Memory and Locality in Natural Language

by

Richard Landy Jones Futrell

B.A., Linguistics, Stanford University (2010)
M.A., Linguistics, Stanford University (2012)

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Cognitive Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Author . Signature redacted .....
Department of Brain and Cognitive Sciences
May 5, 2017

Certified by .. Signature redacted .....
Edward Gibson
Professor
Thesis Supervisor

Signature redacted

Certified by .....
Roger Levy
Associate Professor
Thesis Supervisor

Accepted by . Signature redacted .....
Matthew A. Wilson
Sherman Fairchild Professor of Neuroscience and Picower Scholar
Director of Graduate Education for Brain and Cognitive Sciences

Memory and Locality in Natural Language

by

Richard Landy Jones Futrell

Submitted to the Department of Brain and Cognitive Sciences
on May 5, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Cognitive Science

Abstract

I explore the hypothesis that the universal properties of human languages can be explained in terms of efficient communication given fixed human information processing constraints. I argue that under short-term memory constraints, optimal languages should exhibit **information locality**: words that depend on each other, both in their interpretation and in their statistical distribution, should be close to each other in linear order. The information-theoretic approach to natural language motivates a study of quantitative syntax in Chapter 2, focusing on word order flexibility. In Chapter 3, I show comprehensive corpus evidence from over 40 languages that word order in grammar and usage is shaped by working memory constraints in the form of **dependency locality**: a pressure for syntactically linked words to be close. In Chapter 4, I develop a new formal model of language processing cost, called **noisy-context surprisal**, based on rational inference over noisy memory representations. This model unifies surprisal and memory effects and derives dependency locality effects as a subset of information locality effects. I show that the new processing model also resolves a long-standing paradox in the psycholinguistic literature, structural forgetting, where the effects of memory appear to be language-dependent. In the conclusion I discuss connections to probabilistic grammars, endocentricity, duality of patterning, incremental planning, and deep reinforcement learning.

Thesis Supervisor: Edward Gibson
Title: Professor

Thesis Supervisor: Roger Levy
Title: Associate Professor

Acknowledgments

Thanks are due first to my advisors Ted Gibson and Roger Levy. Having Ted as an advisor for five years has been a rare privilege: he has always encouraged me to seek the most important generalizations and ask the interesing questions, and moreoever he has been there for me as a friend. Roger's great intellectual enthusiasm, authenticity, and conscientiousness have provided direction to my work.

For help on the development of this thesis, thanks are due to my committee chair Josh Tenenbaum, who has always provided provocative and inspiring suggestsions, and committee member Davy Temperley, who provided detailed and diligent feedback. Thanks also to Nancy Kanwisher for her insightful comments in earlier stages of this work.

My ideas have been greatly shaped by my intellectual community here at MIT. In particular I would like to thank Tim O'Donnell and Leon Bergen for inducting me into the probabilistic programming worldview. I would also like to thank members of Tedlab, Evlab, and Rogerlab for making the community what it is: Kyle Mahowald, Evelina Fedorenko, Melissa Kline, Rachel Ryskin, Paula Rubio Fernández, Idan Blank, Meilin Zhan, Mika Braginsky, Alex Paunov, Olessia Jouravlev, Zach Mineroff, Brianna Pritchett, Matt Siegelman, Evgeniia Diachek, Caitlyn Hoeflin, and Peter Graff.

I would also like to thank the professors and intellectual leaders who have supported, inspired, and mentored me along the way, especially Ivan Sag, Dan Jurafsky, Michael Ramscar, Hannah Rohde, Tom Wasow, Gerry Sussman, Tom Gruber, and Chris Potts. My work has also been shaped by conversations and collaborations with many researchers, to whom I owe thanks, especially Nathaniel J. Smith, Christian Bentz, Steve Piantadosi, Celeste Kidd, Yevgeni Berzak, Yonatan Belinkov, Bevil Conway, Adam Albright, Kristina Gulordava, Paola Merlo, Sam Bowman, Joakim Nivre, Melody Dye, Marten van Schijndel, Cory Shain, William Schuler, Damián Blasí, Haitao Liu, Tim Osborne, and Ramon Ferrer i Cancho. Thanks especially to Simon Kirby, Kenny Smith, and Jennifer Culbertson for graciously hosting me at their department for a visit in 2016.

