The Data-Oriented Parsing Approach: Theory and Application

Rens Bod

School of Computer Science, University of St. Andrews, Scotland
rb@cs.st-and.ac.uk

1 Introduction

Parsing models have many applications in AI, ranging from natural language processing (NLP) and computational music analysis to logic programming and computational learning. Broadly conceived, a parsing model seeks to uncover the underlying structure of an input, that is, the various ways in which elements of the input combine to form phrases or constituents and how those phrases recursively combine to form a tree structure for the whole input. During the last fifteen years, a major shift has taken place from rule-based, deterministic parsing to corpus-based, probabilistic parsing. A quick glance over the NLP literature from the last ten years, for example, indicates that virtually all natural language parsing systems are currently probabilistic. The same development can be observed in (stochastic) logic programming and (statistical) relational learning. This trend towards probabilistic parsing is not surprising: the increasing availability of very large collections of text, music, images and the like allow for inducing statistically motivated parsing systems from actual data.

A corpus-based parsing approach that has been quite successful in various fields of AI, is known as Data-Oriented Parsing or DOP. DOP was originally developed as an NLP technique but has been generalized to music analysis, problem-solving and unsupervised structure learning [7, 8, 14, 81]. The distinctive feature of the DOP approach, when it was first presented, was to model sentence structures on the basis of previously observed frequencies of sentence-structure fragments, without imposing any constraints on the *size* of these fragments. Fragments include, for instance, subtrees of depth 1 (corresponding to context-free rules), as well as entire trees.

The DOP model was different from all other statistical parsing models at the time. Other models typically started off with a predefined grammar and used a corpus only for estimating the rule probabilities [5, 6, 27, 49, 80]. The DOP model, on the other hand, proposed not to train a predefined grammar on a corpus, but to directly use corpus fragments as a grammar. This approach has now gained wide usage, as exemplified by the work of [29, 30, 32, 36, 37, 55, 66 and many others].

The other innovation of DOP was to take all corpus fragments, of any size, rather than a small subset. During the last few years, we can observe a shift towards using more and larger corpus fragments with fewer restrictions in parsing models: while the models of [35] and [46] still restricted the fragments to the locality of head-words, later models showed the importance of including context from higher nodes in the tree [29, 60]. The importance of including nonhead-words has also become accepted [for example, 30, 37]. Moreover, [38] argues for "keeping track of counts of arbitrary fragments within parse trees", which has indeed been carried out in [39] and others, who use exactly the same set of (all) subtrees from a parsed corpus as already proposed in [7].

To date, one of the most robust empirical results in natural language parsing is that the parse accuracy increases if larger subtrees are included in the grammar [for instance, 11, 14, 40, 56, 84]. Although the use of all subtrees was for a long time deemed too costly, efficient algorithms have now been developed, ranging from compact Probabilistic Context-Free Grammar (PCFG) reductions of DOP [53] to tree kernels for all-subtrees models [40]. Consequently, the DOP model has been employed to boost a number of concrete applications, such as dialog processing [9], speech understanding [84] and machine translation [55, 57].

In the meantime, the DOP approach has been generalized to other modalities, including music analysis and problem solving. It has turned out that probabilistic corpus-based parsing outperforms deterministic rule-based processing not only for language but also for melodic analysis [12, 13] and problem solving [19, 20]. Our goal for this Chapter is therefore to present the DOP approach from a multi-modal perspective. But in order to do, it is convenient to first explain DOP for language processing, after which we discuss an integrated DOP model that unifies the different modalities. We will go into the various computational issues and show how the model can be tested against hand-annotated corpora. Finally, we will discuss shortcomings of this supervised approach, and present some results of recent work that extends DOP towards unsupervised learning.

2 A DOP Model for Language: Combining Likelihood and Simplicity

The main motivation behind DOP is to integrate rule-based and exemplar-based aspects of natural language. DOP is rule-based in that it proposes a generative system of productive units; it is exemplar-based in that its productive units are concrete fragments from representations of previous input. An example from language may illustrate the approach. Suppose that we start with a very small corpus of only two sentences with their phrase-structure trees that are labeled by traditional lexical-syntactic categories, as shown in Figure 1. Here NP = noun phrase, VP = verb phrase, and PP = prepositional phrase. (Note that actual corpora like the `Penn Treebank` contain tens of thousands of trees – see [74]).

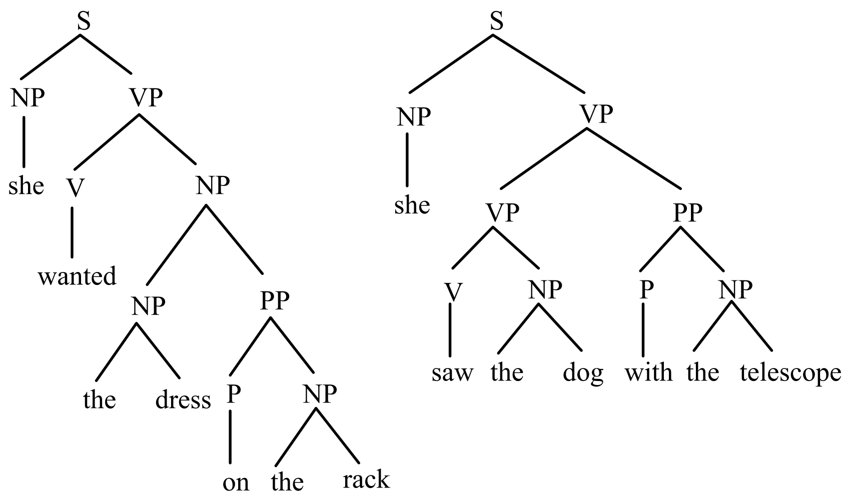

Fig. 1. A small training set of two tree structures

To dispel dogmatic slumbers it is good to realize that a corpus of annotated sentences need not be produced by means of a separate grammar or parser. Instead, most existing annotated corpora or 'treebanks' are created by human annotators who are only given an annotation guideline with some example analyses of sentences. One could claim that human annotators use an internalized grammar to annotate the sentences, but recent work by [63] and [17, 18] has shown that a contrasting position is just as viable: humans learn to understand and produce new sentences entirely in a statistical, item-based way (see [87] for a psycholinguistic motivation). We will go into the unsupervised learning of tree structures in Section 9, but for the moment we will start out from corpora that are already annotated.

Turning back to the corpus in Figure 1, a new sentence can be derived by combining subtrees from the trees in the corpus. The combination operation between subtrees used by DOP is called *label substitution*, indicated as '$\circ$'. The substitution operation identifies the leftmost nonterminal leaf node of one subtree with the root node of a second subtree – that is, the second subtree is substituted on the leftmost nonterminal leaf node of the first subtree, provided that their categories match. Starting out with the corpus of Figure 1, for instance, the sentence "She saw the dress with the telescope" may be derived as shown in Figure 2.

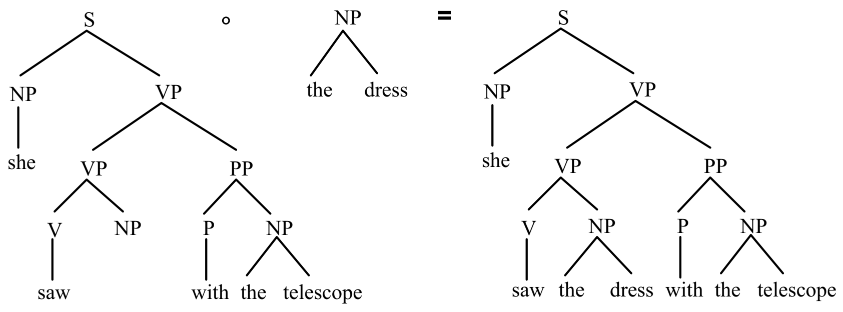

Fig. 2. Analyzing a new sentence by combining subtrees from Fig.1

In Figure 2, the sentence "She saw the dress with the telescope" is interpreted analogously to the corpus sentence "She saw the dog with the telescope": both sentences receive the same phrase structure where the prepositional phrase "with the telescope" is attached to "the VP saw the dress". We can also derive an alternative phrase structure for the test sentence, namely by combining three (rather than two) subtrees from Figure 1, as shown in Figure 3. We will write (t ∘ u) ∘ v as t ∘ u ∘ v with the convention that ∘ is left-associative. In Figure 3, the sentence "She saw the dress with the telescope" is analyzed

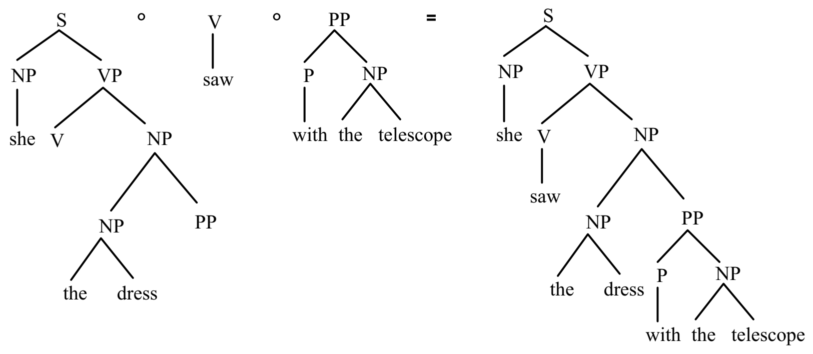

Fig. 3. A different derivation for "She saw the dress with the telescope"

in a different way where the PP with the telescope is attached to the NP the dress, corresponding to a different meaning than the tree in Figure 2. Thus the sentence is ambiguous in that it can be derived in (at least) two different ways which is analogous either to the first or second tree in Figure 1. Both analyses can in principle be perceived by humans although the first analysis in Figure 2 is generally seen as the most plausible one.

Note that subtrees can be of arbitrary size: they can range from just one categorized word to entire sentence-analyses. This allows DOP to take into acount both the rule-based nature of linguistic productivity and the exemplar-based nature of idiomatic phrases, multi-word units and other idiosyncracies in language [52]. Also note that an unlimited number of other sentences can be derived by combining subtrees from the corpus, such as "She saw the dress on the rack with the telescope" and "She saw the dress with the dog on the rack with the telescope", and so forth. Thus we can get infinite productivity from a finite corpus of representations (see [21] for further examples and how this argues against [34]). Most of these sentences are highly ambiguous: many different analyses can be assigned to each of them due to a combinatorial explosion of different prepositional-phrase attachments. Yet, most of these analyses are not plausible: they do not correspond to the structures humans come up with. The phenomenon that the same input can have many different structural organizations while only one of them tends to be perceived is known as the 'ambiguity problem'. This problem is one of the hardest problems in artificial intelligence and cognitive science. [34: 37] estimates that almost every sentence from the Wall Street Journal has many – often more than one million – possible phrase-structure trees!

How can we select from the possible trees of a sentence the 'best' tree as assigned by humans? DOP basically employs a statistical methodology: it computes the best tree from the relative frequencies of partial trees in a large corpus of previous trees (of sentences). Results from psycholinguistics indeed support the idea that the frequency of occurrence of a structure is a very important factor in language comprehension. In particular, frequencies play a key role in disambiguation and well-formedness judgments of sentences (refer to [62] or [72] for overview papers). [50] even argue that "knowledge of grammar includes knowledge of probabilities of syntactic structures".

Of course, the frequency of occurrence of a structure is not the only factor in syntactic disambiguation. Discourse context and semantics also play an important role. In [8, 9], we have shown how discourse and semantics can be incorporated into DOP if we have corpora containing discourse structure and semantic annotations. The other main disambiguation factor that we will go into here, and that appears to be essentially different from frequency, is the notion of 'simplicity' of a structure. It has often been claimed that there is strong preference in favor of the simplest analysis consisting of the smallest number of derivation steps (see [31, 48]). The preference for simplicity may be in competition with the preference for likelihood. Although some accounts propose that likelihood and simplicity are parts of the same coin [31], we showed in [12] that the two principles appear to play a rather distinct role in language processing.

If we indeed take the 'simplest' parse tree as the one that can be derived by the smallest number of steps, which in our case is the smallest number of subtrees, then the simplest analysis corresponds to the shortest derivation, which differs almost always from the most likely analysis [12]. An interesting property of the shortest derivation of a sentence is that it corresponds to the parse tree produced by the largest possible subtrees from the corpus. This means that the preference for simplicity can also be seen as a preference for maximizing the structural similarity or analogy between a sentence and the corpus, regardless of the frequencies of the subtrees. On the other hand, the principle of likelihood favors the parse tree that can be constructed out of the likeliest subtrees, by computing the total likelihood of the parse tree from the relative frequencies of the subtrees – regardless of the size of these subtrees.

