
Dependency-Length Minimization in Natural and Artificial Languages*

David Temperley
Eastman School of Music at the University of Rochester

ABSTRACT

A wide range of evidence points to a preference for syntactic structures in which dependencies are short. Here we examine the question: what kinds of dependency configurations minimize dependency length? We consider two well-established principles of dependency-length minimization; that dependencies should be consistently right-branching or left-branching, and that shorter dependent phrases should be closer to the head. We also add a third, novel, principle; that some “opposite-branching” of one-word phrases is desirable. In a series of computational experiments, using unordered dependency trees gathered from written English, we examine the effect of these three principles on dependency length, and show that all three contribute significantly to dependency-length reduction. Finally, we present what appears to be the optimal “grammar” for dependency-length minimization.

INTRODUCTION

A wide range of evidence from diverse areas of linguistic research points to a preference for syntactic structures in which dependencies between related words are short.¹ This preference is reflected in numerous psycholinguistic phenomena: structures with long dependencies appear to cause increased complexity in comprehension and tend to be avoided in production. The preference for shorter dependencies has also been cited as an explanation for certain

*Address correspondence to: David Temperley, Eastman School of Music, 26 Gibbs Street, Rochester, NY 14604. Tel: 585-274-1557. E-mail: dtemperley@esm.rochester.edu

¹Thanks to Daniel Gildea for valuable comments on a draft of this paper.

cross-linguistic phenomena, notably the tendency for languages to be predominantly “head-first” or “head-last”. Recent work, however, has shown that the head-first/head-last generalization does not characterize languages very well, and that alternative generalizations may capture the linguistic facts better. It appears also – as will be shown below – that purely head-first/head-last configurations are not optimal from the point of view of dependency-length minimization either, and that alternative principles may be more advantageous from this perspective as well.

Given this background, two basic questions arise:

1. What kinds of abstract configurations are preferable from the point of view of minimizing dependency length?
2. To what extent are these preferred configurations reflected in actual human languages?

My main focus in this study will be on the first of these questions. In a series of computational experiments, I will consider different strategies for the linear ordering of dependency structures and examine their consequences with regard to dependency length. However, I will also point to some interesting convergences between the structures that seem optimal in terms of dependency length and those that are found in natural languages.

I begin by reviewing some basic assumptions about dependency structures in language and presenting some general evidence for dependency-length minimization. I then consider three principles of dependency-length minimization. In each case, I review empirical evidence for the principle with regard to natural language, and show informally how the principle appears to reduce dependency length. I then undertake a series of computer simulations, to explore the actual effect of these principles on dependency length. Finally, I present a “grammar” which appears to yield the minimal dependency length for an unordered dependency tree.

EVIDENCE FOR DEPENDENCY-LENGTH MINIMIZATION

A dependency, as the term is normally used in language research, is an asymmetrical syntactic relation between a pair of words, the head and the

dependent; the head of one dependency is the dependent of another (unless it is the head of the entire sentence), so that the dependencies of a sentence form a directed acyclic graph which connects the entire sentence. It is generally assumed that dependencies (when drawn above the words of a sentence) may not cross, nor may any dependency cross above the head word of the sentence.² The dependency structure of a sentence closely corresponds to its constituent structure; in general, each constituent – such as a noun phrase or prepositional phrase – corresponds to a subgraph of the dependency tree consisting of a word (the head of the constituent) and all of its descendants. For present purposes, we define the subgraph of a word as the portion of the graph consisting of the word and its descendants; we will sometimes also speak of such a subgraph as a “phrase,” since – as just mentioned – such subgraphs generally correspond to phrases in conventional linguistic terms. (We must be careful in referring to a word *w* as “the head of a phrase,” as this is potentially ambiguous. In the current context, it means that *w* is within the phrase and all other words in the phrase are its descendants; the head of *w*, *h*, will then be called the “parent head” of the phrase, and we may also speak of the entire phrase as a dependent of *h*.)

There is general agreement as to the nature of dependency structures in language. In general, the head of each major constituent type (NP, VP, AP, and PP) is the category after which the constituent is named; for example, the head of a prepositional phrase is the preposition. The head of a finite clause is the finite verb; in a main clause, this word is then the head of the entire sentence, while in a subordinate clause it is the dependent of an external word such as a subordinating conjunction, relative pronoun, or complementizer. In a few cases, there are differing opinions. For example, while most have viewed the head of a noun phrase to be the main noun, some in recent theoretical linguistics have argued for the determiner as the head (Abney, 1987); however, the “noun-headed” view seems more prevalent in linguistic research as a whole, and we will adopt it here.³

²Most dependency grammars either prohibit crossing dependencies completely (Gaifman, 1965; Mel'cuk, 1987) or allow them only under very limited circumstances (Steedman, 1985; Hudson, 1990). There are, however, some well-known examples of crossing dependencies in certain languages such as Dutch (Bresnan et al., 1982; Joshi, 1990).

³Regarding views on dependencies; see, for example, Jackendoff (1977), Mel'cuk (1987), Pollard and Sag (1987), Hudson (1990), Dryer (1992), Hawkins (1994), Radford (1997),

The idea that syntactic structures with shorter dependencies are preferred has a long history; it can be traced back to Behagel's principle that closely related words in a sentence tend to be close together (Behagel, 1932). More recently, evidence for dependency-length minimization has come from a variety of sources. Gibson (1998, 2000) asserts that structures with longer dependencies are more difficult to process, and shows that this principle predicts a number of phenomena in comprehension.⁴ One example is the fact that subject-extracted relative clauses, like (1a) below, are easier to process than object-extracted relative clauses like (1b) (King & Just, 1991). In both subject and object relatives, the verb of the relative clause (*attacked*) is dependent on the preceding relative pronoun (*who*). In subject relatives, these two words are normally adjacent, while in object relatives they are separated by the relative clause subject (*the senator*); thus object relatives yield longer dependencies.

- (1a). The reporter who attacked the senator admitted the error.
 (1b). The reporter who the senator attacked admitted the error.

Phenomena of ambiguity resolution are also explained by Gibson's theory – for example, prepositional-phrase attachment decisions (Gibson & Pearlmutter, 1994; Thornton et al., 2000) and main-verb/reduced-relative ambiguities (Gibson, 2000). In such cases, the preferred interpretation tends to be the one with shorter dependencies. Dependency-length minimization is also reflected in language production (Hawkins, 1994); this will be discussed further below.

Gibson (1998) and Collins (1999). Another controversial case is co-ordination constructions: some have suggested that the head of a co-ordinate phrase like *John and Fred* is the first conjunct (Mc'cuk, 1987), others argue that it is the conjunction (Munn, 1993), and still others argue that both conjuncts act as heads, connecting to other words outside the coordinate phrase (Hudson, 1990). For the most part, however, the dependency structure of syntactic constructions is clear and unproblematic.