Thanks are also due to those who have provided material support for this research: to the NSF, to Steve Piantadosi and the OpenMind team for letting me use their computer

resources, and to Caroline Singleton for establishing the Henry E. Singleton fellowship which supported me for two years. I would also like to thank the administrative staff in the BCS office for their highly competent assistance, especially Julianne Gale Ormerod and Margarita O'Leary.

Finally I would not have made it this far without the support of my close friends. Thanks especially to Michaela for her love and support. Most importantly I would like to thank my parents for their unconditional love.

This thesis is dedicated to God.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

名不正，則言不順；言不順，則事不成；事不成，則禮樂不興；禮樂不興，則刑罰不中；刑罰不中，則民無所措手足。故君子名之必可言也，言之必可行也。

Confucius, *Analects* 13.3

1.1 Introduction

What kind of thing is natural language? Why are natural languages the way they are? How should we model a natural language mathematically? These questions are crucial for understanding linguistic behavior, as well as for developing technologies that can use natural language.

Here I argue that the distinctive properties of human language can be derived from rational communication among agents with humanlike information processing characteristics, such as incrementality of processing and restrictions on short-term memory and planning capacity. The main result is that under these constraints, natural language should have the property of **information locality**: if elements of an utterance depend on each other, then those elements should be close in linear order. Dependence can take two forms: context-dependent interpretation—where the interpretation of one linguistic element depends on the presence of some other linguistic element—and probabilistic dependence, where one linguistic element makes another more or less likely to occur. I give evidence that information locality holds based on quantitative, information-theoretic analysis of syntax in dependency corpora of many languages.

This work aims to combine cognitive, typological, and computational perspectives in order to explain the universals of human language. From the cognitive side, I use the broad generalizations about human sentence comprehension and production: that it is highly incremental and operates with imperfect memory (Gibson, 1998; Christiansen and Chater, 2016). From the typological side, I use some of the generalizations that have been discovered about languages, especially implicational universals about word order (Greenberg, 1963; Dryer, 2002, 2011; Hawkins, 2004). From the computational side, I use the mathe-

matical language and model of communication from Information Theory (Shannon, 1948; Cover and Thomas, 2006), probabilistic modeling concepts, and richly annotated linguistic resources from the parsing community (Nivre, 2015).

The idea of explaining language universals functionally, in terms of communication efficiency and information processing cost, has a rich history in linguistics (Zipf, 1949; Hockett, 1960; Greenberg, 1966; Slobin, 1973; Comrie, 1981; Bates and MacWhinney, 1989; Givón, 1991, 1992; Dryer, 1992; Hawkins, 1994, 2004, 2014; Dryer, 2006; Croft, 2001, 2003; Haspelmath, 2008; Richie, 2016). This work aims to introduce information locality as a new and multifaceted quantitative generalization into the linguistics discourse.

On the cognitive side, a growing contingent of researchers has been arguing for communicative optimality as an explanatory principle for linguistic phenomena (Ferrer i Cancho and Solé, 2003; Jaeger and Tily, 2011). For example, Regier et al. (2015) argues that linguistic category systems are shaped by simultaneous pressures for communication and simplicity, with evidence from the semantic domains of color (Regier et al., 2007), kinship (Kemp and Regier, 2012), spatial terms (Khetarpal et al., 2013), numeral systems (Xu and Regier, 2014), and others (Xu et al., 2016; Regier et al., 2016). Piantadosi et al. (2011) provide evidence that word lengths are optimized for efficient communication, and Piantadosi et al. (2012) argue that efficiency in communication can explain the presence of ambiguity in natural language (see also Juba et al. (2011) for related arguments). These arguments have typically applied at the level of words and lexicons, rather than syntactic structures (though for an exception see Gildea and Jaeger (2015)). This thesis focuses on the question of how this approach may be used to explain syntax, arguing that syntactic systems conform to locality constraints induced by memory limitations in incremental processing.

Previous work has argued for the pivotal role of incrementality of processing in shaping the organization of linguistic systems (Christiansen and Chater, 2016); this thesis emphasizes locality constraints as a consequence from that hypothesis.

For many years, alongside linguistics and cognitive science, there has existed a sound and actively developed theory of communication, with deep roots in probability theory, in the form of **information theory** (Shannon, 1948; MacKay, 2003; Cover and Thomas, 2006). Information theory has seen great success in theoretical and practical analysis of

digital communication codes, but it has had less success as applied to human languages. This is because many of the properties of information theoretically optimal codes do not seem to hold in natural language (see Section 1.3.5 for detailed discussion).