In [12], we investigated a number of ways to balance these two principles. We came up with a model that selected the tree generated by the shortest derivation from among the top of the distribution of most likely trees for a certain input. A drawback of this approach is that by selecting the shortest derivation from the most likely trees, we first have to compute (the top of) the probability distribution of trees for an input string. While this is feasible for large corpora such as the linguistic `Penn Treebank` [74] or the musical `Essen Folksong Collection` [82], containing tens of thousands of analyses, it is not so for the much smaller corpora for deductive explanations or proof trees used in modeling problem-solving (see Sect.6). As a consequence, the probability distribution of trees for a new input is almost flat for such corpora, resulting in poor predictions for the 'best' tree. Therefore the model in [12] does not suffice as a general disambiguation technique.

In [16, 20], we proposed an alternative combination of simplicity and likelihood within the DOP framework which seems to be a better candidate for a modality-independent integration of the two principles. According to this combination, the best tree is the one that is generated by the shortest derivation consisting of the fewest subtrees, and in case the shortest derivation is not unique we select the most probable tree from among the shortest derivations. Since in almost all real-world situations the shortest derivation is indeed not unique, this approach *de facto* takes into account both simplicity and likelihood, and can still be described by an overall probabilistic model (see below).

We will illustrate this integration with the linguistic example given above. We start with the criterion of simplicity, in other words the shortest derivation. According to this criterion the tree structure in Figure 2 would be preferred because it can be generated by just two subtrees from the training set. Any other tree structure, such as in Figure 3, would need at least three subtrees from the training set in Figure 1. Note that the tree generated by the shortest derivation indeed tends to be structurally more similar to the corpus (that is, having a larger overlap with one of the corpus trees) than the tree generated by the longer derivation.

Had we restricted the subtrees to smaller sizes – for example to depth-1 subtrees, which makes DOP equivalent to a (stochastic) context-free grammar – the shortest derivation would not be able to distinguish between the two trees in Figures 2 and 3 as they would both be generated by 9 rewrite rules. The same is true if we used subtrees of maximal depth 2 or 3. It seems that only if we do not restrict the subtree depth can we take into account arbitrarily far-ranging dependencies and model new sentences as closely as possible on previous sentence-analyses by the shortest derivation. It is of course an empirical question as to how far the inclusion of large subtrees also leads to better predictions for the tree structure as assigned by humans (Section 8).

When the shortest derivation is not unique, our model selects from the remaining derivations the one that generates the most probable tree (as computed from the relative frequencies of the subtrees – see below). Why don't we do this the other way round – more specifically, why don't we first compute the most probable tree(s) of a sentence and next select the shortest derivation? We already mentioned above that such an approach obtains poor predictions of the best tree in domains with highly specific labels for which it is difficult to gather meaningful statistics. Instead, by first computing the shortest derivations we constrain the set of candidates for the best tree to those that are structurally most similar to trees in the corpus, on which next statistical computations are applied.

We refer to this instantiation of the DOP framework as DOP+ [20] which we will now define more formally. Let us first give the definition of an analysis tree of an input string (see Definition 1).

Definition 1 *Given a corpus C of trees T_1, T_2, ... , T_n, and a label substitution operation $\circ$, then an analysis tree of an input string W with respect to C is a tree T such that (i) there are subtrees t_1, t_2, ..., t_k in T_1, T_2, ... , T_n for which $t_1 \circ t_2 \circ ... \circ t_k = T$, (ii) the root of T is equal to the distinguished symbol S and (iii) the yield of T is equal to W.*

The tree generated by the shortest derivation T_{sd} according to DOP+ is given by Definition 2.

If T_{sd} is not unique, we select from among the shortest derivations the tree with highest probability. The probability of a tree is defined in terms of the probabilities of the derivations that generate it, which are in turn defined in terms of the probabilities of the subtrees these derivations consist of [8], as given by Definition 3.

Let r(t) return the root label of t. Then we may write:

Definition 2 *Let $L(d)$ be the length of derivation d in terms of its number of subtrees, that is if $d = t_1 \circ t_2 \cdots \circ t_k$ then $L(d) = k$. Let d_T be a derivation which results in tree T. Then T_{sd} is the tree which is produced by a derivation of minimal length: $T_{sd} = argmin_T L(d_T)$*

Definition 3 *The probability of a subtree t, P(t), is defined 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. (For practical purposes this simple relative frequency may be adjusted if the frontier of t contains unknown words – see Section 8).*

$$P(t) = \frac{\mid t \mid}{\sum_{t':r(t')=r(t)} \mid t' \mid} \tag{1}$$

Under the assumption that subtrees are stochastically independent, the probability of a derivation $t_1 \circ t_k$ is defined as the product of the probabilities of its subtrees t_i:

$$P(t_1 \circ ... \circ t_k) = \prod_i P(t_i) \tag{2}$$

There may be different derivations that generate the same analysis tree. Assuming that the derivations partition the space of tree occurrences, the probability of a tree T is defined as 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:

$$P(T) = \sum_d \prod_i P(t_{id}) \tag{3}$$

The best parse tree T_{best} maximizes the probability of T_{sd} given input string W:

$$T_{best} = argmax_{T_{sd}} P(T_{sd} \mid W) \tag{4}$$

It should be emphasized that the DOP+ model deals exclusively with tree structures. Several richer models, based on more sophisticated linguistic representations, have been proposed, ranging from (Lexical-Functional Grammar) LFG-annotated corpora consisting of trees enriched with attribute-value matrices [22] to (Head-driven Phrase Structure Grammar) HPSG-annotated databases consisting of feature structures [79] and TAG-based DOP models that allow for richer combination operations [58]. But since these richer DOP models all use trees as their backbone, it is convenient to base our exposition of DOP on the use of trees. Moreover, since all international benchmarks for NLP (and other modalities) currently consist of phrase-structure trees, we will stick to tree-based DOP models for the scope of this review and refer to [24] for a general overview of linguistic DOP models.

Yet, even for tree-based DOP models there exist many different versions. For example, DOP+ uses a simple relative frequency estimator for assigning weights to the subtrees. While this estimator obtains good results on the `Penn Treebank`, it does not maximize the likelihood of the training set (see [61]). In [18], a DOP model was developed which does maximize the likelihood of the corpus by using the relative frequency estimator only as an initial parameter which is next re-estimated by the well-known Expectation-Maximization (EM) algorithm [43]. This model, called ML-DOP, uses cross-validation to avoid overtraining. While ML-DOP does not improve over previous DOP models, the maximum likelihood estimator in ML-DOP does boost unsupervised versions of DOP. We will go into these unsupervised extensions in Section 9. For an overview of the various statistical estimators used in DOP and many probabilistic grammars, we refer the reader to [95] or [96].

3 A DOP Model for Music

It is rather straightforward to apply the DOP approach to melodic analysis of musical pieces. As in natural language, a listener segments a sequence of notes into groups or phrases that form a grouping structure for the whole piece [69]. For example, according to [68: 37], a listener hears the grouping structure in Figure 4 for the first few bars of melody in the Mozart G Minor Symphony, K. 550.

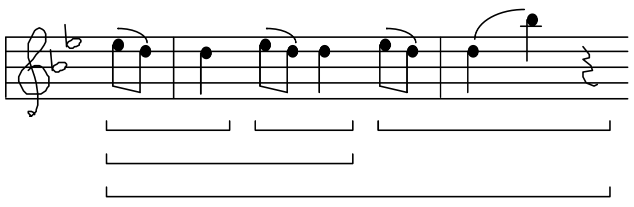

Fig. 4. Grouping structure for the opening theme of Mozart's G minor symphony

Each group is represented by a slur beneath the musical notation. A slur enclosed within a slur means that a group is heard as part of a larger group. This hierarchical structure of melody can, without loss of generality, also be represented by a phrase structure tree, as illustrated in Figure 5.

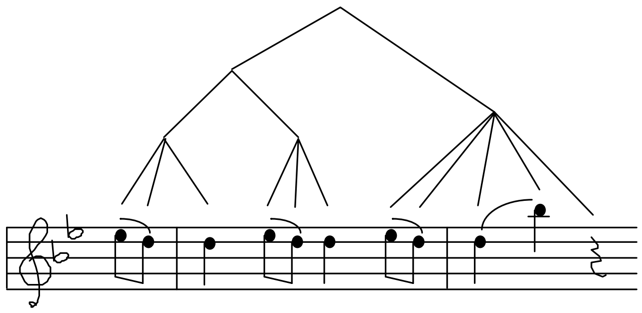

Fig. 5. Tree structure for the grouping structure in Fig.4

Note the analogy with phrase structure trees in linguistics: a tree describes how parts of the input combine into constituents, how these constituents combine into larger constituents, and so on into a representation for the whole input. Apart from this analogy, there is also a difference: while the nodes in a linguistic tree structure are typically labeled with syntactic categories such as S, NP, VP and the like, musical tree structures are usually unlabeled. This is because in language there are syntactic constraints on how words can be combined into larger constituents (for instance, in English a determiner can be combined with a noun only if it precedes that noun), while in music there are no such restrictions: in principle any note may be combined with any other note. This makes the problem of ambiguity in music much harder than in language. [70] note that "Any given sequence of note values is in principle infinitely ambiguous, but this ambiguity is seldom apparent to the listener." For example, the first few bars of Mozart's G Minor Symphony could also be assigned the alternative grouping structure in Figure 6 (among the many other possible structures).

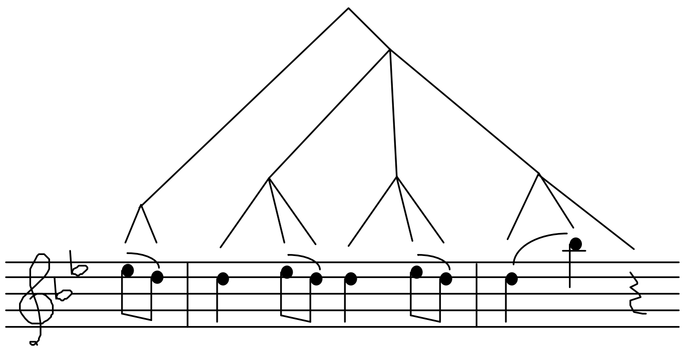

Fig. 6. Alternative grouping structure for Mozart's opening theme

While this alternative structure is possible in that it can be perceived, it does not correspond to the structure that is actually perceived by a human listener. As in natural language, there is thus an important question as to how to select the perceived tree structure from the set of possible tree structures of a musical input. Many systems attempt to disambiguate melodic structure in an entirely rule-based way. For example, [68] and [86] use preference rules that describe Gestalt-perceptions of the kind identified by [91]. However, similar to language, there are extremely many 'multi-note units' and other idiomatic phrases and musical clichés that melodic analysis has to deal with (see [12, 13]). Most parsing approaches to melodic analysis are nowadays probabilistic and exemplar-based (for instance,[12, 41, 47]). These approaches are trained on corpora such as the `Essen Folksong Collection` (EFC) which contains musical trees of over 6,000 folk songs [82].

Since the encoding of note sequences is not as straightforward as in natural language, let us briefly explain how the folk songs in the EFC are annotated (see [82] for the full annotation scheme). The Essen folk songs are represented by the so-called Essen Associative Code (ESAC). The pitch encodings in ESAC resemble 'solfege': scale degree numbers are used to replace the movable syllables 'do', 're', 'mi', and so forth. Thus 1 corresponds to 'do', 2 corresponds to 're', and so on. 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 respectively above or below the principle octave (thus -1, 1 and +1 all refer 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. 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 annotated by hard returns in ESAC, which we automatically convert into bracket representations where '(' indicates the start and ')' the end of a phrase. These phrase boundaries were manually assigned on the basis of the pitch encodings only (the lyrics were not taken into account – see [82]). The phrases in the EFC are unlabeled.

However, to use the DOP approach for parsing the EFC, we first need to (automatically) add three basic labels to the phrase structures: S for the whole song, P for each phrase and N for each note. In this way, we obtain conventional tree structures that can directly be used by DOP+. To illustrate this, consider the simple corpus in Figure 7 of two musical tree structures.

This corpus contains two very simple melodies, the first consisting of two phrases, (1 2) (2 3), and the second consisting of only one phrase, (1 2 3 1). If we take this corpus as our training set, then a new melody, such as '1 2 1 2', can be parsed by DOP+ by combining subtrees from this corpus, again by means of the substitution operation $\circ$.

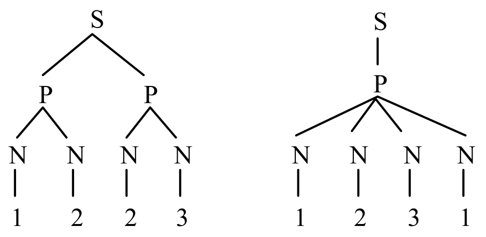

Fig. 7. A simple musical corpus

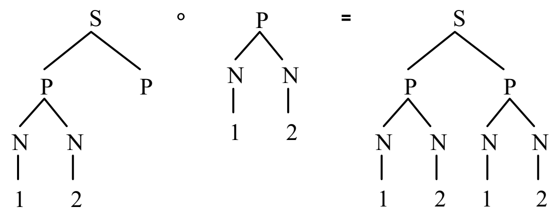

Fig. 8. Parsing a new melody by combining subtrees from Fig.6

But this melody can also be parsed in a different way, resulting in a different parse tree, for example by combining the following subtrees from Figure 7.