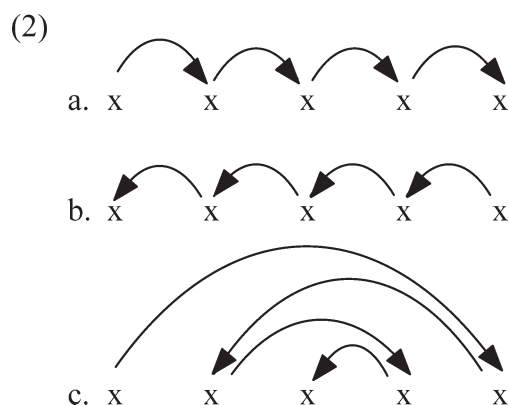
⁴Gibson first stated this principle as part of his Syntactic Prediction Locality Theory (1998), which he later modified and renamed the Dependency Locality Theory (2000). As Gibson acknowledges, a number of other theories of syntactic complexity have been put forth, going back over several decades. Gibson argues, however, that no other theory accounts for the range of phenomena that is accounted for by dependency-length minimization (see especially Gibson, 1998, pp. 1–8).

Further support for dependency-length minimization, albeit of an indirect kind, comes from computational linguistics. Several dependency-based parsing models have incorporated dependency length as a factor in parsing, preferring parses with shorter dependencies (Sleator & Temperley, 1991; Collins, 2003; Eisner & Smith, 2005). This proves to be a useful heuristic which significantly increases parsing accuracy.

THREE PRINCIPLES OF DEPENDENCY-LENGTH MINIMIZATION

Let us consider a dependency structure as a directed acyclic graph, without any inherent ordering of the vertices (we will call this an unordered dependency graph, or UDG). A sentence in a language takes such a graph and “linearizes” it in some way, assigning an order to the vertices. Any linearization of a UDG results in a certain dependency length – where the length of a dependency is defined as the number of words spanned (a dependency between adjacent words has a length of one), and the total dependency length for a sentence is the sum of the lengths of all of its dependencies. The question is; what strategies or principles of ordering minimize dependency length? (We will assume the usual restrictions, discussed earlier, that dependencies may not cross and may not pass over the root word.) In what follows, we consider three principles that appear to contribute to dependency-length minimization. For each principle, we show informally how it reduces dependency length, and also consider evidence for the principle in the grammar and usage of natural languages.

Consider first a situation where each word in a sentence has one dependent (except for one word which has no dependents). In that case, it can be seen that dependency length will be minimized if the head of each dependency is on the same side – either on the left, as in (2a) below, or on the right, as in (2b). A situation such as (2a) could be described as head-first or right-branching; a situation such as (2b) is head-last or left-branching. (A word is right-branching if it is to the right of its head, and a subgraph is right-branching if its head is right-branching.) If some words are right-branching and others are left-branching, as in (2c), that results in longer dependencies. (In all diagrams, dependencies are indicated by arrows pointing from the head to the dependent.)

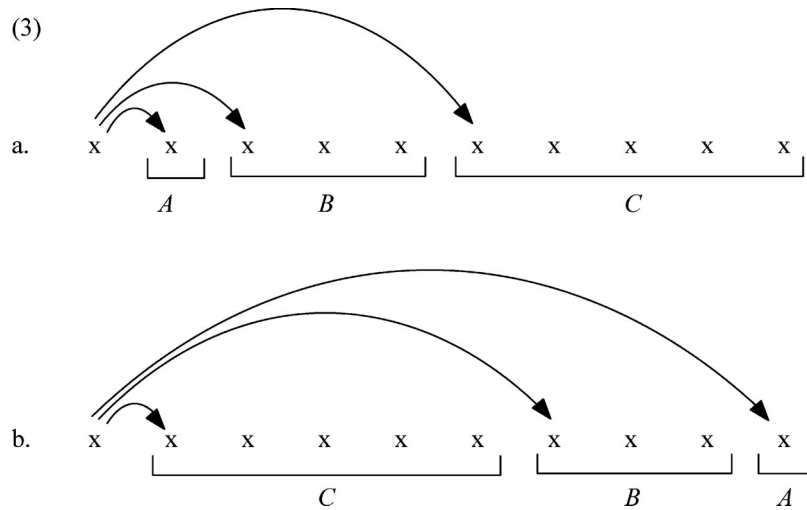


This suggests that a grammar that is consistently right-branching or left-branching – we will call this a “same-branching” grammar – will result in shorter dependencies than one with “mixed” branching. This leads to our first dependency-length minimization rule (DLMR):

DLMR 1 (same-branching rule). Prefer structures that are consistently left-branching or consistently right-branching.

In research on regularities or universals of grammar, one of the most often-cited principles is that languages have a tendency to be either consistently “head-first” or consistently “head-last” (Lehmann, 1973; Venneman, 1974; Hawkins, 1994; Radford, 1997). For example, languages in which the object follows the verb (sometimes known as “VO” languages) are predominantly prepositional languages – that is, just as verbs precede their objects, adpositions precede their objects. (“Adposition” is the general name for prepositions, which precede their objects, and postpositions, which follow them.) Languages in which the object precedes the verb (“OV” languages) tend to be postpositional. Similarly, in VO languages, genitive and adjective modifiers tend to follow nouns, while in OV languages they precede them. (Examples of “head-first” languages are English and the Romance languages; examples of “head-last” languages are Japanese and Turkish.) Several authors have observed, also, that a consistently head-first or head-last grammar might serve to minimize the distances between heads and dependents (Frazier, 1985; Rijkhoff, 1990; Hawkins, 1994).

Let us now consider the situation where a head has multiple dependents. DLMR 1 states that all the dependents of a head should be on the same side, but it does not specify their relative closeness to the head – what we will call the issue of “nesting”. It seems clear that dependency length will be minimized if the dependents with shorter subgraphs are placed closer to the head; we will call this “ordered nesting”. (The length of a subgraph is defined as the number of words it contains, including its head.) For example, consider a UDG in which a word w has three dependent phrases, A , B , and C , with lengths 1, 3, and 5, respectively. Let us assume for the moment that all three of the dependents are on the right side. Ordering them as in (3a) below, in accordance with ordered nesting, yields a dependency length of 8, while ordering them as in (3b) yields a length of 16. (We disregard dependencies within the phrases, which will be the same under any ordering.)



This leads to our second rule:

DLMR 2 (ordered nesting rule). In cases where a head has multiple dependents, the shorter dependent phrases should be located closer to the head.