Here I will argue that the distinctive properties of natural language can be derived using the standard tools of information theory while appropriately factoring in *human-specific information processing constraints*. In particular, I consider the communicative utility of languages given limitations on the representational capacity of short-term memory. I find that under these limitations, optimal languages have the property of information locality: utterance elements that depend on each other, either for their meaning or their distribution, must be close. This thesis is devoted to exploring the idea of information locality from a theoretical perspective and building evidence for it using crosslinguistic studies of quantitative syntax, based on parsed dependency corpora of many languages.

The basic idea of information locality has been proposed in various forms in previous work. It is recognizable as Behaghel (1932)'s fundamental principle of word order: "that which belongs close together mentally is also placed close together."[1] It can be seen as a generalization of the idea that is variously known as **dependency locality** (Gibson, 2000), **domain minimization** (Hawkins, 2004, 2014) or **dependency length minimization** (Ferrer i Cancho, 2006; Liu, 2008; Gildea and Temperley, 2010). Inasmuch as dependency locality is a subset of information locality, information locality can explain pervasive word order universals in language (Hawkins, 2004, 2014; Ferrer i Cancho, 2006; Gildea and Temperley, 2010).

Given this previous work on locality concepts, the contribution of this thesis is threefold. First, I provide a sound mathematical footing for information locality by defining it in terms of mutual information. Second, I derive information locality formally from a utility function for languages that factors in short-term memory constraints (see Section 1.5 and Chapter 4). Third, I provide unprecedented large-scale evidence that languages follow information locality, both in the form that dependency lengths are short (Chapter 3) and that words that covary in general are close (Section 4.4).

I am arguing that natural languages satisfy a notion of communicative optimality, but I

[1] *Das oberste Gesetz ist dieses, daß das geistig eng Zusammengehörige auch eng zusammengestellt wird.*

remain agnostic as to the mechanism by which languages end up becoming more optimal—whether it happens over generational language change or some more dynamic process. This work aims to describe a communication-related utility function for languages that makes interesting predictions about syntax and to show that languages apparently maximize that function. I leave the detailed investigation of *how* that optimization happens to other work. The most common proposals for how languages become more functional include biases that emerge through learning and cultural transmission in conjunction with communication (Kirby, 1999, 2002; Griffiths and Kalish, 2007; Fedzechkina et al., 2012; Culbertson et al., 2012; Kirby et al., 2014), although see Bybee and Slobin (1982) for arguments against the idea that language learners drive language change.

The rest of this introduction will be devoted to developing the idea of the communicative utility of a language and how information processing constraints interact with it. I introduce the conceptual framework for thinking about language as an optimal code in Section 1.2, and introduce mathematical concepts from information theory in Section 1.3. In Section 1.4, I distill these concepts into a utility function for languages as efficient communication systems, but this utility function does not incorporate the effects of information processing constraints. In Section 1.5, I show how to incorporate information processing constraints into the language utility function, and give examples showing how to derive information locality effects from this utility function when we assume limits on short-term memory capacity.

1.2 Conceptual framework

My goal here is to provide an explanatory framework for natural languages in terms of efficient communication subject to information processing constraints. In this section, I first lay some groundwork about what precisely I mean by language and communication.

1.2.1 Language as a code vs. language as a distribution

The term "language" is used ambiguously in the literature and in common usage. "Language" can mean a code for expressing meanings, or it can mean a set (or probability

distribution) of sentences or utterances. In this section I will refer to these distinct ideas as "language as a code" and "language as a distribution". Here I am making only terminological distinctions; no nontrivial empirical claims should be implied. **Language as a code** is a *mapping* from a meaning to a distribution over utterances to express that meaning. I will notate a language as a code as a function $\mathcal{L}(\cdot)$. **Language as a distribution** is a *probability distribution* over utterances, as we might observe in a corpus. I will notate a language as a distribution as L.