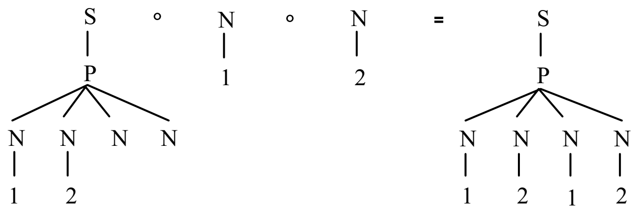

Fig. 9. Generating a different tree structure for the same melody

Remember that this free combination of subtrees only defines the set of possible structures of an input. To predict the best musical tree structure as assigned by humans, DOP+ selects the resulting tree structure in Figure 8, as it corresponds to the shortest derivation, and thereby recognizes that the phrase (1 2) can be parsed as one (previously observed) pattern. Had we restricted the subtrees to the smallest ones (which would lead to a probabilistic context-free grammar), the resulting tree in Figure 9 would have corresponded to the shortest derivation since it only consists of 6 rules (or depth-1 subtrees), while the tree in Figure 8 would have needed 7 rules. Thus large subtrees are important, and should not be restricted if we want to maximize similarity. In case there is more than one shortest derivation, the most probable tree is selected, just as with language (Definition 3).

Of course this musical example is exceedingly simple. The `Essen Folksong Collection` provides a much more challenging set of melodies, and will be used for our experiments in Sect.8. For an in-depth discussion of the variety and complexity of the melodies in the `Essen Folksong Collection`, see [59].

4 A DOP Model for Problem Solving in Physics

What counts for syntactic and melodic analysis also counts for problem solving and reasoning: given a problem or theorem, there can be (extremely) many different possible solutions or derivations. As in language and music, a major challenge is to select the 'best' derivation among the many possibilities. As a case study, we will concentrate on derivations for physics problems.

Let us first discuss what problem solutions or 'derivational explanations' in physics look like. Physics textbooks provide many examples of problem solutions which are typically used to solve new problems. Although textbook problem solutions may not reflect scientific practice, they are part of the training of every scientist and highly influence their reasoning. We will start with a simple, idealized example. In their Physics textbook, Alonso and Finn derive the Earth's mass from the Earth-Moon system as follows [2: 247]:

> Suppose that a satellite of mass m describes, with a period P, a circular orbit of radius r around a planet of mass M. The force of attraction between the planet and the satellite is $F = \frac{GMm}{r^2}$. This force must be equal to m times the centripetal acceleration $v^2/r = 4\pi^2 r/P^2$ of the satellite. Thus,
>
> $$4\pi^2 mr/P^2 = GMm/r^2$$
>
> Canceling the common factor m and solving for M gives
>
> $$M = 4\pi^2 r^3/GP^2$$

This rather textual derivation can be represented by means of the proof tree or derivation tree of Figure 10. Proof trees are widely used data structures in automated reasoning and theorem proving [3] and form the main representations in Explanation-Based Learning [76], Inductive Logic Programming [42] and Statistical Relational Learning [77].

Thus the derivation tree in Figure 10 represents the various steps from general laws to an equation for the mass M. In general, a derivation tree is a labeled tree where each node is annotated with a formula (the boxes are only convenient representations that have no additional meaning). The formulas at the top of each 'vee' (in other words, each pair of connected branches) in the tree can be viewed as premises, and the formula at the bottom of each 'vee' can be viewed as a conclusion which is arrived at by simple term substitution. The last derivation step in the tree is not formed by a 'vee' but consists of a

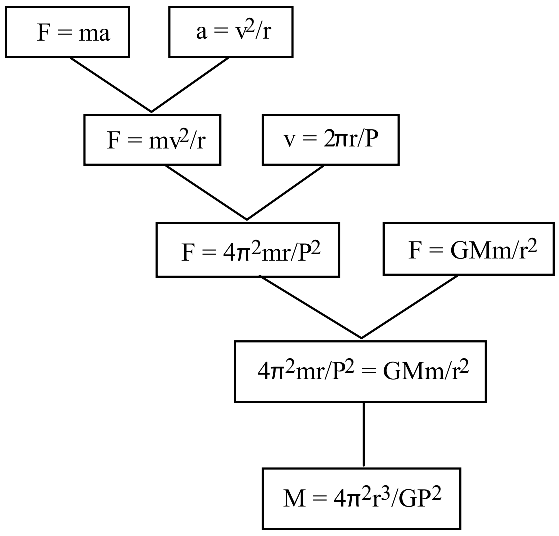

Fig. 10. Derivation tree for the derivation of the Earth's mass

unary branch which solves the directly preceding formula for a certain variable (in the tree above, for the mass M). Unary branches may also appear at other places in a derivation tree. In general, a unary branch refers to a mathematical derivation step, while a binary branch refers to a combination of laws (or conditions) by means of simple term substitution. Note that derivation trees are conventionally represented in an 'upside-down' manner with respect to phrase-structure trees in language and music (see [3]).

Suppose that the derivation tree in Figure 10 constitutes our training corpus, then the regularity known as Kepler's third law, which states that r^3/P^2 is constant, can be easily derived by using the subtree in Figure 11 which is extracted from the tree in Figure 10. The root of this subtree only needs to be solved for r^3/P^2, which is accomplished by a mathematical derivational step added in Figure 12.

We can thus already note a difference between derivations in language and music and derivations in physics: for the latter we need an additional mathematical component that can solve equations.

Of course, it is not the typical case that we can derive a new phenomenon by just one subtree. Often we need to combine several smaller subtrees, as is the case for instance in deriving the velocity of a satellite at a certain distance from a planet. This is accomplished by using the following two subtrees in Figure 13 from the tree in Figure 10, that are first combined by term substitution (represented by the operation '$\circ$')[1] and then solved for the velocity v.

[1] As long as no confusion arises we will use the same symbol for label substitution in language and music and term substitution in derivational problem solving.

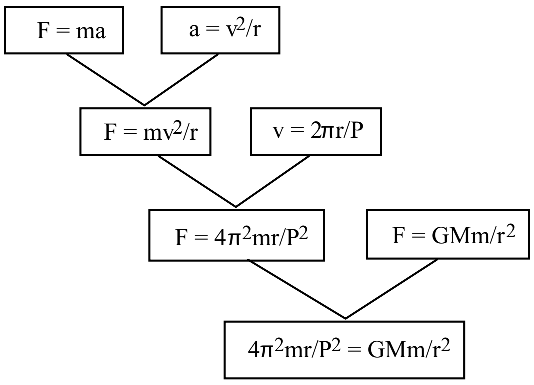

Fig. 11. A subtree from the tree in Fig.10

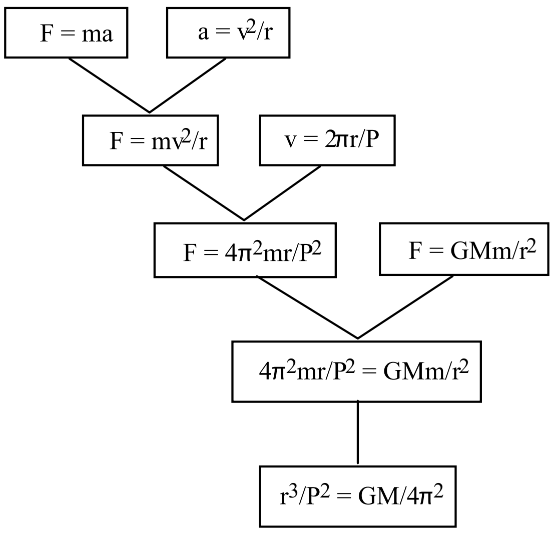

Fig. 12. Deriving Kepler's Law by the subtree in Fig.11

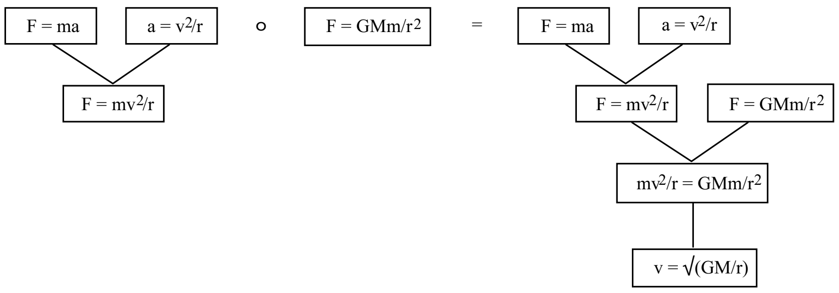

Fig. 13. Deriving the velocity of a satellite by combining two subtrees from Fig.10

Note the analogy with linguistic and musical processing: new input can be derived by combining subtrees from previously derived input, where the subtrees may be of any size – from single laws to entire derivation trees. But there are also some differences. Apart from the additional mathematical component, there is a difference in combining subtrees: while in music and language the 'combination operation' between subtrees consists of simple (leftmost) label substitution, the term 'substitution operation' in problem solving also expands the tree with a new root node (see Figure 13). We will therefore specify this operation more explicitly as follows:

The term 'substitution operation $\circ$' is a partial function on pairs of labeled trees, and its range is the set of labeled trees. The combination of tree t and tree u, written as $t \circ u$, is defined iff the equation at the root node of u can be substituted in the equation at the root node of t (that is, iff the lefthand side of the equation at the root node of u literally appears in the equation at the root node of t). If $t \circ u$ is defined, it yields a tree that expands the root nodes of copies of t and u to a new root node where the righthand side of the equation at the root node of u is substituted in the equation at the root node of t. Note that the substitution operation can be iteratively applied to a sequence of trees, with the convention that $\circ$ is left-associative.

The idea that new problems and phenomena can be solved by reusing parts of previous problem solutions is reminiscent of the notion of 'exemplar' in Thomas Kuhn's account of normal science. According to [67], exemplars are "problem solutions that students encounter from the start of their scientific education", and "Scientists solve puzzles by modeling them on previous puzzle-solutions" [67: 187-190]. In the following, we will use the terms 'exemplary problem solution' and 'exemplar' interchangeably. Our approach to problem solving is also congenial to Case-Based Reasoning (for example, [28]), where new problems are solved by modeling previous problems.

As noted above, there is a problem with derivational reasoning and problem solving which is analogous to linguistic and musical analysis: ambiguity. To illustrate this, it is convenient to enlarge our training corpus in Figure 10 with one other example from Alonso and Finn's textbook. This example again provides an exemplary problem solution for the Earth's mass but this time using an alternative derivation (2: 246). Both solutions are used as exemplars on which other problems are modeled. This second exemplar computes the Earth's mass from the acceleration of an object near the Earth's surface and which, following the derivation steps in [2], can be represented by the derivation tree in Figure 14 (where for the sake of conciseness the initial conditions $a = g$ and $r = R$ are represented by one label).

By substituting the values for g (the acceleration at the Earth's surface), R (the Earth's radius) and G (the gravitational constant), [2] obtain roughly the same value for the Earth's mass as in the previous derivation in Figure 10. They argue that this agreement is 'a proof of the consistency of the theory'

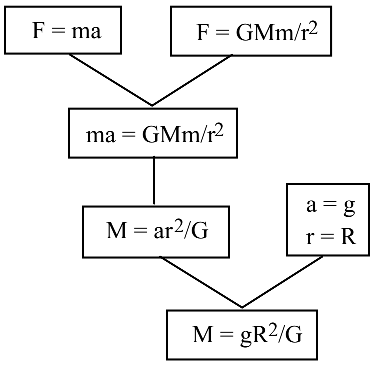

Fig. 14. An additional exemplar in the training corpus

[2: 247]. When we add this exemplar to our corpus, we get many different derivations for new phenomena or problems. For example, Kepler's regularity can now also be derived by the following alternative derivation in Figure 15, which uses a large subtree from Figure 13 in combination with two small subtrees from Figure 10.