One might wonder if the advantage of ordered nesting depends on the position of the heads within the dependent phrases. It can be shown that

it does not. Consider a head h with two right-branching dependent phrases $P1$ and $P2$, headed by $w1$ and $w2$ (Figure 1); we will show that the total length of the two dependencies $D1$ (from h to $w1$) and $D2$ (from h to $w2$) will always be less if the shorter phrase is closer to the head, regardless of the positions of $w1$ and $w2$ within $P1$ and $P2$. Within each dependency, we define an “internal segment” IS from the head of the dependent phrase to its leftmost word, and an “external segment” ES from h to the leftmost word in the leftmost of the two dependent phrases. If $P1$ is placed closer to h (as in Figure 1) – we will call this the $(P1, P2)$ ordering – the combined length of $D1$ and $D2$ is $(ES1 + IS1) + (ES2 + length(P1) + IS2)$; under the $(P2, P1)$ ordering, the combined length is $(ES2 + IS2) + (ES1 + length(P2) + IS1)$. The two expressions are the same, except that the expression for the $(P1, P2)$ ordering includes the term $length(P1)$ and the expression for $(P2, P1)$ includes the term $length(P2)$; thus the total length for $(P1, P2)$ will be greater if and only if $P1$ is longer. (If the two phrases are equal in length, the two orderings yield equal dependency lengths.) This logic can be extended to the case where a head has more than two dependent phrases. If some ordering of the phrases violates ordered nesting, so that the length of the phrases is not monotonically increasing with distance from the head, this means that in at least one case there will be two adjacent phrases where $P1$ is closer to the head but longer than $P2$, and dependency length will be reduced if they are swapped.

As with DLMR 1, there is abundant support for DLMR 2 as an important principle in natural languages. Hawkins (1994, 2004) shows that many cross-linguistic grammatical patterns reflect ordered nesting, such as the fact that, in cases where a noun is modified by an adjective and a relative clause, the adjective (which is presumably generally shorter

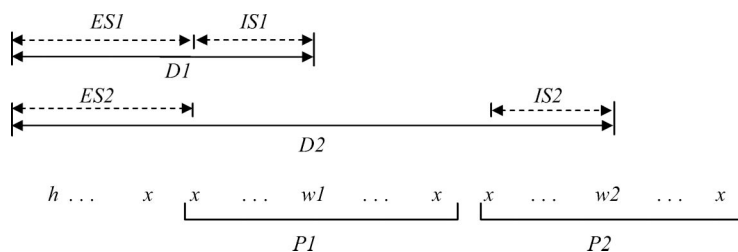


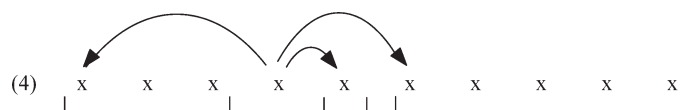
Fig. 1. Dependency lengths for a construction with a head and two right-branching dependent phrases.

than the relative clause) is usually required to be closer to the noun. Further evidence for ordered nesting comes from situations of syntactic choice – where there are multiple correct ways of saying the same thing. Hawkins shows, through a series of corpus analyses, that when there is more than one possible ordering of constituents, the ordering following DLMR 2 – with shorter subgraphs closer to the head – tends to be preferred. For example, in cases where a verb has two prepositional-phrase dependents, the shorter one tends to be placed closer to the verb. This is found both in head-first languages like English, where PPs follow verbs and the shorter of two PPs tends to be placed first, and in head-last languages like Japanese, where PPs precede verbs and the shorter one tends to be placed second.⁵ Hoffman (1999) gives further support for the “short-long” pattern of PP ordering in English, and Köhler (1999) shows that longer constituents in English generally tend to occur later in the mother constituent, as is predicted by DLMR 2. A similar phenomenon is so-called “heavy-NP shift” in English: when a verb has both a direct-object NP and a prepositional phrase, the NP tends to be “extraposed” – placed after the PP – only when it is much longer than the PP (Hawkins, 1994; Wasow, 1997).

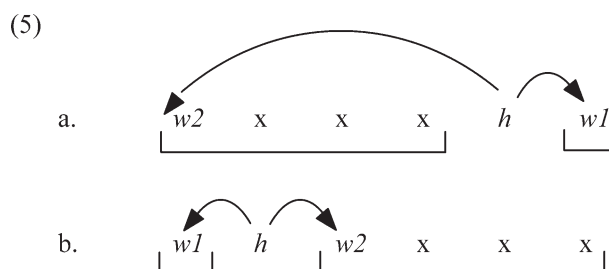
In (3) above, the three dependent constituents are all placed to the right of the head, following DLMR 1. However, it can be seen that this does not, in fact, minimize the total dependency length. Informally speaking, the dependents get in the way of each other, so that the distance from the head to the furthest dependent is the sum of the lengths of all other dependent phrases. While this is unavoidable to a certain

⁵Hawkins’ research brings together a wide variety of phenomena relating to dependency-length minimization, and the current study draws heavily on his findings. Hawkins’ EIC (Early Immediate Constituent) theory argues that language processing is facilitated if the heads of the children within each constituent are clustered together within a short “window”, known as the “constituent recognition domain”; this is advantageous as it provides the parser with “earlier and more rapid access” to the children of the larger constituent (1994, p. 66). While this theory is clearly related to dependency-length minimization, it is not quite the same, and in some cases the two theories make different predictions. For example, in cases where a word has three dependent phrases on the same side, Hawkins’ EIC principle predicts only that the longest phrase will be furthest from the head; it predicts no ordering preference between the shorter two phrases, as the constituent recognition domain will be the same size in either case. By contrast, the dependency-length view predicts that the shortest phrase will be closest to the head. A study of this situation in the case of verbs with three adjunct phrases supports the dependency-length view (Temperley, 2007).

extent, it can be alleviated somewhat by balancing the dependents more evenly on either side of the head, as in (4) below. The result is a lower dependency length: in (3a), the total distance is 8, while in (4) it is only 6.



This suggests that a completely “same-branching” language is, in fact, not optimal from the point of view of dependency length: Lower dependency lengths can be achieved if there is some “mixed branching”. Let us assume that each language has a predominant branching direction – either right-branching or left-branching; the question is, what kinds of dependent phrases should branch in the opposite direction? Consider a situation where a word has two dependents, w_1 and w_2 , with phrases of lengths 1 and 4, respectively. Let us also assume that the language is predominantly right-branching, and that the dependent(s) of w_2 are right-branching, so that w_2 is at the left end of its phrase. When w_2 is left-branching and w_1 is right-branching, as in (5a) below, the total dependency length is 5; when w_1 is left-branching and w_2 is right-branching, as in (5b), it is only 2.



It appears, then, that “opposite-branching” is most advantageous when the opposite-branching dependent phrases are one word long. One might also express this as a preference for opposite-branching phrases to be short; it can be seen from (5) above that opposite-branching phrases will become gradually less desirable as they get longer. This formulation of the idea has been proposed by Hawkins (1994). Hawkins observes

that, in a primarily right-branching language, a long left-branching phrase will tend to separate the head of the dependent phrase from the parent head. Thus long opposite-branching phrases should be avoided; with short phrases, however, opposite-branching carries little or no disadvantage in relation to same-branching. (Hawkins fails to note, however, that opposite-branching of short phrases may actually be *advantageous*, as we have shown above.) While one could certainly express the opposite-branching preference in terms of “short” phrases, we will retain the “one-word” version of the principle here, for reasons that will become clear below.