Formally, let a **meaning** be an element of the set $\mathcal{M}$, called the **meaning space**. Let M be a probability distribution over $\mathcal{M}$. Going forward, I will try to make as few assumptions as possible about meanings and the space of meanings. I will often assume that M can be represented as a discrete distribution, but this should not be taken as crucial. I will also assume frequently that M is stationary, meaning that it does not change over time. If M is not stationary, then the extent to which languages adapt to its value at any particular point in time may be reduced. I discuss the effects of nonstationarity of M further in Section 5.3.

Let an **utterance** be an element of the set $\mathcal{U}$. An utterance is a sequence of symbols called linguistic elements (phonemes or morphemes, depending on the particular level of analysis). A **language as a code** $\mathcal{L}(\cdot)$ is a function from $\mathcal{M}$ to probability distributions over $\mathcal{U}$. A **language as a distribution** L is a probability distribution over $\mathcal{U}$, generated by drawing samples from M and expressing them into utterances with $\mathcal{L}(\cdot)$. A **corpus** is a finite set of samples from L. I also write L as $\mathcal{L}(M)$.

In formal language theory, a language is viewed as a set of strings, corresponding to language as a distribution (Hopcroft and Ullman, 1979). A probabilistic formal language (**p-language**: Ellis 1969; Kornai 2011) is exactly language as a distribution. Language as a code is just a p-language conditional on the meanings that one wishes to express when speaking.

Relation with Previous Concepts

The distinction of language as a code vs. language as a distribution overlaps with previous distinctions from the linguistic literature, but is separate from them in important ways. Chomsky (1988) made a distinction between **I-language**, meaning language as a formal

system known in the minds of speakers, and **E-language**, meaning observable linguistic behavior. I-language corresponds roughly to language as a code inasmuch as it instantiates a relationship between form and meaning. E-language corresponds to language as a distribution: or more precisely, E-language is a set of observations drawn from a language as a distribution. While I-language is mental and E-langauge is physically observable, I do not make any such distinction for language as a code vs. language as a distribution. My distinction is relevant for the description of languages as mathematical objects, and does not imply anything about the instantiation of these objects in minds and in the world.

The distinction orthogonal to the previous distinction of competence vs. performance. **Competence** is a speaker's knowledge of how language normatively works; Chomsky (1965) views it as a formal system. **Performance** is linguistic behavior, which is a function of competence but including practical constraints and sources of error. Along similar lines, Saussure (1916) made a distinction between ***langue***, meaning language as an abstract formal system as might be described in a grammar, and ***parole***, meaning observable speech. He gave the example of a language derived by taking French and mapping its phonemes onto other phonemes one-to-one. The utterances in the resulting language consist of phonetic sequences that are isomorphic to French phonetic sequences. Saussure (1916) claimed that the resulting language is the same *langue*, but results in different *parole*. In my framework, language as a code describes any mapping from meanings to utterances. The mapping may be factored into an idealized system (competence/*langue*) that is noisily translated into physical utterances (performance/*parole*), or it may not be, depending on the analysis.

The concept of performance introduces the notion of **error**: people might know a language and desire to speak according to that language, but still end up producing utterances that are not licensed by the language. For example, disfluencies, false starts, and sentence "blends" (Fromkin, 1971; Garrett, 1975, 1980; Fromkin, 1980), are all in the domain of performance and E-language, rather than competence and I-language. My concepts of language as a code and language as a distribution do not treat the notion of error in the same way. A speaker may know a language should be spoken a certain way, but fail to speak it that way due to planning errors; in my terminology, these errors happen in the mapping

from meaning to form, i.e. in language as a code. A language as a code is an object describing any mapping from meaning to utterance; if we wish to model the effects of speech errors, then a mapping $\mathcal{L}$ may be derived by adding speech errors to some other error-free mapping $\mathcal{L}'$.

The common distinction of **grammar** and **usage** is also separate from the one I wish to draw. I am intending language as a code to encompass all aspects of the mapping from a meaning to a distribution over utterances that one might say in a language to express that meaning. In this sense, "language as a code" subsumes phenomena such as pragmatic choice of utterances in particular contexts (Grice, 1975), which is often viewed as part of usage, not grammar. It may well be possible to separate pragmatics from the literal semantics of a language, with a notion of a literal, possibly nonprobabilistic language-as-a-code (grammar), which gives rise to probabilistic pragmatic behavior (usage). I will use a grammar vs. usage distinction in this sense in Section 3.3, for example. By language as a code, I simply mean the probability distribution over utterances a speaker would say conditional on an intended meaning.

