musical input in the same way as has been explained for linguistic parsing in Section 3.

6. Experimental Evaluation and Comparison

To evaluate our DOP models, we used the blind testing method which randomly divides a treebank into a training set and a test set, where the strings from the test set are parsed by means of the subtrees from the training set. We applied the standard PARSEVAL metrics of *precision* and *recall* to compare a proposed parse tree P with the corresponding correct test set parse tree T as follows (cf. Black et al. 1991):

$$\text{Precision} = \frac{\#\text{ correct constituents in } P}{\#\text{ constituents in } P} \qquad \text{Recall} = \frac{\#\text{ correct constituents in } P}{\#\text{ constituents in } T}$$

A constituent in P is "correct" if there exists a constituent in T of the same label that spans the same atomic symbols (i.e. words or notes).[3] Since precision and recall can obtain rather

[3] The precision and recall scores were computed by using the "evalb" program (available via http://www.cs.nyu.edu/cs/projects/proteus/evalb/)

different results (see Bod, 2001b), they are often balanced by a single measure of performance, known as the F-score (see Manning & Schütze, 1999):

$$\text{F-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

For our experiments, we divided both treebanks (i.e. the WSJ and the Essen Folksong Collection) into 10 training/test set splits: 10% of the WSJ was used as test material each time (sentences ≤ 40 words), while for the Essen Folksong Collection test sets of 1,000 folksongs were used each time. For words in the test set that were unknown in the training set, we guessed their categories by using statistics on word-endings, hyphenation and capitalization (cf. Bod, 2001a); there were no unknown notes. As in previous work (Bod, 2001a), we limited the maximum size of the subtrees to depth 14, and used random samples of 400,000 subtrees for each depth > 1 and ≤ 14.[4] Next, we restricted the maximum number of atomic symbols in each subtree to 12 and the maximum depth of unlexicalized subtrees to 6. All subtrees were smoothed by the technique described in Bod (1998: 85-94) based on simple Good-Turing estimation (Good, 1953).

Table 1 shows the mean F-scores obtained by SL-DOP and LS-DOP for language and music and for various values of *n*. Recall that for *n*=1, SL-DOP is equal to Likelihood-DOP while LS-DOP is equal to Simplicity-DOP.

	SL-DOP (simplest among *n* likeliest)		LS-DOP (likeliest among *n* simplest)	
n	Language	Music	Language	Music
1	87.9%	86.0%	85.6%	84.3%
5	89.3%	86.8%	86.1%	85.5%
10	90.2%	87.2%	87.0%	85.7%
11	90.2%	87.3%	87.0%	85.7%
12	90.2%	87.3%	87.0%	85.7%
13	90.2%	87.3%	87.0%	85.7%
14	90.2%	87.2%	87.0%	85.7%
15	90.2%	87.2%	87.0%	85.7%
20	90.0%	86.9%	87.1%	85.7%
50	88.7%	85.6%	87.4%	86.0%
100	86.8%	84.3%	87.9%	86.0%
1,000	85.6%	84.3%	87.9%	86.0%

Table 1: F-scores obtained by SL-DOP and LS-DOP for language and music

[4] These random subtree samples were not selected by first exhaustively computing the complete set of subtrees (this was computationally prohibitive). Instead, for each particular depth > 1 we sampled subtrees by randomly selecting a node in a random tree from the training set, after which we selected random expansions from that node until a subtree of the particular depth was obtained. We repeated this procedure 400,000 times for each depth > 1 and ≤ 14.

The Table shows that there is an increase in accuracy for both SL-DOP and LS-DOP if the value of n increases from 1 to 11. But while the accuracy of SL-DOP decreases after n=13 and converges to Simplicity-DOP (i.e. LS-DOP at n=1), the accuracy of LS-DOP continues to increase and converges to Likelihood-DOP (i.e. SL-DOP at n=1). The highest accuracy is obtained by SL-DOP at $11 \leq n \leq 13$, for both language and music. Thus SL-DOP outperforms both Likelihood-DOP and Simplicity-DOP, and the selection of the simplest structure out of the top likeliest ones turns out to be a more promising model than the selection of the likeliest structure out of the top simplest ones. According to paired *t*-testing, the accuracy improvement of SL-DOP at n=11 over SL-DOP at n=1 (when it is equal Likelihood-DOP) is statistically significant for both language ($p<.0001$) and music ($p<.006$).