Two further points should be emphasized about the “opposite-branching one-word” preference. First, the preference applies only in cases where the parent head has multiple dependents. If a parent head h has only a single dependent w , it is clearly better for w to branch in the same direction as h , even if w 's phrase is only one word long. Secondly, even in the case where a word has multiple dependents, we have not shown that one-word dependents should *always* be opposite-branching; we have only shown that, in some cases, a structure with some opposite-branching of one-word constituents yields shorter dependency length than an entirely same-branching structure. We express all this in our third DLMR.

DLMR 3 (limited mixed-branching rule). Some opposite-branching of dependent phrases is desirable, in cases where the dependent phrases are one word long and the parent head has multiple dependents.

Admittedly this rule is somewhat vague; later we will consider ways of expressing it more precisely.

Empirical support for DLMR 3 is found in the work of Dryer (1992), who presents a study of word order patterns in 625 languages. Dryer examines the validity of the proposal that languages are consistently head-first or head-last. He finds that this generalization is borne out in some cases but not others; for example, V-PP, P-NP, and copula-predicate pairs tend to position consistently across a language (in either a head-first or head-last fashion), but noun-adjective, demonstrative-noun, and intensifier-adjective pairs do not reflect such consistent positioning. Dryer proposes an explanation for this data, the “branching direction theory” (BDT): Head-dependent pairs tend to pattern consistently (following the prevailing branching direction of the language) only in

cases where the dependent is phrasal.⁶ That is to say, in a right-branching language, phrasal dependents will always be right-branching, but one-word dependents may not be. Note that the BDT does not predict the branching of non-phrasal dependents to be consistently *opposite* to phrasal dependents; rather, the branching of one-word dependents is predicted to be inconsistent.

As an example, let us consider how well English conforms to Dryer's rule. English is usually considered a "head-first" language, and indeed, phrasal dependents generally follow their heads. Most kinds of phrasal dependents of verbs are normally right-branching, including participle phrases, object NPs, prepositional phrases, infinitival and sentential complements, and subordinate clauses. (One exception is subject NPs, which generally precede verbs; prepositional phrases and subordinate clauses also sometimes occur sentence-initially.) Phrasal modifiers of nouns such as relative clauses, prepositional phrases, and appositives are also right-branching. Prepositional objects follow prepositions, dependent clauses follow conjunctions (or complementizers or relative pronouns), and adjectival complements such as infinitival phrases (e.g. *eager to please*) follow adjectives. Left-branching elements in English include determiners, attributive adjectives, and noun modifiers (all modifying nouns), and intensifiers modifying adjectives (*very big*); all of these are typically one-word units. As Dryer discusses, the distinction between phrasal and non-phrasal dependents is sometimes problematic: "phrasal" dependents such as NPs are sometimes one word in length, and "non-phrasal" dependents like attributive adjectives may sometimes have multiple words (*the very big dog*). On the whole, however, English seems to offer considerable support for Dryer's theory.

It can be seen that Dryer's theory corresponds closely with our DLMR 3. Essentially, Dryer's theory says that one-word dependents sometimes branch opposite to the predominant direction of the language. DLMR 3 says that one-word dependents should sometimes branch opposite to the predominant direction, in cases where the parent head has multiple dependents. The main difference is that in our rule, opposite-branching is

⁶The theory presented here is the "alternate version" of the BDT (Dryer, 1992, p. 116). An earlier version of the theory is phrased not in terms of heads and dependents but in terms of constituent structure, stating that a pair of constituents X and Y will pattern consistently only if X is non-phrasal and Y is phrasal (1992, p. 89). Dryer concludes that the alternate version of the theory is more elegant, but notes that it relies on assumptions about head-dependent relationships that are in some cases controversial.

conditioned on the parent head having multiple dependents, while in Dryer's rule it is not.

The connection between Dryer's BDT and dependency-length minimization has also been noted by Hawkins (1994). As mentioned earlier, Hawkins observes that an avoidance of long opposite-branching constituents will tend to minimize dependency length, and he notes that this accords well with Dryer's observation about phrasal and non-phrasal dependents. Hawkins also points to a phenomenon that offers striking confirmation for Dryer's theory: In some cases, a certain constituent type is opposite-branching when it is non-phrasal and same-branching when it is phrasal. For example, in English, attributive adjectives generally precede the noun, as in *a yellow book*, but when they are phrasal – containing a complement such as a prepositional phrase or infinitival phrase – they must follow the noun, as in *a book yellow with age* (this phenomenon is also mentioned by Dryer). Adverbs show a similar pattern: Single-word adverbial phrases often precede the verb, as in (6a), but adverbs with phrasal complements must follow it, as in (6b).

- (6a) He [rapidly] recovered his form.
 (6b) He recovered his form [rapidly for an old man].

Thus there is considerable evidence for opposite-branching of one-word (or short) constituents in natural languages, and there is reason to believe that this pattern may be advantageous for dependency-length minimization.

TESTING THE PRINCIPLES

We have noted that same-branching structures yield shorter dependency length than mixed-branching structures, that dependency length will be minimized if shorter dependent phrases are closer to the head, and that some opposite-branching placement of one-word dependents is desirable. And we have found support for each of these principles in studies of grammar and language usage. However, our discussion so far has been very informal and qualitative. One might wonder how much benefit each of these principles has in quantitative terms. This question is of particular interest with regard to DLMRs 1 and 3, since they are in a sense contradictory: DLMR 1 states that a “same-branching”

grammar is optimal, while DLMR 3 states that in some cases it is not. This also relates to the plausibility of dependency-length minimization as an explanation for linguistic phenomena. If the effect of (for example) the ordered-nesting rule on dependency length turns out to be very small, then it is not very convincing to argue that the preference for ordered nesting in natural languages is motivated by dependency-length minimization; if the effect is very large, the argument becomes more compelling.

In this section I present a quantitative exploration of the dependency-length minimization principles proposed above. Our investigation hinges on the idea – presented earlier – that a dependency tree can be regarded as an essentially topological structure, a directed acyclic graph, which has been linearized in a particular way. Each DLMR can be viewed as a strategy – or part of a strategy – for linearizing UDGs. The question is, given some set of UDGs characteristic of natural language, what effect does each principle have in reducing dependency length – in comparison, for example, to a random linearization?