1.2.2 Means of explaining properties of languages

The conceptual framework above for languages is extremely generic and does not yield any predictions at all about what the form of languages should be. I have only assumed that there are objects called meanings that are related to objects called utterances, which are sequences of symbols. I say that any mapping from meanings to sequences of symbols can be called a language. Thus I consider the range of possible, describable languages to be much larger than the set of observed languages. I will aim to describe the set of observed human languages—a small subset of possible languages—by hypothesizing that they are the languages which maximize a utility function related to communication under information processing constraints.

This conceptual approach differs from the common approach taken in linguistics, where the formalism in which languages are defined places constraints on what languages are possible—the notion of "explanation by constrained description" (Pollard and Sag, 1994;

Newmeyer, 1998, 2005; Haspelmath, 2010), as seen in Optimality Theory (McCarthy, 2002), and prominently associated with the field of generative formal syntax (e.g., Chomsky, 1965, 1988; Travis, 1989; McCloskey, 1993; Kayne, 1994). Even if the research program of explanation by constrained description succeeds, it raises the question of why the formal definitional restrictions that constrain languages exist. These restrictions may be justified externally, in terms of effort and utility, as is sometimes done in Optimality Theory (Bresnan, 1997; Aissen, 2003), or claimed to be arbitrary and innate: this is the notion of Universal Grammar (Chomsky, 1988).

The distinction between description and explanation of languages here is like the distinction between kinematics and dynamics in physics. Kinematics is the mathematical language for describing objects in motion without reference to the cause of motion; dynamics is the study of forces acting on those objects. It is possible to describe in kinematic language many spatial trajectories which are not licensed by any known laws of physics, just as it is possible to describe many bizarre languages as a function from meanings to utterances. Explanation of observed motion comes through dynamics, just as explanation of observed languages here comes through a utility function based on communication with information processing constraints.

The utility function for language as a code $\mathcal{L}$ with respect to meaning distribution M is to maximize the quantity of information transmitted about M in utterances, minus the cost of sending and receiving those utterances, all under information processing constraints. Utility of a language $\mathcal{L}$ is calculated according to the best possible behavior of the speaker and listener respecting their information processing constraints. These constraints induce bounds on how closely the speaker and listener can approximate the behavior described by $\mathcal{L}$. Languages that cannot be spoken or understood will have low utility, even if they would allow excellent communication in principle between ideal agents.

Note that the formulation here includes the possibility of learnability as a constraint, though I will not talk directly about learnability constraints in this work. Learnability affects the ability of the speaker and listener to approximate a language. It can also affect the cost of sending and receiving utterances, because if a language is hard to learn, then the speaker and hearer will have high uncertainty about the grammar, making processing more

costly. There is considerable evidence that learnability constraints, along with communicative pressure, are behind the emergence of systematicity in linguistic structure, because they force the language to be compressible (Kirby et al., 2008, 2014, 2015; Cornish et al., 2017). Although I will talk mostly about online processing constraints, this should not be taken to exclude the effects of learnability in this framework. Exploring how learnability interacts with the frameworks and phenomena discussed here is rich ground for future work.

Crucially, the utility function for languages depends on language as it is used. We will see this claim fleshed out in more detail below, but for now it is enough to note that appropriateness for communication is a function of what people actually say when they wish to express meanings. Thus communicative efficiency depends on language as it ends up being used; it is not solely based on language as a normative ideal abstract formal system known to speakers (competence, I-language, *langue*, etc.) We will also see below that, because language processing is affected by probabilistic expectations, the communicative efficiency of a language under information processing constraints depends to some extent on the language as a distribution independently from the language as a code (Section 1.5.4).

The fact that efficiency has to do with usage justifies a quantitative approach to syntax, analyzing the frequencies with which different constructions appear and the information theoretic properties of the resulting joint distribution over constructions. It will turn out that the probabilistic aspects of language as a distribution have major consequences for communication and efficiency; as a result, this work can be considered in the vein of usage-based linguistics (Langacker, 1987, 1991; Tomasello, 2003; Bybee, 2010). The importance of statistical distributions arises because linguistic processing involves forming probabilistic expectations about future material (Marslen-Wilson, 1975; Kutas and Hillyard, 1984; Hale, 2001; Kamide et al., 2003; Kliegl et al., 2004; Frisson et al., 2005; Dambacher et al., 2006; Levy, 2008a; Demberg and Keller, 2009). Chapter 2 is a case study in the practical issues that arise when attempting to use information theoretic concepts to study quantitative syntax in this way.