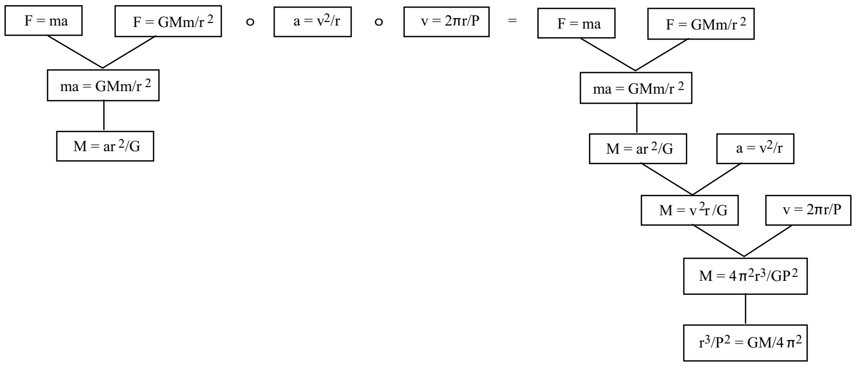

Fig. 15. An alternative derivation of Kepler's regularity

There is nothing wrong with this alternative derivation: there are no spurious non-explanatory laws that are irrelevant to this derivation (as would be Hooke's or Boyle's law). The only difference is that the derivation in Figure 15 is modeled on a different exemplar (that is, on a planet-particle model at rest) than the derivation in Figure 12 (which is modeled on an orbiting planet-satellite exemplar). In fact, the alternative derivation in Figure 15 is insightful as it expresses the conceptual equivalence between terrestrial and celestial mechanics in Newtonian dynamics. However, problem-solving experiments with physics students show that humans don't come up with this alternative derivation (Section 6). Unfortunately, the ambiguity problem is much worse: by combining subtrees from the two exemplars in Figures 10 and 14 in different ways, we get a combinatorial explosion of possible derivation trees of Kepler's law.

5 Towards a Unifying Approach

As with linguistic and musical analysis, also for problem-solving we hypothesize that the best tree is the one that can be constructed by the shortest derivation and that in case there are more shortest derivations the most probable tree should be selected among them. For example, the derivation tree of Kepler's regularity in Figure 12 corresponds to the shortest derivation since it can be constructed by just one large subtree from an exemplar (modulo the mathematical derivation step), while the derivation tree for the same regularity in Figure 15 needs at least three subtrees to be constructed. As a consequence, the tree in Figure 12 is more structurally similar to an exemplar in the training corpus than is the tree in Figure 15.

However, the use of the shortest derivation alone is not enough. It may occur that a phenomenon can be derived by two or more shortest derivations containing the same number of subtrees (and even the same number of labels) but resulting in different derivation trees. In such a case we also take into account the relative frequencies of the subtrees in a representative corpus of exemplars, and compute the most probable tree from among the trees generated by the shortest derivations. A higher subtree frequency expresses a wider usability of the derivational pattern for deriving phenomena. (We already argued in Section 4 why the reverse order of first computing the most probable tree, and next selecting the shortest derivation, is problematic for sparse corpora with highly specific labels like problem solving. We will support this argument with computational experiments in Section 8.)

It should be noted that previous statistical approaches to reasoning and problem solving were mainly based on stochastic enrichments of context-free grammars (CFGs) or definite-clause grammars (DCUs). These approaches, known as Stochastic Logic Programs or Statistical Relational Learning [42, 44, 78] cannot cover all possible dependencies in a derivation tree. We have shown in [19] that, as with language and music, there may be arbitrarily distant dependencies in problem solving, both structurally and sequentially. The examples we discussed in [19] came from the field of fluid mechanics. But distant dependencies already occur in derivations for much simpler systems, such as Galileo's pendulum (between leaf nodes and formulas later in the derivation tree). Entire subtrees must be preserved, otherwise they lose the particular dependency. It is well-known that students of physics typically have to go through various example-derivations before they can successfully solve new problems by themselves, usually by modeling the new problem on similar, previously solved problems [51, 67].

Thus it appears that DOP+ can also be employed for (equational) reasoning and problem solving. We only need to slightly modify the DOP+ definitions in Section 4. That is, we need to change 'label substitution' into 'term substitution' (which we already defined in Section 4). Second, the root node of a derivation tree must correspond to a mathematical description of a phenomenon, and the leaf nodes must be laws, conditions or any knowledge that cannot be derived from other equations (such as empirical corrections and normalizations). This brings us to Definition 4 for a DOP+ model of problem solving and reasoning.

Definition 4 *Given a corpus C of trees $T_1, T_2, ..., T_n$ representing derivations of phenomena, and a term substitution operation $\circ$, a derivation tree of a phenomenon P with respect to C is a tree T such that (i) there are subtrees $t_1, t_2, ..., t_k$ in $T_1, T_2, ..., T_n$ for which $t_1 \circ t_2 \circ ... \circ t_k = T$, (ii) the root of T is mathematically equivalent to P and (iii) the yield of T consists of either laws or antecedent conditions or any other equations that cannot be derived from higher-level equations.*

Definitions 2 and 3 in Section 4 – for the tree generated by the shortest derivation T_{sd} and the best tree T_{best}, respectively – remain the same (provided that we substitute the word string W by the phenomenon P in the definition of T_{best}). Given this commonality between problem solving and perceptual (linguistic and musical) processing, DOP+ may be a viable candidate for a general model of these modalities.

We can also try to further integrate Definitions 1 and 4 by referring to trees of natural phenomena, word strings and musical pieces as 'exemplars', and by referring to analysis trees and derivation trees as 'DOP+ generated trees'. But we then need to abstract from the differences between label substitution in language and music and term substitution in problem solving, for example, by viewing label substitution as a special case of term substitution (in case the entire substitutable labels are exactly equivalent no node expansion is created). This results in Definition 5, where we use a generalized notion of 'substitution operation' which is left unspecified.

Thus while there remain differences between problem solving in physics on the one hand and syntactic and melodic analysis on the other hand, there appears be a common level of representation and computation. This is the level of tree structures (the common representation) that are decomposed and recomposed to analyze new input in the shortest and most probable way (the common computation). The labeling of the trees and details of the combination operation differ across the modalities, but the formula for the best tree of an input I is the same for all modalities: $T_{best} = argmax_{T_{sd}} P(T_{sd} \mid I)$.

Definition 5 *Given a corpus C of trees $T_1, T_2, ..., T_n$ representing exemplars and a substitution operation $\circ$, a DOP+ generated tree with respect to C is a tree T such that there are subtrees $t_1, t_2, ..., t_k$ in $T_1, T_2, ..., T_n$ for which $t_1 \circ t_2 \circ ... \circ t_k = T$. T is said to be a derivation tree of a phenomenon P iff the root of T is mathematically equivalent to P and the yield of T cannot be further derived. T is said to be an analysis tree of an input string W iff the root of T is equal to the distinguished symbol S and the yield of T is equal to W.*

6 Test Corpora for DOP+

While annotated corpora are widely available for language [1, 74] and music [82], corpora of tree structures for physics problems are still very rare. Previous work that deals with students' problem solutions does not formalize these solutions by means of trees [71, 88]. In [16, 20] we therefore developed a corpus of physics problems whose solutions were directly converted to tree structures by students. The following briefly describes the construction of this 'physics corpus'.

A total of 19 third-year physics students from the University of Amsterdam (academic year 2005-2006) were paid to construct both a test corpus and a training corpus. 10 students were involved in constructing the test corpus while the remaining 9 students constructed the training corpus. As to the test corpus, the 10 students were asked to solve 14 elementary problems from classical mechanics and 10 elementary problems from fluid mechanics. The students had previously followed courses in classical mechanics and more recently an extensive course in fluid mechanics. The 24 problems given to them consisted in deriving a phenomenon from law and initial conditions. Four of these problems are given below (for more details, see [20]):

Problem nr. 1
Show that the period of the Earth's rotation for which an object at the equator would become weightless is given by $P = 2\pi\sqrt{(R/g)}$, where R is the Earth's radius and g is the gravitational acceleration at the Earth's surface.

Problem nr. 2
Show that the theoretical velocity which an object attains in free fall from height h is given by $v = \sqrt{(2gh)}$ where g is the gravitational acceleration at the Earth's surface.

$\vdots$

Problem nr. 23
When water flows through a right-angled V-notch, show that the discharge is given by $Q = KH^{5/2}$ in which K is a constant and H is the height of the surface of the water above the bottom of the notch.

Problem nr. 24
Show that the theoretical rate of flow through a rectangular notch is given by $Q = (2/3)B\sqrt{(2g)}H^{3/2}$, where B is the width of the notch and H is the height of the water level above the bottom of the notch.

After the students had solved the problems on paper, they were given a short, ten-minute tutorial on the concept of derivation tree, especially on the difference between binary branches in a tree (used for combining laws, conditions and similar), and unary branches (used for mathematical derivation steps of which the exact operations could be left implicit). The students were told that the exact order of the laws in a tree was not important as long as these laws could be properly combined by term substitution to solve the problem. After this brief tutorial, the students were asked to draw derivation trees for their problem solutions.

There was a high agreement among the derivation trees constructed by the students: on average 95.4% (SD=1.5) of the derivation trees per problem matched (modulo law order). In creating a gold standard, only the most voted tree was put in the corpus. In our case, the 24 most voted (that is, most frequently created) derivation trees for each problem constituted the test corpus.

As to the construction of the training corpus, the remaining 9 students were asked – after the same brief tutorial – to draw derivation trees for 33 problem solutions from classical and fluid mechanics that are used as exemplars in the textbooks by [2: Chaps.9-11] and [45: Chap.7]. The three examples in Figures 9, 12 and 13 were among these exemplary solutions. The agreement among the constructed derivation trees for the exemplary solutions was very high: 98.0% (SD=0.6). The most voted tree for each exemplary solution was put in the training corpus.

All 24 test problems could be solved by subtrees from the training corpus of 33 exemplars but this fact was not told to any of the students. Our total corpus of problem solutions thus consists of 57 trees including a training set of only 33 exemplars. This stands in strong contrast with the considerably larger linguistic and musical corpora. However, a representative corpus for elementary classical and fluid mechanics is necessarily much smaller than a representative corpus for language or music. It has for example been estimated by [51: 88] that the number of exemplars available to a physics expert lies around a few hundred. For undergraduates, who have only knowledge of elementary physics, this number is of course much lower, and the 33 exemplary problem solutions from [2] and [45] cover the typical exemplars from mechanics learned by undergraduate physics students (in other words, the planet-satellite model, the frictionless plane, the harmonic oscillator, the vena contracta, and the like). Thus our physics corpus is likely to correspond to the full set of exemplars learned by physics students in their curriculum.

7 Computing T_{best}

Before we can test DOP+ on the various corpora, we need to go into the problem of computing T_{best} for a given input. The computation of T_{best} is especially challenging for language and music where our corpora contain thousands of trees which correspond to millions of subtrees. We will therefore first go into linguistic and musical parsing and next come back to problem solving. The way DOP+ combines subtrees into new trees is formally equivalent to a Tree-Substitution Grammar or TSG [8]. Moreover, the way DOP+ defines the best tree is covered by the notion of a Stochastic TSG or STSG. There are standard algorithms that compute the tree structures (a packed parse forest) of an input string given an STSG. These algorithms run in Gn^3 time, where G is the size of the grammar (the number of subtrees) and n is the length of the input string (the number of words or notes).

Existing parsing algorithms for context-free grammars or CFGs, such as the CKY algorithm [94], can be easily extended to TSGs by converting each subtree t into a context-free rewrite rule where the *root* of t is rewritten by its yield: $root(t) \rightarrow yield(t)$. Indices are used to link each rule to its original subtree. Next, T_{best} can be computed by a best-first beam search technique known as Viterbi optimization [73]. However, the direct application of these techniques to DOP(+) is infeasible mainly because the number of subtrees usually grows exponentially with the corpus size [83]. The relatively small `Air Travel Information System` (ATIS) corpus of 750 trees [74] contains over 40,000 subtrees, and the `Wall Street Journal` (WSJ) corpus of 50,000 trees contains more than 100 million subtrees.

To make parsing with DOP feasible, several heuristics have been proposed, ranging from randomly sampling subtrees [8] to restricting the subtrees on linguistic grounds [84]. DOP's ideal to parse with all, arbitrarily large subtrees might have died an early death as being computationally prohibitive, if it were not for an insight by [53, 54] that the unwieldy DOP grammar can be reduced to a set of eight Probabilistic Context-Free Grammars (PCFGs) which is linear rather than exponential in the number of nodes in the corpus[2]. Goodman's PCFG reduction was initially developed for the probabilistic version of DOP but it can also be applied to computing the shortest derivation. The following briefly summarizes the method.

[2] [40] have used kernel methods to develop an efficient parsing algorithm for an all-subtrees representation.

The key idea is to re-label the nodes in the corpus trees. Every node in every tree is assigned a unique number which is called its 'address'. The notation $A@k$ denotes the node at address k where A is the nonterminal labeling of that node. A new nonterminal is created for each node in the training data. This nonterminal is called A_k. Nonterminals of this form are called 'interior', while the original nonterminals in the parse trees are called 'exterior'. Let a_j represent the number of subtrees headed by the node $A@j$. Let a represent the number of subtrees headed by nodes with nonterminal A, that is $a = \sum_j a_j$.