The first step is to obtain some unordered dependency graphs. In the current study, we use data from the *Wall Street Journal* portion of the Penn Treebank (hereafter the “WSJ corpus”; see Marcus et al., 1994). The tests reported below use a test set of 1915 sentences, taken from section 00 of the WSJ corpus.⁷ The corpus does not explicitly indicate dependencies; rather, the data is annotated with conventional syntactic phrase markers, indicating NPs, VPs, Ss, and the like; terminal elements (words) are also marked with part-of-speech tags (NN for singular noun, IN for preposition, and so on). To recover the dependencies from this data, we use an algorithm proposed by Collins (1999) for identifying dependencies from Penn Treebank data. The algorithm has rules for choosing a head from the children of each constituent (where children may be either constituents or terminal elements); see Table 1. These rules can be applied recursively in a “bottom-up” fashion to convert the constituent representation into a dependency representation. Dependency trees were extracted from the WSJ corpus using this method, and were then treated as unordered graphs, to be linearized in different ways. Non-lexical punctuation symbols like commas and periods were not included in the dependency trees, due to the lack of consensus as to their

⁷The test set included all sentences in section 00 of the treebank containing 100 words or less. One sentence in section 00 was excluded for this reason.

Table 1. Collins's head-finding rules.*

Parent category	Direction to search	Head categories
ADJP	Left	NNS QP NN \$ ADVP JJ VBN VBG ADJP JJR NP JJS DT FW RBR RBS SBAR RB
ADVP	Right	RB RBR RBS FW ADVP TO CD JJR JJ IN NP JJS NN
CONJP	Right	CC RB IN
FRAG	Right	–
INTJ	Left	–
LST	Right	LS :
NAC	Left	NN NNS NNP NNPS NP NAC EX \$ CD QP PRP VBG JJ JJS JJR ADJP FW
PP	Right	IN TO VBG VBN RP FW
PRN	Left	–
PRT	Right	RP
QP	Left	\$ IN NNS NN JJ RB DT CD NCD QP JJR JJS
RRC	Right	VP NP ADVP ADJP PP
S	Left	TO IN VP S SBAR ADJP UCP NP
SBAR	Left	WHNP WHPP WHADVP WHADJP IN DT S SQ SINV SBAR FRAG
SBARQ	Left	SQ S SINV SBARQ FRAG
SINV	Left	VBZ VBD VBP VB MD VP S SINV ADJP NP
SQ	Left	VBZ VBD VBP VB MD VP SQ
UCP	Right	–
VP	Left	TO VBD VBN MD VBZ VB VBG VBP VP ADJP NN NNS NP
WHADJP	Left	CC WRB JJ ADJP
WHADVP	Right	CC WRB
WHNP	Left	WDT WP WPS WHADJP WHPP WHNP
WHPP	Right	IN TO FW

*These are the rules used in the head-finding algorithm of Collins (1999). The left column shows the parent category. The middle column indicates the direction from which one should start looking for a head (left or right). The right column lists the head categories in order of preference. If none of the head categories is found (or if the table indicates no preferred head category), the first child found (searching from the preferred direction) is chosen. The head of an NP is chosen by a complex rule not shown here.

dependency status. Excluding punctuation, the average length of sentences in the test set was 21.2 words. Figure 2 shows the first sentence from the test set in Penn Treebank notation and the dependency tree produced by Collins's algorithm.

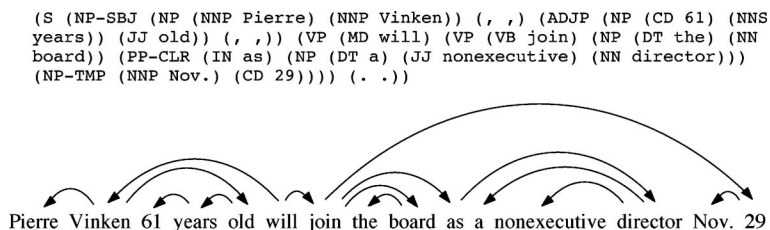


Fig. 2. The first sentence of the WSJ corpus, in its original Penn Treebank notation (above) and as analysed by Collins' head-finding algorithm, with punctuation removed (below).

In what follows, we examine various different linearization algorithms (or “grammars”) and the dependency length that results when each one is applied to the UDGs of the test set. As can be seen from our earlier discussion, the linearization of a UDG really involves two separate issues: branching (the direction of each dependency in relation to the head) and nesting (the ordering of dependent phrases on the same side of a head). Once the branching and nesting of all dependents is determined in relation to their heads, the linearization of the UDG is completely determined. All of the grammars defined below assume the usual restrictions on dependency structures: dependencies may not cross and may not pass above the root word.

We begin by considering two simple grammars. The first is completely random; the second implements “ordered nesting”.

Grammar 1. All dependents branch randomly, and are randomly nested.

In other words, for each child, a random decision is made as to whether it will be right-branching or left-branching in relation to the head; the dependents on the same side of each head are then randomly ordered in closeness to the head.

Grammar 2. All dependents branch randomly. Nesting is ordered: that is, shorter dependents are placed closer to the head.

On the WSJ test set, grammar 1 yields an average (per-sentence) dependency length (ADL) of 75.69; grammar 2 yields an ADL of 62.47. The ADL for grammar 2 is significantly lower ($t(1914)=21.87$,

$p < 0.0001$), yielding a reduction of 17.5% in relation to grammar 1. (All comparisons reported here are two-tailed paired-sample t -tests.) These and other test results are shown in Table 2. Thus, we can see that ordered nesting confers a considerable benefit in reducing dependency length.

We now consider the issue of branching. Grammars 3 and 4 implement completely “same-branching” grammars, with random nesting and ordered nesting, respectively.

Grammar 3. All dependents branch to the right. Nesting is random.

Grammar 4. All dependents branch to the right. Nesting is ordered.

Grammar 3 can be compared with grammar 1; both use random nesting, but grammar 3 has consistent right-branching while grammar

Table 2. Average dependency length (ADL) on the UDGs of Section 00 of the WSJ Corpus for various grammars.

Grammar	Description	ADL
Grammar 1	Random branching, random nesting	75.69
Grammar 2	Random branching, ordered nesting	62.47
Grammar 3	Same-branching, random nesting	69.62
Grammar 4	Same-branching, ordered nesting	43.42
Grammar 5	All one-word dependencies opposite-branching; ordered nesting	46.46
Grammar 6	All one-word dependencies of multi-dependent heads opposite- branching; ordered nesting	44.04
Grammar 7	One one-word dependency of multi-dependent heads opposite-branching; random nesting	56.25
Grammar 8	One one-word dependency of multi-dependent heads opposite-branching; ordered nesting	39.87
Grammar 9	Original branching, original nesting (English)	46.96
Grammar 10	Original branching, ordered nesting	44.17
Grammar 11	Original branching, random nesting	53.11
Grammar 12	Alternate branching, ordered nesting	33.28
–	Minimum possible ADL found through exhaustive search	33.28

1 had random branching. Grammar 3 shows a fairly modest reduction in dependency length over grammar 1 (69.62 vs. 75.69, or a reduction of 8.0%, $t(1914)=8.93$, $p < 0.0001$). Now compare grammar 4 to grammar 2; both have ordered nesting, but grammar 2 has random branching while grammar 4 has right branching. This time the advantage of the right-branching grammar is considerably larger, 43.42 vs. 62.47 (a reduction of 30.5%, $t(1914)=30.46$, $p < 0.0001$). Compared to a grammar with random nesting and random branching (grammar 1), grammar 4 (incorporating both ordered nesting and same-branching) yields a reduction of 42.6%, $t(1914)=35.93$, $p < 0.0001$. Thus DLMR 1 has a significant effect on dependency-length minimization, particularly when ordered nesting is also used. (A grammar that is completely left-branching instead of right-branching, but otherwise equivalent, achieves the same ADL as the corresponding right-branching grammar.)