It is surprising that SL-DOP reaches highest accuracy at such a small value for n. But it is even more surprising that exactly the same model (with the same parameter setting) obtains maximum accuracy for both language and music. This model embodies the idea that the perceptual system strives for the simplest structure but in doing so it only searches among a few most probable structures.

To compare our results for language with others, we also tested SL-DOP at n=11 on the now standard division of the WSJ, which uses sections 2 to 21 for training (approx. 40,000 sentences) and section 23 for testing (2416 sentences ≤ 100 words) (see e.g. Manning & Schütze, 1999; Charniak, 2000; Collins, 2000). On this division, SL-DOP achieved an F-score of 90.7% while the best previous models obtained an F-score of 89.7% (Collins, 2000; Bod, 2001a). In terms of error reduction, SL-DOP improves with 9.6% over these other models. It is common to also report the accuracy for sentences ≤ 40 words on the WSJ, for which SL-DOP obtained an F-score of 91.8%.

Our musical results can be compared to Bod (2001b/c), who tested three probabilistic parsing models of increasing complexity on the same training/test set splits from the Essen Folksong Collection. The best results were obtained with a hybrid DOP-Markov parser: 80.7% F-score. This is significantly worse than our best result of 87.3% obtained by SL-DOP on the same splits from the Essen folksongs. This difference may be explained by the fact that the hybrid DOP-Markov parser in Bod (2001b/c) only takes into account context from higher nodes in the tree and not from any sister nodes, while the DOP models presented in the current paper take any subtree into account of (almost) arbitrary width and depth, thereby covering a larger amount of musical context. Moreover, as mentioned in Section 5, the models in Bod (2001b/c) did not use the label "N" for notes; instead, a Markov approach was used to parse new sequences of notes.

It would also be interesting to compare our musical results to the melodic parser of Temperley (2001), who uses a system of preference rules similar to Lerdahl and Jackendoff (1983), and which is also evaluated on the Essen Folksong Collection. But while we have tested on several test sets of 1,000 randomly selected folksongs, Temperley used only one test set of 65 folksongs that was moreover cleaned up by eliminating folksongs with irregular meter (Temperley, 2001: 74). It is therefore difficult to compare our results with Temperley's; yet, it is noteworthy that Temperley's parser correctly identified 75.5% of the phrase boundaries. Although this is lower than the 87.3% obtained by SL-DOP, Temperley's parser is not "trained" on previously analyzed examples like our model (though we note that Temperley's results were obtained by tuning the optimal phrase length of his parser on the average phrase length of the Essen Folksong Collection).

It should perhaps be mentioned that while parsing models trained on treebanks are widely used in natural language processing, they are still rather uncommon in musical processing.

Most musical parsing models, including Temperley's, employ a rule-based approach where the parsing is based on a combination of low-level rules -- such as "prefer phrase boundaries at large intervals" -- and higher-level rules -- such as "prefer phrase boundaries at changes of harmony". The low-level rules are usually based on the well-known Gestalt principles of proximity and similarity (Wertheimer, 1923), which prefer phrase boundaries at larger intervallic distances. However, in Bod (2001c) we have shown that the Gestalt principles predict incorrect phrase boundaries for a number of folksongs, and that higher-level phenomena cannot alleviate these incorrect predictions. These folksongs contain a phrase boundary which falls just *before* or *after* a large pitch or time interval (which we have called *jump-phrases*) rather than *at* such intervals -- as would be predicted by the Gestalt principles. Moreover, other musical factors, such as melodic parallelism, meter and harmony, predict exactly the same incorrect phrase boundaries for these cases (see Bod, 2001b/c for details). We have conjectured that such jump-phrases are inherently memory-based, reflecting idiom-dependent pitch contours (cf. Huron, 1996; Snyder, 2000), and that they can be best captured by a memory-based model that tries to mimic the musical experience of a listener from a certain culture (Bod, 2001c).