Goodman shows that there is a PCFG with the following property: for every subtree in the training corpus headed by A, the grammar will generate an isomorphic subderivation with probability $1/a$ [53]. The construction is as follows. For a node $(A@j(B@k, C@l))$, the following eight PCFG rules are generated, where the number in parentheses following a rule is its probability:

$$\begin{array}{ccccccc} A_j & \rightarrow & BC(1/a_j) & \quad A & \rightarrow & BC(1/a) \\ A_j & \rightarrow & B_kC(b_k/a_j) & \quad A & \rightarrow & B_kC(b_k/a) \\ A_j & \rightarrow & BC_l(c_l/a_j) & \quad A & \rightarrow & BC_l(c_l/a) \\ A_j & \rightarrow & B_kC_l(b_kc_l/a_j) & \quad A & \rightarrow & B_kC_l(b_kc_l/a) \end{array} \tag{5}$$

Goodman next shows by simple induction that subderivations headed by A with external nonterminals at the roots and leaves; internal nonterminals elsewhere have probability $1/a$ [53]. Further, subderivations headed by A_j with external nonterminals only at the leaves, and internal nonterminals elsewhere, have probability $1/a_j$. This can be easily demonstrated by multiplying the relevant probabilities of the rules, which brings Goodman to his main theorem, that his construction produces PCFG derivations isomorphic to DOP derivations with equal probability [53: 130-133].

Note that the PCFG reduction can also be used to compute the shortest derivation, since the most probable derivation is equal to the shortest derivation if each subtree is given equal probability. This can be seen as follows. Suppose we give each subtree a probability p, then the probability of a derivation involving n subtrees is equal to p^n, and since $0 < p < 1$, the derivation with the fewest subtrees has the greatest probability. For all our experiments with linguistic and musical corpora, we employ Goodman's reduction in combination with a best-first CKY parsing algorithm [94] that computes the most probable parse tree from among the shortest derivations.

Let us now turn to computing T_{best} for problem solving. In practice the computation of T_{best} for a mathematical description of a phenomenon is less hard, simply because a corpus of exemplary problem solutions tends to be much smaller than corpora for language and music (as discussed in the previous section). The training set of our problem solving corpus contains only 33 trees, which correspond to 408 different subtrees. Although this number of subtrees is computationally tractable, the root of each subtree may be extended with mathematical derivation steps at any point in a derivation (as we have seen in Sect.4). Thus there can be no *a priori* reduction of a problem-solving corpus into a compact PCFG, because we do not know beforehand which mathematical operations are needed at the subtree-roots.

In principle we could generate all possible mathematical extensions for all subtrees (in other words, all solutions for possible variables in the equations at the subtree-roots). But this would lead to a combinatorial explosion of possible subtree extensions. Fortunately, there are standard equational reasoning systems that can efficiently solve an equation given a set of other equations, such as `TK Solver` (http://www.uts.com/). In our experiments below, we first convert each derivation tree from the training corpus into its subtrees. Next, we extract the equations from the subtree-roots, which are indexed to remember the subtrees they are extracted from. This results in a list of 408 equations. For each test problem (namely, the equation to be solved), we use `TK Solver` to derive a set of solutions given the list of 408 equations. It turns out that virtually all problems receive more than 60 different solutions, even *after* abstracting from the order of the equations used in the solution, which gives an idea of the ambiguity if we do not have any mechanism to break ties.

From the output of `TK Solver`we select the shortest solution(s) for each problem that use(s) the fewest equations. Next, the equations of the shortest solution(s) are converted back to their corresponding subtrees, which are combined into the tree corresponding to the shortest derivation, T_{sd}. In case T_{sd} is not unique we compute the probability for each T_{sd} and select the most probable tree, which yields T_{best}.

8 Experiments with DOP+

For language, we used the now standard division of the `Wall Street Journal` (WSJ) corpus in the `Penn Treebank`, of which Sections 2 through 21 are used as training set (approximately 40,000 sentences), and of which Section 23 is used as test set (2,416 sentences $\leq$ 100 words). As in other experiments with the WSJ, all trees were stripped of their semantic tags, co-reference information and quotation marks (see [73]). In case a word from a test sentence was unknown in the training set, we employed the unknown word model in [11], based on statistics on word-endings, hyphenation and capitalization. For music, we used the same (random) division of the `Essen Folksong Collection` (EFC) as in [13] into a training set of 5,251 trees and a test set of 1,000 trees. There were no unknown notes for this division. As explained in Section 3, the root of each EFC tree was labeled with the distinguished symbol 'S', the notes were labeled with 'N' and the internal nodes with 'P'. For problem solving, we used the training set of 33 exemplary problem solutions and the test set of 24 problems, as described in Section 7.

The training set trees for each modality are used to extract the subtrees employed by DOP+, while the test data without the trees are used as input. The best trees predicted by DOP+ are compared with the trees in the respective test sets. The degree to which these best trees match the test set trees is a measure for the accuracy of the system. An evaluation metric which has become standard in the field of NLP, and which is also used in the field of music analysis, is the `PARSEVAL` metric of precision and recall [4]. This metric compares a so-called 'proposed' parse tree P (that is, our T_{best}) with the corresponding correct test set parse tree T as follows:

$$Precision = \frac{\#\ correct\ constituents\ in\ P}{\#\ constituents\ in\ P} \tag{6}$$

$$Recall = \frac{\#\ correct\ constituents\ in\ P}{\#\ constituents\ in\ T} \tag{7}$$

A constituent in P is said to be 'correct' if there exists a constituent in T of the same label that spans the same sequence of leaves. Since precision and recall can obtain rather different results (see [13]), they are typically balanced by a single measure of performance, known as the F-score:

$$Fscore = \frac{2\ \times\ Precision\ \times\ Recall}{Precision\ \times\ Recall} \tag{8}$$

We will use these definitions for evaluating T_{best} in language and music. However, they cannot be directly applied to evaluating T_{best} in problem solving. This is because the exact sequence of leaves is irrelevant here. While in language and music the sequence of leaves of a tree constitutes respectively the sentence and the musical piece, the leaves of a problem-solving tree constitute the laws and conditions in a derivation. Also, it does not matter whether we put a law as a premise at a left daughter node or at a right daughter node, as long as their combination results in the same conclusion (which we also explained to the students in creating the problem solving corpus – see Section 6). Thus we can only reasonably apply the metrics above to problem solving if we substitute 'same sequence of leaves' by 'same leaves' in the definition of 'correct' constituent above, such that we abstract from the order of the leaves.

It is one of the most essential features of DOP+ that arbitrarily large subtrees are taken into consideration. To test the usefulness of this feature, we performed a number of experiments where we restricted the training set subtrees to a certain maximum size. We define the size of a subtree by its depth, which is the length of a subtree's longest path from root to leaf. In this way we can test a range of other models; for example, by restricting the maximum depth of the subtrees to one, DOP+ is equivalent to a stochastic context-free grammar. As pointed out in [54: 134], Goodman's reduction method can still be applied to DOP when the training set subtrees are constrained to a certain depth. The following Table shows the results of our experiments where we give for increasing subtree depths the F-scores (in percentages) for respectively language (the WSJ corpus), music (the EFC corpus) and problem solving (the problem-solving corpus in Section 6). The maximum tree depth in the `Essen Folksong Collection` is 3, while the maximum tree depth in the problem solving corpus is 6.

Table 1. F-scores of DOP+ for different subtree depths

Max. Subtree Depth	Language *(Wall St. Journal)*	Music *(Essen Folksongs)*	Problem-Solving *(Physics Corpus)*
1	68.5	58.4	45.8
2	77.2	77.3	62.5
3	80.9	88.9	75.0
4	82.1		83.3
6	86.0		87.5
8	88.2		
10	88.8		
unrestricted	91.1		

Table 1 shows that there is a consistent increase in accuracy with increasing subtree depth for all modalities. Note that the maximum tree depth in the `Essen Folk song Collection` is 3, while the maximum tree depth in the Problem Solving corpus is 6. We have previously observed this phenomenon for language in [8, 11] where we called it 'the DOP hypothesis'. This hypothesis has been corroborated for Dutch and English [8, 11, 85], for Chinese [55] and for Hebrew [85]. Moreover, the DOP hypothesis has been tested not only for tree-annotations but also for LFG-annotations [25], HPSG-annotations [79] and TAG-analyses [58]. Thus the hypothesis is robust, and seems to be independent of the nature of the annotations. Table 1 shows that the DOP hypothesis also seems to hold for music and problem solving.

Our results are very competitive compared to other parsers for language and music. For the `WSJ`, DOP+ outperforms stochastic lexicalized grammars, such as in [36, 37] and [29, 30] – see [14] for a detailed quantitative comparison. Yet, there is more recent work which outperforms the DOP+ model, in particular [75], who extend their parser with discriminative self-training, achieving a 92.1% F-score on the same standard `WSJ` split, which is a 1% improvement over DOP+. It would be interesting to see how DOP+ performs if extended with self-training. Our scores on the Essen folk songs are higher than those reported by [86: 74], but unfortunately the results are not exactly comparable, since Temperley uses a smaller test set of only 65 folk songs [86].

There is an important question as to how other proposals for a unified DOP model perform. For this Chapter, we therefore accomplished an additional series of experiments with an alternative unifying DOP model which first computes the most probable tree and next selects the shortest derivation in case the most probable tree is not unique. Table 2 shows the results of these experiments for different subtree depths using the same training/test set divisions as for Table 1.

Table 2. F-scores of an alternative DOP model for different subtree depths

Max. Subtree Depth	Language *(Wall St. Journal)*	Music *(Essen Folksongs)*	Problem-Solving *(Physics Corpus)*
1	70.4	62.7	20.8
2	78.3	76.9	25.0
3	80.1	86.5	37.5
4	82.6		45.8
6	84.4		50.0
8	85.6		
10	86.3		
unrestricted	88.7		

We again note that there is an increase in accuracy with increasing subtree size, but this time the best F-scores are considerably lower than in Table 1. For language and music the differences are a few percent only, but for problem solving the difference is nearly 40%: while DOP+ predicts for 21 out of 24 problems the correct derivation tree, the alternative DOP model only gets 12 out of 24 correct. Thus by first computing the most probable tree instead of the shortest derivation, the best score of the alternative DOP model is even worse than DOP+'s score at subtree-depth 2 for problem solving. We already explained why this may be the case: the most probable tree is a bad metric for small corpora, especially if such corpora have very specific labels. The shortest derivation, on the other hand, is a good metric in almost all cases, and by selecting among a few remaining shortest derivations (in case the shortest derivation is not unique), the differences in frequency apparently do work out well, also for the small problem-solving corpus.

To check whether these differences are statistically significant, we performed a series of experiments using a 10-fold division into random training and test sets for language and music with unrestricted subtree depth only. It turns out that the differences in the best accuracies between DOP+ and the alternative unifying DOP model are statistically significant, both for language ($p \leq 0.05$) and music ($p \leq 0.02$) using paired t-testing. We did not test on different training/test set divisions of the problem solving corpus, since the training set already consists of the *actual* exemplars used in the textbooks on classical and fluid dynamics. Moreover, these exemplars closely correspond to those in other textbooks (see [51] for a comparison between physics textbooks).

What should we learn from these experiments? While we have already qualitatively explained why large subtrees are important for language (Section 2), music (Section 3) and problem-solving (Section 4), our experiments show that this can also be quantitatively supported. Our results show that directly applying statistical computations is inferior to first computing the space of 'most similar' trees by means of the shortest derivations. The best model first maximizes similarity and next probability. The maximization of similarity may be reminiscent of Case-Based Reasoning [28, 90], but DOP+ additionally defines a probabilistic distribution over equally similar structures to break ties.

9 Current Developments: Unsupervised DOP

The DOP approach in this Chapter presents a fully supervised learning technique: it starts out from corpora of example-derivations for language, music and problem-solving. A drawback of supervised learning is that it is extremely costly to create such annotated corpora. Moreover, all supervised approaches have reached an asymptote on annotated corpora. It has therefore become increasingly clear that the next major step consists of generalizing these supervised approaches to semi-supervised or even unsupervised learning since they can directly operate with unlabeled raw data, of which virtually unlimited quantities are available.