It was suggested earlier that a certain degree of “mixed branching” may be advantageous. In particular, it seems desirable for some one-word dependent phrases to be opposite-branching, especially when the parent head has multiple children. Let us first consider the case where *all* one-word children are opposite-branching. (We will assume ordered nesting for now.) Will this be better or worse than the “same-branching” grammar (grammar 4)?

Grammar 5. All multi-word dependents branch to the right; all one-word dependents branch to the left. Nesting is ordered.

Grammar 5 yields an ADL of 46.46 – somewhat worse than the result for Grammar 4 (43.42), $t(1914)=40.89$, $p < 0.0001$. Thus, a grammar in which all one-word dependent phrases are opposite-branching is not superior to a same-branching grammar with regard to dependency length. Next we consider a grammar in which one-word dependent phrases are opposite-branching, but only if the head has multiple children.

Grammar 6. All multi-word dependents branch to the right. One-word dependents branch to the left, but only if the head has multiple children. Nesting is ordered.

The resulting ADL is 44.04; this is still marginally worse than that of the same-branching grammar (grammar 4), $t(1914)=7.74$, $p < 0.0001$.

Inspection of the results shows that, in many cases, a word has several one-word dependents and no multi-word dependents; to apply opposite-branching for all of the dependents creates a very “unbalanced” configuration. It would be better if only some of the one-word dependents were opposite-branching. One way to enforce this is to insist that only one dependent of a head may be opposite-branching. The following two grammars apply this principle; they are identical except that one has random nesting (grammar 7) and the other has ordered nesting (grammar 8).

Grammar 7. All multi-word dependents branch to the right. All one-word dependents also branch to the right; EXCEPT: If a head has multiple dependents including one or more one-word dependents, then exactly one one-word dependent branches to the left. Nesting is random.

Grammar 8. All multi-word dependents branch to the right. All one-word dependents also branch to the right; EXCEPT: If a head has multiple dependents including one or more one-word dependents, then exactly one one-word dependent branches to the left. Nesting is ordered.

Since Grammars 7 and 8 are essentially right-branching grammars with an exception for certain one-word phrases, it is appropriate to compare them to right-branching grammars. Grammar 7, a random-nesting mixed-branching grammar, proves to be much better than grammar 3, the random-nesting right-branching grammar (56.25 vs. 69.62, a reduction of 19.2%, $t(1914) = 39.48$, $p < 0.0001$). And grammar 8, an ordered-nesting mixed-branching grammar, is somewhat better than grammar 4, the ordered-nesting right-branching grammar (39.87 vs. 43.42, a reduction of 8.2%, $t(1914) = 56.12$, $p < 0.0001$). This shows, then, that a grammar with a limited amount of mixed branching can yield significantly lower dependency length than a same-branching grammar.

Our investigation has shown that all three of our DLMRs contribute to the reduction of dependency length. With regard to DLMR 1, a same-branching grammar achieves much lower dependency lengths than one with random branching (especially if ordered nesting is also assumed). With regard to DLMR 2, the use of ordered nesting achieves a significant

reduction in comparison to random nesting. And with regard to DLMR 3, a grammar in which one one-word dependent of heads with multiple children is assigned opposite branching achieves an improvement over a same-branching grammar (both with ordered nesting and without).

We should emphasize that there are really two aspects of language at issue here. One is grammar – hard-and-fast rules governing the syntactic structures of a language. The other is syntactic choice – choices between alternative constructions that are more or less equivalent in meaning. (In terms of theoretical linguistics, one might characterize this as a distinction between “competence” and “performance.”) As noted earlier, dependency-length minimization has been cited as a shaping factor with regard to both grammar and syntactic choice. From the point of view of our computational tests, this distinction is not crucial. Our concern has only been to show that our principles significantly reduce dependency length, and might therefore be plausibly invoked to explain phenomena of either competence or performance.

Perhaps the most problematic of the three rules is DLMR 3. While there clearly seems to be some benefit to limited mixed branching, it is unclear exactly how this preference should be stated. The main support for the rule in linguistic research comes from Dryer’s observation that one-word phrases are sometimes opposite-branching. However, this phenomenon is difficult to capture using the abstract dependency grammars proposed above. Our grammar 8 (and grammar 7) offered one specification of the rule – in which one one-word phrase is opposite-branching when the head has multiple dependents. But in reality, of course, the branching of words depends not only on abstract dependency structures – as in our grammars above – but on the specific syntactic categories of the words and phrases. For our grammar 8 to correspond to an actual language, there would have to be some category X that fulfilled the grammar’s requirements for opposite-branching phrases: X must always be exactly 1 word long, there must never be more than one X modifying a particular word, and the parent head of X must always have multiple dependents. Such a category seems unlikely to exist in any real language. Thus, our grammar 8 is only a crude approximation to linguistic reality. Still, the success of grammar 8 shows that, under certain conditions, opposite branching of some one-word phrases can significantly reduce dependency length; and it strengthens the argument that the opposite branching of one-word dependents observed by Dryer might be due to pressures of dependency-length minimization.

DEPENDENCY STRUCTURES IN ENGLISH

It is of interest to consider the average dependency length for the actual linearizations of the sentences in the WSJ test set. This was determined by extracting the dependency trees using Collins' algorithm, but this time retaining the original order of the words, and using this to calculate the total dependency length. For the "original-branching/original-nesting" grammar (grammar 9 in Table 2), the ADL is 46.96. In a further experiment, the original linearizations were extracted but were then subjected to "ordered nesting" – arranging dependent phrases in increasing order of size on each side of a head, but retaining the original branching. The resulting ADL for this "original-branching/ordered-nesting" grammar (grammar 10 in Table 2) was 44.17. It is of particular interest to compare grammar 10 with grammar 4 above – a entirely "right-branching" grammar with ordered nesting, which yielded an ADL of 43.42; the difference in ADL between the two is very small (though still significant, $t(1914)=4.57$, $p < 0.0001$). Since both grammar 4 and grammar 10 feature ordered nesting, the only difference is that grammar 10 features original branching while grammar 4 has all right-branching.