7. Discussion and Conclusion

We have seen that our combination of simplicity and likelihood is quite rewarding for linguistic and musical structuring, suggesting an interesting parallel between the two modalities. Yet, one may question whether a model which massively memorizes and re-uses previously perceived structures has any cognitive plausibility. Although this question is only important if we want to claim cognitive relevance for our model, there appears to be some evidence that people store various kinds of previously heard fragments, both in language (Jurafsky, 2002) and music (Saffran et al. 2000). But do people store fragments of *arbitrary* size, as proposed by DOP? In his overview article, Jurafsky (2002) reports on a large body of psycholinguistic evidence showing that people not only store lexical items and bigrams, but also frequent phrases and even whole sentences. For the case of sentences, people not only store idiomatic sentences, but also "regular" high-frequency sentences.[5] Thus, at least for language there is some evidence that humans store fragments of arbitrary size provided that these fragments have a certain minimal frequency. And this suggests that humans need not always parse new input by the rules of a grammar, but that they can productively re-use previously analyzed fragments. Yet, there is no evidence that people store *all* fragments they hear, as suggested by DOP. Only high-frequency fragments seem to be memorized. However, if the human perceptual faculty needs to *learn* which fragments will be stored, it will initially need to keep track of all fragments (with the possibility of forgetting them) otherwise frequencies can never accumulate. This results in a model which continuously and incrementally updates its fragment memory given new input -- which is in correspondence with the DOP approach, and also with some other approaches (cf. Daelemans, 1999; Scha et al. 1999; Spiro, 2002). While we acknowledge the importance of a rule-based system in acquiring a fragment memory, once a substantial memory is available it may be more efficient to construct a tree by means of already parsed fragments than constructing it entirely by means of rules. For many cognitive

[5] These results are derived from differences in reaction times in sentence recognition where only the frequency of the (whole) test sentences is varied, while all other variables, such as lexical frequency, bigram frequency, plausibility, syntactic/semantic complexity, etc., are kept constant.

activities it is advantageous to store results, so that they can immediately be retrieved from memory, rather than computing them each time from scratch. This has been shown, for example, for manual reaches (Rosenbaum et al. 1992), arithmetic operations (Rickard et al. 1994), word formation (Baayen et al. 1997), to mention a few. And linguistic and musical parsing may be no exception to this.

It should be stressed that the experiments reported in this paper are limited in at least two respects. First, our musical test domain is rather restricted. While a wide variety of linguistic treebanks is currently available (see Manning & Schütze, 1999), the number of musical treebanks is extremely limited. There is thus a need for larger and richer annotated musical corpora covering broader domains. The development of such annotated corpora may be time-consuming, but experience from natural language processing has shown that it is worth the effort, since corpus-based parsing systems dramatically outperform grammar-based parsing systems. A second limitation of our experiments is that we have only evaluated the parse *results* rather than the parse *process*. That is, we have only assessed how accurately our models can mimic the input-output behavior of a human annotator, without investigating the process by which an annotator arrived at the perceived structures. It is unlikely that humans process perceptual input by computing 10,000 most likely derivations using random samples of 400,000 subtrees – as we did in the current paper. Yet, for many applications it suffices to know the perceived structure rather than the process that led to that structure. And we have seen that our combination of simplicity and likelihood predicts the perceived structure with a high degree of accuracy.

There have been other proposals for integrating the principles of simplicity and likelihood in human perception (see Chater, 1999 for a review). Chater notes that in the context of Information Theory (Shannon, 1948), the principles of simplicity and likelihood are identical. In this context, the simplicity principle is interpreted as minimizing the expected length to encode a message i, which is $-log_2 p_i$ bits, and which leads to the same result as maximizing the probability of i. If we used this information-theoretical definition of simplest structure in Simplicity-DOP, it would return the same structure as Likelihood-DOP, and no improved results would be obtained by a combination of the two. On the other hand, by defining the simplest structure as the one generated by the smallest number of subtrees, independent of their probabilities, we created a notion of simplicity which is provably different from the notion of most likely structure, and which, combined with Likelihood-DOP, obtained improved results.