In particular in NLP, there has been considerable progress in unsupervised learning during the last few years. The performance of unsupervised parsers has gone up from around 40% unlabeled F-score on the ATIS corpus [33, 89] to around 78% F-score on the `Wall Street Journal` (WSJ) corpus [64]. Yet, all unsupervised parsing models proposed so far limit either the lexical or the structural context that is taken into account, or both. That is, these unsupervised models operate by statistically comparing contiguous subsequences of sentences: if substrings appear in similar lexical contexts they are likely to form a constituent of the same category [33, 64, 65, 90]. However, for building accurate unsupervised parsers it is imperative to also take into account *non*-contiguous substrings. This may be illustrated by the comparative construction 'more...than' in the sentence "BA carried more people than cargo in 2004". Furthermore, there exist many more lexical dependencies which may be separated by any number of other words and which can therefore not be described by contiguous substrings. Examples range from linguistic constructions such as 'if...then' to sentences such as "Companies in Vietnam are small-sized", where the subject-verb agreement is non-contiguous (it is not Vietnam that is small-sized but Companies). What would be needed is an 'all-subtrees' approach to unsupervised parsing that statistically compares all possible subtrees rather than all possible substrings.

In [17], an unsupervised generalization of DOP was proposed, termed U-DOP. Instead of using all subtrees from a set of given parse trees, U-DOP initially assigns all possible (binary) trees to a large data-set of initial sentences and next uses the subtrees from these trees to compute the best parse trees for new sentences. The underlying methodology of U-DOP is similar to (supervised) DOP: since we do not know beforehand what kind of structures are appropriate, we should not *a priori* restrict the set of possible structures, but take them all and learn only those structures (and subtrees thereof) that are useful in analyzing new data. U-DOP thus allows initially for any partial non-contiguous string to form a syntactic group and is therefore richer than previous unsupervised parsing methods.

To give an illustration of this U-DOP model, consider the following part-of-speech (p-o-s) string NNS VBD JJ NNS from the Wall Street Journal which may correspond to the sentence "Investors suffered heavy losses", (contrary to DOP, U-DOP currently works with p-o-s strings that are first tagged by a – possibly unsupervised – part-of-speech tagger). U-DOP starts by assigning all possible binary trees to this string, where each root node is labeled 'S' and each internal node is labeled 'X'. Thus NNS VBD JJ NNS has a total of five binary trees as shown in Figure 16 – where for readability we add words as well. New sentences can then be parsed by combining subtrees from all possible

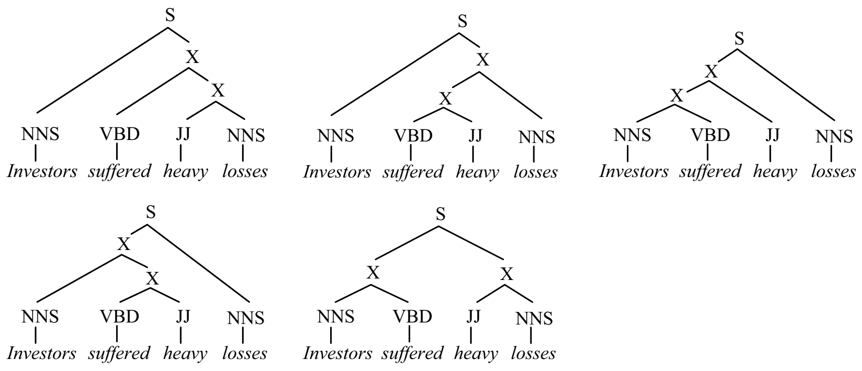

Fig. 16. All binary trees for "Investors suffered heavy losses"

trees for given sentences, just as with the DOP+ model. We again let the DOP approach decide which trees – and subtrees thereof – are most useful in analyzing fresh data. Of course, if we only had the sentence "Investors suffered heavy losses" in our corpus, there would be no difference in probability or derivation length between the five parse trees in Figure 16. However if we also have a different sentence where JJ NNS (heavy losses) appears in a different context, such as in "Heavy losses were reported", its covering subtree gets a relatively higher frequency and the parse tree where 'heavy losses' occurs as a constituent gets a higher total probability.

While we can efficiently represent the set of all binary trees of a string by means of a chart, we need to unpack the chart if we want to extract subtrees from this set of binary trees. Also, since the total number of binary trees for the WSJ10 part (in other words, all WSJ sentences up to 10 words) is already 12 million, it is doubtful that we can apply the unrestricted U-DOP model to the `WSJ` in general. The U-DOP model in [17] therefore randomly samples a large subset from the total number of parse trees from the chart, and next computes the most probable parse trees for new sentences. In [18], U-DOP was extended with maximum likelihood training, using the Expectation-Maximization (EM) algorithm with cross-validation, called UML-DOP. It was shown that UML-DOP obtained the best reported results on inducing tree structures for three benchmarks: the English `WSJ` corpus, the German `Negra` corpus and the Chinese `Treebank for Mandarin` [18]. Moreover, we showed that UML-DOP even outperformed a well-known supervised parsing model, namely the treebank grammar from the `WSJ` corpus. This result was surprising since common wisdom had it that unsupervised approaches performed worse than supervised approaches. This result brought [18] to predict that the end of supervised parsing might be in sight.

While these recently developed unsupervised DOP models are thus very promising, there is still much work to be done: the UML-DOP model does not operate directly with word strings (due to data sparseness) and it neither induces syntactic categories or verb-argument structures. Moreover, unsupervised DOP models must still be developed for music and problem-solving. An overview paper on Unsupervised Data-Oriented Parsing must therefore await further research.

10 Conclusion

All state-of-the-art parsing systems are nowadays probabilistic and corpus-based. In this Chapter, we discussed the details of a well-known parsing approach, called DOP, which parses new data by probabilistically combining subtrees from a corpus of previously parsed data. DOP takes all subtrees and lets the statistics decide which subtrees contribute to the most probable parse trees. We showed how a particular instantiation of DOP, known as DOP+, integrates the notions of 'simplicity' and 'likelihood', and how it can be successfully applied to three different modalities: language, music and problem solving. We reported on experiments that show a consistent increase in accuracy if larger corpus subtrees are taken into account. Finally, we showed how the DOP approach can be extended to unsupervised learning by using the same underlying principle: assign all binary trees to all sentences and let the statistics decide which trees (and subtrees) are most useful in parsing new sentences.

Acknowledgements

We gratefully acknowledge support by way of the NWO Incentive Scheme Project 'Towards a Unifying Model for Linguistic, Musical and Visual Processing'. This research was also partially supported by the EPSRC Advanced Research Fellowship 'Combining Linguistic and Statistical Approaches to Spoken Language Processing'. The author is particularly grateful to Professor John Fulcher for his excellent help with editing this Chapter. As usual, all errors and inconsistencies remain the author's responsibility.

References

1. Abeillé A (ed.) (2003) *Treebanks.* Kluwer Academic Publishers, Dordrecht, The Netherlands.
2. Alonso M, Finn E (1996) *Physics.* Addison Wesley, Reading, MA.
3. Baader F, Nipkow T (1998) *Term Rewriting and All That.* Cambridge University Press, UK.
4. 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: *Proc. 5th DARPA Speech and Natural Language Workshop*, Pacific Grove, CA, Morgan Kaufmann, San Mateo, CA: 306–311.
5. Black E, Lafferty J, Roukos S (1992) Development and evaluation of a broad-coverage probabilistic grammar of English-language computer manuals. In: *Proc. 30th Association Computer Linguistics Conf. (ACL'92)*, Newark, DE, Association for Computer Linguistics, Stroudsburg, PA: 185–192.
6. Black E, Garside R, Leech G (1993) *Statistically-Driven Computer Grammars of English: The IBM/Lancaster Approach.* Rodopi, Amsterdam, The Netherlands.
7. Bod R (1992) Data-oriented parsing. In: *Proc. Computational Linguistics Conf. (COLING'92)*, Nantes, France, Association for Computer Linguistics, Stroudsburg, PA: 854–859.
8. Bod R (1998) *Beyond Grammar: An Experience-Based Theory of Language.* Stanford: CSLI Publications (Lecture Notes number 88), distributed by Cambridge University Press, Cambridge, UK.
9. Bod R (1999) Context-sensitive spoken dialogue processing with the DOP model. *Natural Language Engineering,* 5(4): 309–323.
10. Bod R (2000) Parsing with the shortest derivation. In: *Proc. 18th ACL Computational Linguistics Conf. (COLING'2000)*, Saarbrücken, Germany, Association for Computer Linguistics, Stroudsburg, PA: 69–75.
11. Bod R (2001) What is the minimal set of subtrees that achieves maximal parse accuracy? In: *Proc. 39th Association Computer Linguistics Conf. (ACL'2001)*, Toulouse, France, Association for Computer Linguistics, Stroudsburg, PA: 66–73.
12. Bod R (2002) A unified model of structural organization in language and music. *J. Artificial Intelligence Research,* 17: 289–308.
13. Bod R (2002) Memory-based models of melodic analysis: challenging the Gestalt principles. *J. New Music Research,* 31(1): 27–37.

14. Bod R (2003) An efficient implementation of a new DOP model. In: *Proc. 10th European Association Computer Linguistics Conf. (EACL'03)*, 12-17 April, Budapest, Hungary, Association for Computer Linguistics, Stroudsburg, PA: 19–26.
15. Bod R (2004) Exemplar-based explanation. In: *Proc. Computation and Philosophy Conf. (ECAP04)*, 3-5 June, Pavia, Italy.
16. Bod R (2005) Modeling scientific problem solving by DOP. In: *Proc. Cognitive Science Conf. (CogSci'05)*. Stresa, Italy: 103.
17. Bod R (2006) Unsupervised parsing with U-DOP. In: *Proc. 10th Computational Natural Language Learning Conf. (CONLL'2006)*, 8-9 June, New York, NY, Association for Computer Linguistics, Stroudsburg, PA: 85–92.
18. Bod R (2006) An all-subtrees approach to unsupervised parsing. In: *Proc. ACL Computational Linguistics Conf. (COLING'2006)*, Sydney, Australia, Association for Computer Linguistics, Stroudsburg, PA: 865–872.
19. Bod R (2006) Towards a general model of applying science. *Intl. Studies in the Philosophy of Science,* 20(1): 5–25.
20. Bod R (2006) Exemplar-based reasoning with the shortest derivation. In: Magnani L (ed.) *Model-Based Reasoning in Science and Engineering.* College Publications, London, UK: 119–140.
21. Bod R (2006) Exemplar-based syntax: how to get productivity from examples. *The Linguistic Review* (Special Issue on Exemplar-Based Models in Linguistics), 23(3): 289–318.
22. Bod R, Kaplan R (1998) A probabilistic corpus-driven model for lexical-functional analysis. In: *Proc. ACL Computational Linguistics Conf. (COLING'98)*, 10-14 August, Montreal, Canada, Association for Computer Linguistics, Stroudsburg, PA: 145–152.
23. Bod R, Hay J, Jannedy S (eds.) (2003) *Probabilistic Linguistics.* MIT Press, Cambridge, MA.
24. Bod R, Scha R, Sima'an K (eds.) (2003) *Data-Oriented Parsing.* University of Chicago Press, Chicago, IL.
25. Bod R, Kaplan R (2003) A DOP model for lexical-functional grammar. In: Bod R, Scha R, Sima'an K (eds.) (2003) *Data-Oriented Parsing.* University of Chicago Press, Chicago, IL.
26. Bonnema R, Bod R, Scha R (1997) A DOP model for semantic interpretation. In: *Proc. 4th European Association Computer Linguistics Conf. (EACL'97)*, Madrid, Spain, Association for Computer Linguistics, Stroudsburg, PA: 159–167.
27. Briscoe T, Waegner N (1992) Robust stochastic parsing using the inside-outside algorithm. In: *Proc. AAAI Workshop Statistically-Based Techniques in Natural Language Processing*, Menlo Park, CA, AAAI Press/MIT Press, Cambridge, MA: 39–53.
28. Carbonell J (1993) Derivational analogy: a theory of reconstructive problem solving and expertise acquisition. In: Michalski RS, Carbonell J, Mitchell T (eds.) *Machine Learning II.* Morgan Kaufmann, San Francisco, CA: 371–392.
29. Charniak E (1997) Statistical techniques for natural language parsing. *AI Magazine*, Winter: 32–43.
30. Charniak E (2000) A maximum-entropy-inspired parser. In: *Proc. 1st North American ACL Chapter Conf. (ANLP-NAACL'2000)*, Seattle, WA, Morgan Kaufmann, San Francisco, CA: 132–139.