The key result that I argue for, and the source of the title of this thesis, is that under mild assumptions about memory resources used in the course of incremental processing, we can

derive locality constraints on optimal languages. In particular, words that depend on each other should be close to each other in linear order. Words can depend on each other because they are in a syntactic dependency relationsip, in which case we predict that dependency length should be minimized.

Below, I will first provide some background on the information theoretic concepts needed to analyze language as a code (Section 1.3). Then I will present the utility function for languages in an ideal setting (Section 1.4), and then show how to augment this utility function to account for information processing constraints (Section 1.5).

1.3 Information theoretic concepts

Here I will introduce some of the mathematical concepts that I will use in this thesis. These concepts are drawn primarily from the field of information theory (Shannon, 1948; Cover and Thomas, 2006). Although there were early attempts to use the tools of information theory to describe natural languages as codes (Shannon, 1948; Bell, 1953; Mandelbrot, 1953; Burton and Licklider, 1955; Pereira, 2000), the theory has mostly been used to describe digital codes, a concept I will contrast with natural languages in Section 1.3.5. For accessible and wide-ranging introductions to information theory, see MacKay (2003) and Cruise (2014). While introducing these concepts, I will emphasize connections with notions of efficient communication and coordination among agents.

For those who are already familiar with information-theoretic functions such as entropy and mutual information, Sections 1.3.1 and 1.3.2 are likely familiar. Sections 1.3.3 and 1.3.4 introduce some lesser-known information theoretic functions (interaction information and cross information) which will play a crucial role in the definition of the utility function for a language under information processing constraints.

1.3.1 Entropy

Entropy is the fundamental concept of information theory, providing a link between the probability of a message and the effort required to encode and send that message according to some code.

Suppose we are observing samples from probability distribution A and we want to write down a **codeword** for every sample $a \sim A$, where a codeword is a sequence of symbols drawn from some alphabet. For simplicity, we assume the alphabet has only two possible symbols in it, `0` and `1`, corresponding to a binary code. We notate the set of all finite-length sequences of `0`s and `1`s as $\{0,1\}^*$. The mapping from values $a \sim A$ to codewords in $\{0,1\}^*$ is a **code**. We want to design an **efficient code**, meaning the expected number of symbols we have to write is minimized.[2]

The expected length of codewords in an efficient code is given by the **entropy** of the distribution A (Shannon, 1948):

$$H(A) \equiv \mathop{\mathbb{E}}_{a \sim A} [-\log p_A(a)],$$

where the base of the logarithm is the alphabet size. When the alphabet size is 2 (which I will assume always, for simplicity), then entropy is measured in units called **bits**. The length of the codeword for any particular value $a \sim A$ according to the efficient code is given by the **surprisal** or **information content** of a:

$$h_A(a) \equiv -\log p_A(a).$$

I will also write $h(a)$ when the relevant probability distribution is clear.

The expression $h(a)$ is called surprisal because it is high for low probability values and low for high probability values. It measures how surprising the value a is, and entropy $H(A)$ represents uncertainty about what value will be sampled from A.

We can also see $h(a)$ as the number of random decisions required to generate a from A by an efficient program which minimizes the expected number of random decisions per sample generated. Inasmuch as deciding on random values requires energy, $H(a)$ measures a lower bound on the expected energy usage of such a program (Brillouin, 1953, 1956). Each random decision in an execution trace of the efficient program corresponds to a bit in a codeword for an efficient code. Thus we can see bits as both part of the representation of

[2] Also, efficiency in this narrow sense requires that no codeword for a value be a prefix of a codeword for another value.

a value, and as a representation of the decisions required to generate a value.

1.3.2 Conditional entropy and mutual information

Now I will introduce notions dealing with collections of random variables: conditional entropy and mutual information.

Suppose we have two jointly distributed random variables (A, B). The distribution (A, B) generates ordered pairs of values (a, b). Think of A as a probabilistic program that uses on average $H(A)$ random decisions to generate values. When seen alone, B also makes on average $H(B)$ random decisions when generating its values. But when we view A and B together, we might find that B's decisions are predictable from A's. In that case, then knowing the decisions made by A reduces our uncertainty about the decisions of B. We can measure **conditional entropy** $H(B|A)$, the expected remaining random decisions that B appears to make after we account for the ones that were predictable from A.