Another integration of the two principles may be provided by the notion of Minimum Description Length or MDL (cf. Rissanen, 1978). The MDL principle can be viewed as preferring the statistical model that allows for the shortest encoding of the training data. The relevant encoding consists of two parts: the first part encodes the model of the data, and the second part encodes the data in terms of the model (in bit length). MDL is closely related to stochastic complexity (Rissanen, 1989) and Kolmogorov complexity (Li and Vitanyi, 1997), and has been used in natural language processing for estimating the parameters of a stochastic grammar (e.g. Osborne, 1999). We will leave it as an open research question as to whether MDL can be successfully used for estimating the parameters of DOP's subtrees. However, since MDL is known to give asymptotically the same results as maximum likelihood estimation (MLE) (Rissanen, 1989), its application to DOP may lead to an unproductive model. This is because the maximum likelihood estimator will assign the training set trees their empirical frequencies, and assign 0 weight to all other trees (see Bonnema, 2002 for a proof). This would result in a model which can only generate the training data and no other strings. Johnson (2002) argues that this may be an overlearning problem rather than a problem with

MLE per se, and that standard methods, such as cross-validation or regularization, would seem in principle to be ways to avoid such overlearning. We will leave this issue to future investigation.

The idea of a general underlying model for language and music is not uncontroversial. In linguistics it is usually assumed that humans have a separate language faculty, and Lerdahl and Jackendoff (1983) have argued for a separate music faculty. This work does not propose that these separate faculties do not exist, but wants to focus on the commonalities rather than on the differences between these faculties, aiming at finding a deeper "faculty" which may hold for perception in general. Our hypothesis is that the perceptual system strives for the simplest structure but in doing so it only searches among the likeliest structures.

Acknowledgements

Thanks to Aline Honingh, Remko Scha, Neta Spiro, Menno van Zaanen and three anonymous reviewers for their excellent comments. A preliminary version of this paper was presented as a keynote talk at the LCG workshop ("Learning Computational Grammars", Tübingen, 2001).

References

Baayen, R. H., Dijkstra, T. & Schreuder, R. (1997). Singular and Plurals in Dutch: Evidence for a Parallel Dual-Route Model. *Journal of Memory and Language*, 37, 94-117.

Black, E., Abney, S., Flickinger, D., Gnadiec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B. & Strzalkowski, T. (1991). A Procedure for Quantitatively Comparing the Syntactic Coverage of English, *In Proceedings DARPA Speech and Natural Language Workshop*, Pacific Grove, Morgan Kaufmann.

Bod, R. (1993). Using an Annotated Language Corpus as a Virtual Stochastic Grammar. In *Proceedings AAAI-93*, Menlo Park, Ca.

Bod, R. (1998). *Beyond Grammar: An Experience-Based Theory of Language*. Stanford: CSLI Publications (Lecture notes number 88).

Bod, R. (2000a). Parsing with the Shortest Derivation. In *Proceedings COLING-2000*, Saarbrücken, Germany.

Bod, R. (2000b). Combining Semantic and Syntactic Structure for Language Modeling. In *Proceedings ICSLP-2000*, Beijing, China.

Bod, R. (2001a). What is the Minimal Set of Fragments that Achieves Maximal Parse Accuracy? In *Proceedings ACL'2001*, Toulouse, France.

Bod, R. (2001b). A Memory-Based Model for Music Analysis. In *Proceedings International Computer Music Conference* (ICMC'2001), Havana, Cuba.