As noted earlier, English features a significant amount of left branching; this includes not only many one-word dependents, but also some phrasal dependents such as subject NPs and sentence-initial subordinate clauses. We can use Collins's algorithm to explore this further. We find that, out of 38,749 dependent phrases in section 00 of the WSJ corpus, 18,137 (46.8%) are left-branching; however, 14,418 (79.4%) of the left-branching phrases are one-word phrases, whereas only 3973 (19.4%) of the right-branching phrases are one-word phrases. For now, the important point is that almost half of the dependent phrases in English are left-branching; and yet, once ordered nesting is imposed, English (grammar 10) achieves almost the same ADL as an entirely right-branching grammar (grammar 4). This is further proof that a grammar with a good deal of opposite branching can be at least competitive with (if not superior to) an entirely same-branching grammar.

One might wonder, also, to what degree English reflects ordered nesting. If we compare grammar 9 (the original structures found in the WSJ corpus) with grammar 10 (the original structures except with ordered nesting), we find that the ADL of grammar 9 (46.96) is only slightly higher than that of grammar 10 (44.17), $t(1914)=23.72$, $p < 0.0001$. We could also consider the original branching with *random*

nesting (grammar 11); this yields an ADL of 53.11. With regard to ADL, grammar 9 (original nesting) is much closer to grammar 10 (ordered nesting) than to grammar 11 (random nesting). This suggests that English generally reflects ordered nesting, though not completely. We can also measure the adherence to ordered nesting more directly from the WSJ data itself. This could be done in various ways. One straightforward method is to examine cases where a word has exactly two dependent phrases on the same side that differ in length, counting the proportion of cases in which the shorter phrase is closer to the head. Each word is thus examined twice, once for its left side and once for its right side. In section 00 of the WSJ corpus, there are 2679 eligible “word-sides” (cases where a word has exactly two dependents of different length on the same side); 2175 of these, or 81.2%, have ordered nesting.

THE OPTIMAL DEPENDENCY GRAMMAR

One question that arises here is: What is the *optimal* grammar in terms of dependency-length minimization (assuming abstract dependency grammars of the type presented earlier)? The best we have presented so far is grammar 8, which achieves an average dependency length of 39.87 for the UDGs in the WSJ test set. But can we do better? The answer is “yes”, although this will take us even further from linguistic reality towards a purely mathematical realm.

It was suggested earlier that it seems optimal for the dependent phrases of a head to be roughly evenly balanced on either side of the head. One way to achieve this would be to stipulate that the longest dependent phrase of a head word should branch in the same direction as the head itself; the second longest should be opposite-branching (in relation to the head); the third should be same-branching; and so on. In such an “alternate-branching” grammar, there would always be roughly the same number of dependents on either side of the head. As noted earlier, in a predominantly “same-branching” grammar, this balancing of dependents tends to cause long opposite-branching dependent phrases in which the head of the dependent is far from the parent head, as in (6b). It was for this reason that we suggested that opposite-branching phrases should be short. Another solution, however, would be to arrange each opposite-branching phrase so that its head (call this word w) is close to the parent head. If we recursively repeat the process described above – the longest

dependent of w is same-branching (in relation to w), the second longest is opposite-branching, and so on – this will tend to make w relatively close to its head. We capture this in our final grammar:

Grammar 12. For each head, branching decisions for the dependent phrases are made in decreasing order of length. The longest dependent branches in the same direction as the head; subsequent dependent phrases then branch on alternate sides. Nesting is ordered.

On the WSJ test set, this “alternate-branching” grammar yields an ADL of 33.28. This is substantially better than any other grammar so far considered, and significantly better than the next-best grammar, grammar 8; $t(1914) = 35.96$, $p < 0.0001$. Compared to the completely random grammar (grammar 1), grammar 12 achieves a reduction of 56.0%.

Grammar 12 appears to bear less resemblance to a natural grammar than any of the others discussed so far. Perhaps its most unnatural aspect is that the branching of dependents depends on their rank order in length in relation to other dependents. This means that, for example, a particular phrase might branch to the right if it happened to be the third-longest dependent phrase, but to the left if it was the fourth-longest. This is quite unlike any naturally-occurring grammatical rule, as far as I know. Figure 3 shows grammar 12’s linearization of the first sentence in the WSJ corpus. (The grammar arbitrarily assumes that the longest dependent of the root word branches to the right.) As an illustration of the grammar’s logic, consider the word *Vinken*. This has two dependents, the word *Pierre* and the phrase *61 years old*. Because *Vinken* is left-branching, its longest dependent phrase, *61 years old*, is also left-branching; its second-longest dependent, *Pierre*, is then right-branching. As another example, note that the article *a* is left-branching. This is because its head *director* has two one-word dependents, the article *a* is arbitrarily considered to be the second-longest, and it is therefore opposite-branching in relation to its head (which is right-branching).

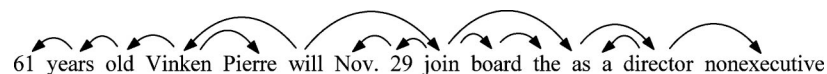


Fig. 3. The first sentence of the WSJ corpus (see Figure 2) as linearised by grammar 12.

By contrast, the article *the* is right-branching, as it is the only dependent of a right-branching noun. On this sentence, grammar 12 achieves a total dependency length of 20; in comparison, the original linearization (shown in Figure 2) yields a dependency length of 32.

It is notable that grammar 12 makes no reference to the idea of opposite-branching one-word phrases. This idea is cited as an important linguistic principle by Dryer, and we found it to be an effective means of dependency-length reduction (as in our grammar 8); yet grammar 12 achieves a much lower ADL than grammar 8 without using this strategy. It appears that the one-word opposite-branching principle is a way of achieving a substantial reduction in dependency length within the other constraints governing dependency structures. As noted above, grammatical rules that condition the branching direction of a dependency on the branching direction of the head word or other dependent phrases seem highly unnatural, and may simply be incompatible with human cognitive capacities. However, a grammar that assigns opposite branching to certain one-word (or short) syntactic categories *is* compatible with human capacities; and such grammars allow a significant reduction in dependency length, though they do not achieve the absolute optimum in this regard.

Is grammar 12 the optimal dependency grammar? An algorithm was devised for searching the entire space of linearizations of a UDG and finding the best one. (The algorithm, which will not be described in detail here, uses dynamic programming and has complexity $O(n^3)$.) Using this algorithm, the best linearization was computed for each of the UDGs in the test set. The resulting ADL is 33.28 – exactly the same value as for grammar 12. This seems to warrant the conjecture that grammar 12 is the optimal dependency grammar.

CONCLUSIONS

Dependency-length minimization provides an explanation for a wide range of phenomena from diverse areas of linguistic research, including psycholinguistics (phenomena of comprehension and ambiguity resolution), corpus linguistics (patterns of syntactic choice), linguistic typology (cross-linguistic word-order regularities), and computational linguistics (the efficacy of “distance” heuristics in parsing). This study gathers evidence relating to two well-established principles of dependency-length

minimization – the “same-branching” principle and the “ordered-nesting” principle – and adds a third, the principle that dependency length is reduced if some one-word dependents of heads with multiple dependents are opposite-branching. A computational study in which unordered dependency graphs were extracted from written English and linearized in different ways showed that all three principles significantly reduce dependency length. In comparison to completely random linearizations, the ordered-nesting principle reduces dependency length by 17.5%, and the same-branching principle yields a reduction of 8.0%; in combination, these two principles yield a reduction of 42.6%. Compared with a right-branching grammar (with random nesting), the principle of “opposite-branching one-word phrases” yields a further reduction of 19.2%. The fact that these three principles reduce dependency length so markedly gives them plausibility as shaping forces in language and linguistic behaviour.