31. Chater N (1999) The search for simplicity: a fundamental cognitive principle? *The Quarterly J. Experimental Psychology*, 52A(2): 273–302.
32. Chiang D (2000) Statistical parsing with an automatically extracted tree adjoining grammar. In: *Proc. 38th Association Computer Linguistics Conf. (ACL'2000)*, October, Hong Kong, China, Association for Computer Linguistics, Stroudsburg, PA: 456–463.
33. Clark A (2001) Unsupervised induction of stochastic context-free grammars using distributional clustering. In: *Proc. Computational Natural Language Learning Conf. (CoNLL'2001)*, July, Toulouse, France, Association for Computer Linguistics, Stroudsburg, PA: 97–104.
34. Chomsky N (1965) *Aspects of the Theory of Syntax.* MIT Press, Cambridge MA.
35. Collins M (1996) A new statistical parser based on Bigram lexical dependencies. In: *Proc. 34th Association Computer Linguistics Conf. (ACL'96)*, 23-28 June, Santa Cruz, CA, Association for Computer Linguistics, Stroudsburg, PA: 184–191.
36. Collins M (1997) Three generative lexicalised models for statistical parsing. In: *Proc. 35th Association Computer Linguistics Conf. (ACL'97)*, July, Madrid, Spain, Association for Computer Linguistics, Stroudsburg, PA: 16–23.
37. Collins M (1999) Head-Driven Statistical Models for Natural Language Parsing. *PhD Thesis*, University of Pennsylvania, PA.
38. Collins M (2000) Discriminative reranking for natural language parsing. In: *Proc. 17th Intl. Conf. Machine Learning (ICML-2000)*, Stanford, CA: 175–182.
39. Collins M, Duffy N (2001) Convolution kernels for natural language. In: Dietrich TG, Becker S, Gharamani Z (eds.) *Advances in NIPS 14* (Proc. NIPS'2001), 3-8 December, Vancouver, Canada, MIT Press, Cambridge, MA: 617–624.
40. Collins M, Duffy N (2002) New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In: *Proc. 38th Association Computer Linguistics Conf. (ACL'2002)*, Philadelphia, PA, Association for Computer Linguistics, Stroudsburg, PA: 263–270.
41. Conklin D (2006) Melodic analysis with segment classes. *Machine Learning,*65(2-3): 349–360.
42. Cussens J (2001) Parameter estimation in stochastic logic programs. *Machine Learning,* 44(3): 245–271.
43. Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society,* 39: 1–38.
44. De Raedt L, Kersting K (2004) Probabilistic inductive logic programming. In: *Proc. Algorithmic Learning Theory (ALT) Conf.*, Lecture Notes in Computer Science 3244, Springer-Verlag, Berlin: 19–36.
45. Douglas J, Matthews R (1996) *Fluid Mechanics 1 (3rd ed.).* Longman, Essex, UK.
46. Eisner J (1996) Three new probabilistic models for dependency parsing: an exploration. In: *Proc. 18th ACL Computational Linguistics Conf. (COLING'96)*, August, Copenhagen, Denmark, Association for Computer Linguistics, Stroudsburg, PA: 340–345.
47. Ferrand M, Nelson P, Wiggins G (2003) Unsupervised learning of melodic segmentation: a memory-based approach. In: *Proc. 5th European Society for the Cognitive Sciences of Music Conf. (ESCOM'2003)*, 8-13 September, Hanover, Germany.

48. Frazier L (1978) On Comprehending Sentences: Syntactic Parsing Strategies. *PhD Thesis*, University of Connecticut.
49. Fujisaki T, Jelinek F, Cocke J, Black E, Nishino T (1989) A probabilistic method for sentence disambiguation. In: *Proc. 1st Intl. Workshop Parsing Technologies*, 28-31 August, Pittsburgh, PA: 85–94.
50. Gahl S, Garnsey S (2004) Knowledge of grammar, knowledge of usage: syntactic probabilities affect pronunciation variation. *Language,* 80(4): 748–775.
51. Giere R (1988) *Explaining Science: A Cognitive Approach.* University of Chicago Press, Chicago, IL.
52. Goldberg A (2006) *Constructions at Work.* Oxford University Press, Oxford, UK.
53. Goodman J (1996) Efficient algorithms for parsing the DOP model. In: *Proc. Empirical Methods in Natural Language Processing*, Philadelphia, PA: 143–152.
54. Goodman J (2003) Efficient parsing of DOP with PCFG-reductions. In: Bod R, Scha R, Sima'an K (eds.) *Data-Oriented Parsing.*University of Chicago Press, Chicago, IL.
55. Hearne M, Way A (2003) Seeing the wood for the trees: data-oriented translation. In: *Proc. Machine Translation Summit IX*, September, New Orleans, LO: 165–172.
56. Hearne M, Way A (2004) Data-oriented parsing and the Penn Chinese Treebank. In: *Proc. 1st Intl. Joint Conf. Natural Language Processing*, May, Hainan Island, China: 406–413.
57. Hearne M, Way A (2006) Disambiguation strategies for data-oriented translation. In: *Proc. 11th Intl. Conf. European Association for Machine Translation*, 19-20 June, Oslo, Norway.
58. Hoogweg L (2003) Extending DOP with insertion. In: Bod R, Scha R, Sima'an K (eds.) *Data-Oriented Parsing.* University of Chicago Press, Chicago, IL.
59. Huron D (1996) The melodic arch in western folksongs. *Computing in Musicology,* 10: 2–23.
60. Johnson M (1998) PCFG models of linguistic tree representations. *Computational Linguistics,* 24(4): 613–632.
61. Johnson M (2002) The DOP estimation method is biased and inconsistent. *Computational Linguistics,* 28(1): 71–76.
62. Jurafsky D (2003) Probabilistic modeling in psycholinguistics. In: Bod R, Scha R, Sima'an K (eds) *Data-Oriented Parsing.* University of Chicago Press, Chicago, IL: 39–96.
63. Klein D (2005) The unsupervised learning of natural language structure. *PhD Thesis*, Department of Computer Science, Stanford University, CA.
64. Klein D, Manning C (2002) A general constituent-context model for improved grammar induction. In: *Proc. 40th Association Computer Linguistics Conf. (ACL'2002)*, July, Philadelphia, PA, Association for Computer Linguistics, Stroudsburg, PA: 128–135.
65. Klein D, Manning C (2004) Corpus-based induction of syntactic structure: models of dependency and constituency. *Proc. 42nd Association Computer Linguistics Conf. (ACL'2004)*, 21-26 July, Barcelona, Spain, Association for Computer Linguistics, Stroudsburg, PA: 438.
66. Kudo T, Suzuki J, Isozaki H (2005) Boosting-based parse reranking with subtree features. In: *Proc. 43rd Association Computer Linguistics Conf. (ACL'2005)*, June, Ann Arbor, MI, Association for Computer Linguistics, Stroudsburg, PA: 189–196.

67. Kuhn T (1970) *The Structure of Scientific Revolutions (2nd ed.).* University of Chicago Press, Chicago, IL.
68. Lerdahl F, Jackendoff R (1983) *A Generative Theory of Tonal Music.* MIT Press, Cambridge, MA.
69. Longuet-Higgins H (1976) Perception of melodies. *Nature,* 263, October 21: 646–653.
70. Longuet-Higgins H, Lee C (1987) The rhythmic interpretation of monophonic music. In: Longuet-Higgins H (ed.) *Mental Processes: Studies in Cognitive Science*, MIT Press, Cambridge, MA.
71. Makatchev M, Jordan P, VanLehn K (2004) Abductive theorem proving for analyzing student explanations to guide feedback in intelligent tutoring systems. *J. Automated Reasoning*, (Special Issue: Automated Reasoning and Theorem Proving in Education), 32(3): 187–226.
72. Manning C (2003) Probabilistic syntax. In: Bod R, Hay J, Jannedy S (eds.) *Probabilistic Linguistics.* MIT Press, Cambridge, MA: 289–342.
73. Manning C, Schtze H (1999) *Foundations of Statistical Natural Language Processing.* MIT Press, Cambridge, MA.
74. Marcus M, Santorini B, Marcinkiewicz M (1993) Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics,* 19(2): 313–330.
75. McClosky D, Charniak E, Johnson M (2006) Effective self-training for parsing. In: *Proc. North American Chapter of ACL Conf. Human Language Technology (NAACL-HLT 2006)*, June, New York, NY, Association for Computer Linguistics, Stroudsburg, PA: 152–159.
76. Mitchell T, Keller R, Kedar-Cabelli S (1986) Explanation-based learning: a unifying view. *Machine Learning,* 1: 47–80.
77. Mooney J, Zelle J (1994) Integrating ILP and EBL. *SIGART Bulletin,* 5(1): 12–21.
78. Muggleton S (1996) Stochastic logic programs. In: De Raed L (ed.) *Advances in Inductive Logic Programming* (Proc. 5th Intl. Workshop Inductive Logic Programming), IOS Press, Amsterdam, The Netherlands: 254–264.
79. Neumann G (2003) A data-oriented approach to HPSG. In: Bod R, Scha R, Sima'an K (eds) *Data-Oriented Parsing.* University of Chicago Press, Chicago, IL.
80. Pereira F, Schabes Y (1992) Inside-outside reestimation from partially bracketed corpora. In: In: *Proc. 30th Association Computer Linguistics Conf. (ACL'92)*, Newark, DL, Association for Computer Linguistics, Stroudsburg, PA: 128–135.
81. Scha R (1990) Taaltheorie en taaltechnologie; competence en performance. In: de Kort Q, Leerdam G (eds) *Computertoepassingen in de Neerlandistiek* . Landelijke Vereniging van Neerlandici (LVVN-jaarboek), Almere, The Netherlands.
82. Schaffrath H (1995) The Essen Folksong Collection in the Humdrum Kern Format. In: Huron D (ed.) *Probabilistic Grammars for Music.* Center for Computer Assisted Research in the Humanities, Menlo Park, CA.
83. Sima'an K (1996) Computational complexity of probabilistic disambiguation by means of tree grammars. In: *Proc. 14th Computational Linguistics Conf. (COLING'96)*, 5-9 August, Copenhagen, Denmark, Association for Computer Linguistics, Stroudsburg, PA: 1175–1180.
84. Sima'an K (1999) Learning Efficient Disambiguation. *ILLC Dissertation Series 1999-02*, Utrecht University, The Netherlands.

85. Sima'an K, Itai A, Winter Y, Altman A, Nativ N (2001) Building a tree-bank of modern Hebrew text. *J. Traitement Automatique des Langues* (Special Issue on Natural Language Processing and Corpus Linguistics), 42(2): 347–380.
86. Temperley D (2001) *The Cognition of Basic Musical Structures.* MIT Press, Cambridge, MA.
87. Tomasello M (2003) *Constructing a Language.* Harvard University Press, Harvard, MA.
88. Van Lehn K (1998) Analogy events: how examples are used during problem solving. *Cognitive Science,* 22(3): 347–388.
89. van Zaanen M (2000) ABL: alignment-based learning. In: *Proc. 18th Computational Linguistics Conf. (COLING'2000)*, 31 July - 4 August, Saarbrücken, Germany, Association for Computer Linguistics, Stroudsburg, PA: 961–967.
90. van Zaanen M (2002) Bootstrapping Structure into Language. *PhD thesis.* School of Computing, University of Leeds, UK.
91. van Zaanen M, Bod R, Honing H (2003) A memory-based approach to meter induction. In: *Proc. 5th European Society for the Cogntivie Sciences of Music Conf. (ESCOM5)*, September, Hanover, Germany: 250–253.
92. Veloso M, Carbonell J (1993) Derivational analogy in PRODIGY: automating case acquisition, storage, and utilization. *Machine Learning,* 10(3): 249–278.
93. Wertheimer M (1923) Untersuchungen zur lehre von der gestalt. *Psychologische Forschung,* 4: 301–350.
94. Younger D (1967) Recognition and parsing of context-free languages in time n3. *Information and Control*, 10(2): 189–208.
95. Zollmann A, Sima'an, K (2005) A consistent and efficient estimator for data-oriented parsing. *J. Automata, Languages and Combinatorics,* 10: 367–388.
96. Zuidema W (2006) What are the productive units of natural language grammar? A DOP approach to the automatic identification of constructions. In: *Proc. 10th Computational Natural Language Learning Conf. (CONLL'2006)*, 8-9 June, New York, NY, Association for Computer Linguistics, Stroudsburg, PA: 29–36.

A

Resources

A.1 Key Books

Cassell J, Sullivan J, Prevost S, Churchill E (eds.) (2000) *Embodied Conversational Agents.* MIT Press, Cambridge, MA.

Damasio A (1994) *Descartes Error.* Macmilliam Publishers, London, UK.

Evans D (2001) *Emotion: the Science of Sentiment.* Oxford University Press, New York, NY.

LeDoux J (1996) *The Emotional Brain.* Simon and Schuster, New York, NY.

Oatley K, Jenkins JM (1996) *Understanding Emotions.* Blackwell Publishers, Oxford, UK.

Picard R (1997) *Affective Computing.* MIT Press, Cambridge, MA.

Plantec P (2004) *Virtual Humans.* Amacon, New York, NY.

Prendinger H, Ishizuka M (2004) *Life-Like Characters. Tools, Affective Functions, and Applications.* Springer-Verlag , Berlin.

Reeves B, Nass C (1996) *The Media Equation: How People Treat Computers, Televisions, and New Media Like Real People and Places.* Cambridge University Press, New York, NY.