Bod, R. (2001c). Memory-Based Models of Melodic Analysis: Challenging the Gestalt Principles. *Journal of New Music Research,* 31(1), 26-36. (available at http://staff.science.uva.nl/~rens/jnmr01.pdf)

Bod, R., Hay, J. & Jannedy, S. (Eds.) (2002a). *Probabilistic Linguistics*. Cambridge, The MIT Press. (in press)

Bod, R., Scha, R. & Sima'an, K. (Eds.) (2002b). *Data-Oriented Parsing*. Stanford, CSLI Publications. (in press)

Bonnema, R. (2002). Probability Models for DOP. In Bod et al. (2002b).

Bonnema, R., Bod, R. & Scha, R. (1997). A DOP Model for Semantic Interpretation, In *Proceedings ACL/EACL-97*, Madrid, Spain.

Buffart, H., Leeuwenberg, E. & Restle, F. (1983). Analysis of Ambiguity in Visual Pattern Completion. *Journal of Experimental Psychology: Human Perception and Performance*. 9, 980-1000.

Charniak, E. (1993). *Statistical Language Learning*, Cambridge, The MIT Press.

Charniak, E. (1997). Statistical Techniques for Natural Language Parsing, *AI Magazine*, Winter 1997, 32-43.

Charniak, E. (2000). A Maximum-Entropy-Inspired Parser. In *Proceedings ANLP-NAACL'2000*, Seattle, Washington.

Chater, N. (1999). The Search for Simplicity: A Fundamental Cognitive Principle? *The Quarterly Journal of Experimental Psychology*, 52A(2), 273-302.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*, Cambridge, The MIT Press.

Collard, R., Vos, P. & Leeuwenberg, E. (1981). What Melody Tells about Metre in Music. *Zeitschrift für Psychologie*. 189, 25-33.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*, PhD-thesis, University of Pennsylvania, PA.

Collins, M. (2000). Discriminative Reranking for Natural Language Parsing, In *Proceedings ICML-2000*, Stanford, Ca.

Collins, M. & Duffy, N. (2002). New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron. In *Proceedings ACL'2002*, Philadelphia, PA.

Daelemans, W. (1999). Introduction to Special Issue on Memory-Based Language Processing. *Journal of Experimental and Theoretical Artificial Intelligence* 11(3), 287-296.

Dastani, M. (1998). *Languages of Perception*. ILLC Dissertation Series 1998-05, University of Amsterdam.

Dempster, A., Laird, N. & Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society*, 39, 1-38.

De Pauw, G. (2000). Aspects of Pattern-matching in Data-Oriented Parsing, In *Proceedings COLING-2000*, Saarbrücken, Germany.

Eisner, J. (1997). Bilexical Grammars and a Cubic-Time Probabilistic Parser, In *Proceedings Fifth International Workshop on Parsing Technologies*, Boston, Mass.

Frazier, L. (1978). *On Comprehending Sentences: Syntactic Parsing Strategies*. PhD. Thesis, University of Connecticut.

Good, I. (1953). The Population Frequencies of Species and the Estimation of Population Parameters, *Biometrika* 40, 237-264.

Goodman, J. (1996). Efficient Algorithms for Parsing the DOP Model, In *Proceedings Empirical Methods in Natural Language Processing*, Philadelphia, PA.

Goodman, J. (2002). Efficient Parsing of DOP with PCFG-Reductions. In Bod et al. 2002b.

von Helmholtz, H. (1910). *Treatise on Physiological Optics* (Vol. 3), Dover, New York.

Hoffman, D. (1998). *Visual Intelligence*. New York, Norton & Company, Inc.

Huron, D. (1996). The Melodic Arch in Western Folksongs. *Computing in Musicology* 10, 2-23.

Johnson, M. (2002). The DOP Estimation Method is Biased and Inconsistent. *Computational Linguistics*, 28, 71-76.

Jurafsky, D. (2002). Probabilistic Modeling in Psycholinguistics: Comprehension and Production. In Bod et al. 2002a. (available at http://www.colorado.edu/ling/jurafsky/prob.ps)

Kersten, D. (1999). High-level vision as statistical inference. In Gazzaniga , S. (Ed.), *The New Cognitive Neurosciences*, Cambridge, The MIT Press.