Alternatively, imagine we are assigning codewords to ordered pairs $(a, b) \sim (A, B)$. $H(A)$ is the expected number of bits we have to write for a and $H(B|A)$ is the expected number of bits we have to write for b. The intuition is that if b is predictable from a, then we can get away with writing fewer bits for b. Conditional entropy is:

$$H(B|A) \equiv \mathop{\mathbb{E}}_{a,b\sim A,B} \left[-\log p_{B|A}(b|a) \right].$$

We can also define **conditional surprisal**, the number of bits that have to be written to represent a particular b after a particular value a:

$$h_{B|A}(b|a) \equiv -\log p_{B|A}(b|a).$$

Conditional surprisal has the perhaps surprising property that it is possible for $h(b|a)$ to be greater than $h(b)$. That is, it is possible that one has to write down more bits for b as part of a code for (A, B) than one would have had to write in a code for only B. This situation arises when the value a makes b *more surprising* than it would have been otherwise. Nevertheless, *on average*, conditional entropy must be less than unconditional entropy (Cover

and Thomas, 2006):

$$H(B|A) \leq H(B).$$

Now I arrive at the notion of mutual information. Consider again the joint random variables (A, B). When we considered B alone, it appeared to be making $H(B)$ random decisions, but when we considered A and B together, we saw that B was actually only making $H(B|A)$ random decisions on its own, and the rest were copied from A. So the expected number of copied random decisions is $H(B) - H(B|A)$. Equivalently, this is the expected number of bits that are copied from codewords for A to codewords for B. **Mutual information** measures the number of bits that B is copying from A, as we observe when we see the two of them together. Mutual information is calculated as:

$$I(A;B) \equiv \mathop{\mathbb{E}}_{a,b\sim A,B}\left[\log \frac{p_{A,B}(a,b)}{p_A(a)p_B(b)}\right] = H(B) - H(B|A). \tag{1.1}$$

Since $H(B|A) \leq H(B)$, it follows that mutual information must be nonnegative:

$$I(A;B) \geq 0.$$

From the definition in Equation 1.1, we can also see that mutual information is symmetrical:

$$\begin{aligned} I(A;B) &= H(B) - H(B|A) \\ &= H(A) - H(A|B) \\ H(B|A) &= H(B) - I(A;B) \\ H(A|B) &= H(A) - I(A;B). \end{aligned}$$

That is, mutual information tells how many bits are being copied from one distribution to the other, but it does not tell us who is doing the copying. It simply tells us that when we view (A, B) *as a system*, it looks like there are fewer random decisions being made than when we viewed A and B separately. This view corresponds to the following equations for

mutual information:

$$I(A;B) = H(A) + H(B) - H(A,B) \tag{1.2}$$

$$H(A,B) = H(A) + H(B) - I(A;B). \tag{1.3}$$

These equations express that mutual information is the discrepancy between the number of bits that appear to be present in the system when we view A and B separately ($H(A) + H(B)$), and the number of bits in the system when we view A and B together ($H(A,B)$).

We can also define **pointwise mutual information** for a particular pair of values (a,b) as the difference between the length of the codeword for b with respect to B, and the number of bits representing b in the codeword for $(a,b) \sim (A,B)$:

$$\begin{aligned}
\text{pmi}_{A,B}(a;b) &= \log \frac{p_{A,B}(a,b)}{p_A(a)p_B(b)} \\
&= h_B(b) - h_{B|A}(b|a) \\
&= h_A(a) - h_{A|B}(a|b) \\
h_{B|A}(b|a) &= h_B(b) - \text{pmi}(a;b) \\
h_{A|B}(a|b) &= h_A(a) - \text{pmi}(a;b).
\end{aligned}$$

When pointwise mutual information is positive, we can see it as the number of shared bits in the representations $h(a)$ and $h(b)$, which get merged together in a joint representation $h(a,b)$. When it is negative, this means that writing b required more bits in the context of a than it would have required otherwise. While pointwise mutual information can be unboundedly negative, it cannot exceed the surprisal of either a or b:

$$\begin{aligned}
-\infty < \text{pmi}(a;b) &\leq h(a) \\
&\leq h(b).
\end{aligned}$$

These equations say that a cannot provide more information about b than is contained in a itself, nor can a provide more information about b than is contained in b to begin with.