A.2 Key Survey/Review Articles

Bates J (1994) The role of emotion in believable agents. *Communications ACM,* 37(7): 122–125.

Cowie R, et al. (2001) Emotion recognition in human-computer interaction. *IEEE Signal Processing Magazine,* 18(1): 32–80.

Dehn D, Van Mulken S (2000) The impact of animated interface agents: a review of empirical research. *Intl. J. Human-Computer Studies,* 52(1): 1–22.

Brave S, Nass C (2002) Emotion in human-computer interaction. In: Jacko JA, Sears A (eds.) *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications.* Lawrence Erlbaum Associates, Mahwah, NJ: 81–96.

Picard RW, Klein J (2002) Computers that recognise and respond to user emotion: theoretical and practical implications. *Interacting with Computers,* 14(2): 141–169.

http://emotion-research.net/deliverables HUMAINE (Human-Machine Interaction Network on Emotion) deliverable Dxx: Proposed exemplar and work towards it:

Scherer K et al. (2005)	D3e: Theory of Emotion.
Douglas-Cowie E, et al. (2005)	D5e: Data and Databases.
Kollias S, et al. (2005)	D4d: Signals and Signs of Emotion.
Pelachaud C, et al. (2005)	D6d: Emotion in Interaction.
Canamero L, et al. (2005)	D7d: Emotion in Cognition and Action.
Stock O, et al. (2005)	D8d: Communication and Emotions.
Hook K, et al. (2005)	D9d: Usability.
Goldie P, et al. (2005)	D10b: Interim report on ethical frameworks for emotion-oriented systems.

A.3 Organisations, Societies, Special Interest Groups

CHIL (Computers in the Human Loop)
http://chil.server.de/servlet/is/101/

COSY (Cognitive Systems for Cognitive Assistants)
http://www.cognitivesystems.org/

Design and Emotion Society
http://www.designandemotion.org/

Enactive Interfaces (EU Network of Excellence)
http://www.reflex.lth.se/enactive/

HUMAINE (Human-Machine Interaction Network on Emotion)
http://emotion-research.net/

International Society for Research on Emotion
http://isre.org/prd/index.php

SIMILAR (The European taskforce creating human-machine interfaces SIMILAR to human-human interfaces)
http://www.similar.cc/

Virtual Human (Anthropomorphic Interaction Agents)
http://www.virtual-human.org/start_en.html

A.4 Research Groups

Affective Computing at MIT Media Lab
http://affect.media.mit.edu/

Cognition and Affect Project at University of Birmingham (UK)
http://www.cs.bham.ac.uk/research/projects/cogaff

Geneva Emotion Research Group
http://www.unige.ch/fapse/emotion/

LeDoux Lab, New York University
http://www.cns.nyu.edu/home/ledoux/

Relational Agents Group, Northeastern University
http://www.ccs.neu.edu/research/rag/

RITL (Center for Research of Innovative Technologies for Learning, Florida State University)
http://ritl.fsu.edu/

Virtual Reality Lab, Swiss Federal Institute of Technology
http://ligwww.epfl.ch/

A.5 Discussion Groups, Forums

The Emotion Forum
http://homepages.feis.herts.ac.uk/ comqlc/emotion.html

Emotional Intelligence Information Website
http://www.unh.edu/emotional_intelligence/

Facial Action Coding System (FACS) Manual
http://face-and-emotion.com/dataface/facs/description.jsp

Facial Expressions Resources Page
http://www.kasrl.org/facial_expression.html

Socially Intelligent Agents
http://homepages.feis.herts.ac.uk/~ comqkd/aaai-social.html

Stanford University Persuasive Technology Lab
http://captology.stanford.edu/

Virtual Humans
http://www.ordinarymagic.com/v-people/#

A.6 Key International Conferences/Workshops

ACII 2005: 1st Intl. Conf. Affective Computing and Intelligent Interaction
http://www.affectivecomputing.org/2005/

ACE 2006: Agent Construction and Emotions: Modeling the Cognitive Antecedents and Consequences of Emotion
http://www.ofai.at/~ paolo.petta/conf/ace2006/

Theories and Models of Emotion (HUMAINE Workshop – 2004)
http://emotion-research.net/ws/wp3

From Signals to Signs of Emotion and Vice Versa (HUMAINE Workshop – 2004)
http://emotion-research.net/ws/wp4

Data and Databases (HUMAINE Workshop – 2004)
http://emotion-research.net/ws/wp5

Emotion in Interaction (HUMAINE Workshop – 2005)
http://emotion-research.net/ws/wp6/

Emotion in Cognition and Action (HUMAINE Workshop – 2005)
http://emotion-research.net/ws/wp7

Emotion in Communication (HUMAINE Workshop – 2005)
http://emotion-research.net/ws/wp8/proceedings-wswp8.pdf

Innovative Approaches for Evaluating Affective Systems (HUMAINE Workshop – 2006) *http://emotion-research.net/ws/wp9/*

A.7 (Open Source) Software

Croquet (Software for creating 3D collaborative multi-user online applications) *http://www.opencroquet.org/*

Emofilt (Simulate emotional arousal with speech synthesis)
http://felix.syntheticspeech.de/publications/emofiltInterspeech05.pdf

FEELTRACE (Tool for rating the emotion expressed in audio-visual stimuli)
http://emotion-research.net/download/Feeltrace%20Package.zip

OpenAL (Cross Platform 3D Audio)
http://www.openal.org/

OpenGL (Graphics API)
http://www.opengl.org/

OpenMary (Open Source Emotional Text-to-Speech Synthesis System)
http://mary.dfki.de

TraceTools (Tools for tracing the presence of emotion)
http://emotion-research.net/download/ECatPack.zip

A.8 Data Bases

A.8.1 Multimodal Databases

Belfast Naturalistic Database
http://www.idiap.ch/mmm/corpora/emotion-corpus

ISLE project corpora
http://isle.nis.sdu.dk/

SMARTKOM
http://www.phonetik.uni-muenchen.de/Bas/BasMultiModaleng.html#SmartKom

SALAS
http://www.image.ntua.gr/ermis/

A.8.2 Face Databases

AR Face Database
http://cobweb.ecn.purdue.edu/~aleix/aleix_face_DB.html

CMU Facial Expression Database (Cohn-Kanade)
http://vasc.ri.cmu.edu//idb/html/face/facial_expression/index.html

CMU PIE (Pose, Illumination and Expression) Database
http://www.ri.cmu.edu/projects/project_418.html

CVL Face Database
http://www.lrv.fri.uni-lj.si/facedb.html

Psychological Image Collection at Stirling
http://pics.psych.stir.ac.uk/

Japanese Female Facial Expression (JAFFE) Database
http://www.kasrl.org/jaffe.html

Yale Face Database
http://cvc.yale.edu/projects/yalefaces/yalefaces.html

Yale Face Database B
http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html

Index

ambiguity problem, 5
analysis tree, 8
annotated sentence, 3
annotation, HPSG, 26
annotation, LFG, 26
annotator, human, 3
artificial intelligence (AI), 5
ATIS corpus, 28
automated reasoning, 13

benchmark, natural language processing (NLP), 9

case-based reasoning (CBR), 28
cognitive science, 5
combination operation, 16
computational learning, 1
computational music analysis, 1
context-free grammar (CFG), 7, 18, 22
context-free grammar, probabilistic (PCFG), 22
context-free grammar, stochastic, 25
context-free rule, 1
corpora, LFG-annotated, 8
corpus, 1
corpus fragment, 2
corpus, parsed, 2
corpus-based parsing system, 30
corpus-based, probabilistic parsing, 1

data-oriented parsing (DOP), 1, 30
definite clause grammar, 18
depth-1 subtree, 1, 7
derivation length, 8
derivation tree, 13, 18
derivation, language, 14
derivation, most probable, 23
derivation, music, 14
derivation, physics, 14
derivation, shortest, 18, 23
derivational problem solving, 14
deterministic parsing, 1
dialog processing, 2
disambiguation, 6
disambiguation, syntactic, 5
discourse structure, 5
DOP model, tree-based, 9
DOP+, 7, 30
DOP, ML, 9
DOP, supervised, 29
DOP, UML, 30
DOP, unsupervised, 2, 28, 30

Essen Associative Code (ESAC), 11
Essen Folksong Collection (EFC), 6, 11, 24
exemplar, 16
exemplary problem solution, 16
expectation-maximization (EM) algorithm, 9, 30
explanation-based learning, 13

grammar, 1
grammar, context-free (CFG), 7, 18, 22
grammar, definite clause, 18
grammar, head-driven phrase structure (HPSG), 8
grammar, internalized, 3

grammar, lexical-functional (LFG), 8
grammar, probabilistic context-free (PCFG), 2, 23, 24
grammar, stochastic context-free, 25
grammar, stochastic lexicalized, 26

head-driven phrase structure grammar (HPSG), 8
head-word, 2
HPSG annotation, 26
human annotator, 3

idiomatic phrase, 5
inductive logic programming, 13
internalized grammar, 3

label substitution, 3
label substitution operation, 7
labeled tree, 13
language comprehension, 5
language derivation, 14
learning, computational, 1
learning, explanation-based, 13
learning, relational, 1
learning, unsupervised, 28
length of derivation, 8
lexical grammar, stochastic, 26
lexical-functional grammar (LFG), 8
lexical-syntactic category, 3
LFG annotation, 26
LFG-annotated corpora, 8
linguistic parsing, 22
linguistic productivity, 5
linguistics, psycho-, 5
logic program, stochastic, 18
logic programming, 1
logic programming, inductive, 13

machine translation, 2
maximum likelihood estimator, 9
melodic analysis, 2, 9
ML-DOP, 9
most probable derivation, 23
Mozart's G minor symphony, 10
music analysis, computational, 1
music derivation, 14
musical parsing, 22

natural language processing (NLP), 1, 28
NLP benchmark, 9
nonhead-word, 2
noun phrase (NP), 3, 4, 10

operation, combination, 16
operation, substitution, 3, 11, 13, 14, 16, 19, 21

parse accuracy, 2
parse tree, 2, 8
parsed corpus, 2
parseval metric, 25
parsing model, 1
parsing system, 30
parsing system, corpus-based, 30
parsing system, probabilistic, 30
parsing, corpus-based probabilistic, 1
parsing, data-oriented (DOP), 1
parsing, deterministic, 1
parsing, linguistic, 22
parsing, musical, 22
parsing, probabilistic, 1
parsing, rule-based deterministic, 1
parsing, supervised, 30
part-of-speech string, 29
part-of-speech tagger, 29
Penn Treebank, 3, 6, 24
phrase structure tree, 3, 9
phrase, idiomatic, 5
phrase, noun (NP), 3, 4, 10
phrase, prepositional (PP), 3
phrase, verb (VP), 3, 4, 10
physics derivation, 14
prepositional phrase (PP), 3
probabilistic context-free grammar (PCFG), 2, 22–24
probabilistic parsing, 1
probabilistic parsing system, 30
probability distribution, tree, 6
probability, tree, 8
problem solution, exemplary, 16
problem solving, 22
problem solving, derivational, 14
productivity, linguistic, 5
programming, logic, 1
proof tree, 13
psycholinguistics, 5

reasoning, automated, 13

relational learning, 1
relational learning, statistical, 13
relational learning, stochastic, 18
rule probability, 1
rule, context-free, 1
rule-based, deterministic parsing, 1

semantic annotation, 5
sentence, 10
sentence structure, 1
sentence structure fragment, 1
sentence, annotated, 3
shortest derivation, 18, 23
simplicity, 5
speech understanding, 2
statistical relational learning, 13
stochastic context-free grammar, 25
stochastic lexicalized grammar, 26
stochastic logic program, 18
stochastic relational learning, 18
stochastic tree substitution grammar (STSG), 22
structure simplicity, 5
substitution operation, 3, 11, 13, 14, 16, 19, 21
substitution, label, 3
subtree, 1
subtree, depth-1, 1, 7
subtree, treebank, 8
supervised DOP, 29
supervised parsing, 30
syntactic disambiguation, 5
system, corpus-based parsing, 30
system, parsing, 30
system, probabilistic parsing, 30

theorem proving, 13
TK Solver, 24
tree probability, 8
tree probability distribution, 6
tree structure, 1
tree substitution grammar (TSG), 22
tree substitution grammar, stochastic (STSG), 22
tree, analysis, 8
tree, derivation, 13, 18
tree, labeled, 13
tree, parse, 2, 8
tree, phrase structure, 3, 9
tree, proof, 13
tree-based DOP model, 9
treebank subtree, 8

UML-DOP, 30
unrestricted DOP (U-DOP), 28, 30
unsupervised DOP, 2, 28, 30
unsupervised learning, 28

verb phrase (VP), 3, 4, 10
Viterbi optimization, 22

Wall Street Journal (WSJ) corpus, 22, 24, 26, 28, 30