Leeuwenberg, E. (1971). A Perceptual Coding Language for Perceptual and Auditory Patterns. *American Journal of Psychology*. 84, 307-349.

Lerdahl, F. & Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. Cambridge, The MIT Press.

Li, M. & Vitanyi, P. (1997). *An Introduction to Kolmogorov Complexity and its Applications* (2nd ed.). New York, Springer.

Longuet-Higgins, H. (1976). Perception of Melodies. *Nature* 263, 646-653.

Longuet-Higgins, H. and Lee, C. (1987). The Rhythmic Interpretation of Monophonic Music. *Mental Processes: Studies in Cognitive Science*, Cambridge, The MIT Press.

Manning, C. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, The MIT Press.

Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: the Penn Treebank, *Computational Linguistics* 19(2).

Marr, D. (1982). *Vision*. San Francisco, Freeman.

Martin, W., Church, K. & Patil, R. (1987). Preliminary Analysis of a Breadth-first Parsing Algorithm: Theoretical and Experimental Results. In Bolc, L. (Ed.), *Natural Language Parsing Systems*, Springer Verlag, Berlin.

Mumford, D. (1999). The dawning of the age of stochasticity. Based on a lecture at the Accademia Nazionale dei Lincei. (available at http://www.dam.brown.edu/people/mumford/Papers/Dawning.ps)

Osborne, M. (1999). Minimal description length-based induction of definite clause grammars for noun phrase identification. In *Proceedings EACL Workshop on Computational Natural Language Learning*. Bergen, Norway.

Palmer, S. (1977). Hierarchical Structure in Perceptual Representation. *Cognitive Psychology*, 9, 441-474.

Raphael, C. (1999). Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4), 360-370.

Restle, F. (1970). Theory of Serial Pattern Learning: Structural Trees. *Psychological Review,* 86, 1-24.

Rickard, T., Healy, A. & Bourne Jr., E. (1994). On the cognitive structure of basic arithmetic skills: Operation, order and symbol transfer effects. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 20, 1139-1153.

Rissanen, J. (1978). Modeling by the shortest data description. *Automatica*, 14, 465-471.

Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry*. Series in Computer Science - Volume 15. World Scientific, 1989.

Rosenbaum, D., Vaughan, J., Barnes, H. & Jorgensen, M. (1992). Time course of movement planning: Selection of handgrips for object manipulation. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 18, 1058-1073.

Saffran, J., Loman, M. & Robertson, R. (2000). Infant Memory for Musical Experiences. *Cognition,* 77, B16-23.

Scha, R., Bod, R. & Sima'an, K. (1999). Memory-Based Syntactic Analysis. *Journal of Experimental and Theoretical Artificial Intelligence,* 11(3), 409-440.

Schaffrath, H. (1995). The Essen Folksong Collection in the Humdrum Kern Format. D. Huron (ed.). Menlo Park, CA: Center for Computer Assisted Research in the Humanities.

Shannon, C. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal.* 27, 379-423, 623-656.

Sima'an, K. (1996). Computational Complexity of Probabilistic Disambiguation by means of Tree Grammars. In *Proceedings COLING-96*, Copenhagen, Denmark.

Simon, H. (1972). Complexity and the Representation of Patterned Sequences as Symbols. *Psychological Review*. 79, 369-382.

Snyder, B. (2000). *Music and Memory*. Cambridge, The MIT Press.

Spiro, N. (2002). Combining Grammar-based and Memory-based Models of Perception of Time Signature and Phase. In Anagnostopoulou, C., Ferrand, M. & Smaill, A. (Eds.). *Music and Artificial Intelligence*, Lecture Notes in Artificial Intelligence, Vol. 2445, Springer-Verlag, 186-197.

Temperley, D. (2001). *The Cognition of Basic Musical Structures*. Cambridge, The MIT Press.

Wertheimer, M. (1923). Untersuchungen zur Lehre von der Gestalt. *Psychologische Forschung* 4, 301-350.

Wundt, W. (1901). *Sprachgeschichte und Sprachpsychologie*. Engelmann, Leipzig.