The optimal dependency-length-minimization algorithm appears to be one in which the longest dependent phrase of a head branches in the same direction as the head and successive phrases branch on alternate sides (with ordered nesting). A dynamic-programming procedure for finding the optimal configuration for each UDG in the test set yielded the same average dependency length as this grammar. The fact that natural languages do not exhibit such patterning no doubt reflects constraints on the kinds of grammars that humans can learn and use.

The current study raises a number of further issues; I will mention just two. One concerns the observation that opposite branching of one-word phrases is only desirable when the head word has multiple dependents. Dryer’s (1992) theory offers considerable support for the general principle of “opposite-branching one-word phrases”; but is opposite-branching of one-word phrases more common when the head word has multiple dependents? This is a prediction of the dependency-length minimization theory that invites further testing.

A second issue concerns unordered dependency structures. In the current study, we used written English as a source of UDGs – taking these as representative of languages in general. However, there may well be significant differences in the nature of UDGs across different languages; thus, undertaking tests like the ones presented here with other languages would certainly be worthwhile. Another interesting possibility is that UDGs themselves may reflect pressures of dependency-length minimization. For example, in cases where syntactic rules dictate

that a phrasal dependent must be opposite-branching, people may avoid long dependencies by keeping such phrases fairly short. I have suggested elsewhere (Temperley, 2007) that this may explain the shorter length of subject noun-phrases compared to object noun-phrases in English. The possibility that the structure of UDGs may itself be shaped, in part, by considerations of dependency length would seem to be an interesting area for further study.

REFERENCES

- Abney, S. (1987). *The English noun phrase in its sentential aspect*. PhD dissertation, MIT.
- Behagel, O. (1932). Deutsche Syntax: Eine geschichtliche Darstellung. In *Wortstellung. Periodenbau*. Vol. IV. Heidelberg: Carl Winters.
- Bresnan, J., Kaplan, R. M., Peters, S., & Zaenen, A. (1982). Cross-serial dependencies in Dutch. *Linguistic Inquiry*, 13, 613–635.
- Collins, M. (1999). *Head-driven statistical models for natural language parsing*. PhD dissertation, University of Pennsylvania.
- Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29, 589–638.
- Dryer, M. S. (1992). The Greenbergian word order correlations. *Language*, 68, 81–138.
- Eisner, J., & Smith, N. (2005). Parsing with soft and hard constraints on dependency length. In *Proceedings of the International Workshop on Parsing Technologies*, pp. 30–41.
- Frazier, L. (1985). Syntactic complexity. In D. R. Dowty, L. Karttunen & A. M. Zwicky (Eds), *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives* (pp. 129–189). Cambridge, UK: Cambridge University Press.
- Gaifman, H. (1965). Dependency systems and phrase-structure systems. *Information and Control*, 8, 304–337.
- Gibson, E. (1998). Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68, 1–76.
- Gibson, E. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. In A. Marantz, Y. Miyashita & W. O’Neil (Eds), *Image, Language, Brain* (pp. 95–126). Cambridge, MA: MIT Press.
- Gibson, E., & Pearlmutter, N. (1994). A corpus-based analysis of psycholinguistic constraints on PP attachment. In C. Clifton, L. Frazier & K. Rayner (Eds), *Perspectives on Sentence Processing* (pp. 181–198). Hillsdale, NJ: Erlbaum.
- Hawkins, J. A. (1994). *A Performance Theory of Order and Constituency*. Cambridge, UK: Cambridge University Press.
- Hawkins, J. A. (2004). *Efficiency and Complexity in Grammars*. Oxford, UK: Oxford University Press.
- Hoffman, C. (1999). Word order and the principle of “Early Immediate Constituents” (EIC). *Journal of Quantitative Linguistics*, 6, 108–116.
- Hudson, R. (1990). *English Word Grammar*. Oxford, UK: Blackwell.
- Jackendoff, R. (1977). *X-Bar Syntax: A Study of Phrase Structure*. Cambridge: MIT Press.

- Joshi, A. K. (1990). Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes*, 5, 1–27.
- King, J., & Just, M. A. (1991). Individual differences in syntactic processing: The role of working memory. *Journal of Memory and Language*, 30, 580–602.
- Köhler, R. (1999). Syntactic structures: Properties and interrelations. *Journal of Quantitative Linguistics*, 6, 46–57.
- Lehmann, W. P. (1973). A structural principle of language and its implications. *Language*, 49, 47–66.
- Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Ferguson, M., Katz, K., & Schasberger, B. (1994). The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the ARPA Human Language Technology Workshop*.
- Mel'cuk, I. A. (1987). *Dependency Syntax: Theory and Practice*. Albany: State University of New York Press.
- Munn, A. (1993). *Topics in the syntax and semantics of coordinate structures*. PhD dissertation, University of Maryland.
- Pollard, C., & Sag, I. (1987). *Information-based Syntax and Semantics*. Menlo Park, CA: Center for the Study of Language and Information.
- Radford, A. (1997). *Syntactic Theory and the Structure of English*. Cambridge, UK: Cambridge University Press.
- Rijkhoff, J. (1990). Explaining word order in the noun phrase. *Linguistics*, 28, 5–42.
- Sleator, D., & Temperley, D. (1991). Parsing English with a link grammar. *Carnegie Mellon University Computer Science technical report CMU-CS-91-196*.
- Steedman, M. (1985). Dependency and coordination in Dutch and English. *Language*, 61, 523–68.
- Temperley, D. (2005). The coordinate structure of dependency phrases: A corpus approach. *Journal of Psycholinguistic Research*, 34, 577–601.
- Temperley, D. (2007). Minimization of dependency length in written English. *Cognition*, 105, 300–333.
- Thornton, R., MacDonald, M. C., & Arnold, J. E. (2000). The concomitant effects of phrase length and informational content in sentence comprehension. *Journal of Psycholinguistic Research*, 29, 195–203.
- Vennemann, T. J. (1974). Topics, subjects and word order: From SXV to SVX via TVX. In J. M. Anderson & C. Jones (Eds), *Historical Linguistics I (Proceedings of the First International Conference on Historical Linguistics)*. New York: American Elsevier Publishing Company.
- Wasow, T. (1997). End-weight from the speaker's perspective. *Journal of Psycholinguistic Research*, 26, 347–361.

Copyright of Journal of Quantitative Linguistics is the property of Routledge and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.