To visualize mutual information, imagine A and B are two dots moving around on a

grid. Let A be a distribution over directions $\{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}\}$; the dot for A moves around according to these sampled directions. Let B be another randomly moving dot, placed next to A. When we look at either dot individually, it appears to be moving around unpredictably. But when we look at these two dots moving together, we might find that they are *synchronized*: B moves as A does. This is a case of high mutual information.

This synchronization is only possible if A and B share information about each other's random decisions. Let's assume that A is making truly random decisions, and B is simply following what A does: that is, the dots are moving around with perfect synchronicity. B has to receive information from A to know how it should move; if A does not make its decisions perceptible to B, then B will not know how to move. Mutual information measures the number of A's random decisions that it must make available to B to enable coordination. Equivalently, mutual information is the expected length of the shortest message that A must send to B to enable coordination.

For this reason, we can see mutual information as a measure of the energy usage required to maintain coordination. The energy required to send a message of some length is at least proportional to the length of the message, simply because sending each symbol takes effort. Thus, mutual information measures a lower bound on the *energy required for coordination* among two agents—they must send messages to each other of length determined by the mutual information. Thus coordination is harder than independence; joint actions that require more coordination require more energy, though they may lead to much higher reward.

The mutual information of a distribution (A, B) measures the coordination energy required to maintain the distribution beyond what would be required if A and B were independent. When A and B are not synchronized and move totally independently, then the total random decisions made by A and B is $H(A) + H(B)$ and mutual information is zero (following Equation 1.2).

Thus, mutual information is a highly general way of quantifying dependence among random variables, and it also gives some insight into the nature of communication. A and B must communicate to coordinate, and mutual information quantifies the amount of communication they will have to do.

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Table 1.1: An example joint distribution A, B, C. A and B are Bernoulli coinflips generating 0 or 1 with equal probability. $C = \text{XOR}(A, B)$.

1.3.3 Interaction information

In the course of this thesis I will use some lesser known information theoretic concepts, including interaction information. For attempts to visualize and give intuition about interaction information, see Bell (2003); Jakulin and Bratko (2003); Crooks (2016).

Interaction information (McGill, 1955) is a generalization of mutual information to the case of more than two variables. Recall that mutual information is the bits shared among 2 variables, which appeared to be random when we viewed the variables separately. Interaction information is the bits shared among n variables, which appeared to be random when we viewed all the strict subsets of the variables. For three variables, it is the difference in joint entropy of three variables (A, B, C) from what one would expect from observing any pair of them:

$$\begin{aligned} I(A;B;C) = H(A,B) + H(A,C) + H(B,C) \\ - [H(A) + H(B) + H(C)] \\ - H(A,B,C) \\ H(A,B,C) = H(A) + H(B) + H(C) \\ - I(A;B) - I(B;C) - I(A;C) \\ - I(A;B;C). \end{aligned} \tag{1.4}$$

Example A classic example of interaction information is the case of three variables A, B, C where A and B are independent Bernoulli variables generating 0 or 1, and $C = \texttt{xor}(A, B)$. The joint distribution of A, B, and C is described in Table 1.1.

The total joint entropy is $H(A, B, C) = H(A) + H(B|A) + H(C|A, B)$, by the chain

rule. The variable C is a deterministic function of A and B so $H(C|A,B) = 0$; and A is independent of B, so $H(B|A) = H(B)$. Thus $H(A,B,C) = H(A) + H(B) = 2$.

Now let's think of how to decompose this joint entropy into interaction informations, according to Equation 1.4. In that equation for this example, the pairwise mutual information terms such as $I(A;B)$ are all equal to 0, because no variable in A, B, C is predictable given *one* other variable. So if we calculate up to the last term, we get $1+1+1-0-0-0 = 3$. Now we know the result we want to get is 2, so we need $3 - I(A;B;C) = 2$. Thus $I(A;B;C) = 1$, indicating that there is 1 bit of information shared among A, B, C that could not be detected when considering the variables in isolation or in pairs.

Another way of looking at this example is to say that neither A nor B alone were informative about C, but when we considered A and B together, they provided 1 bit of information about C. Later I use this logic to describe a case where two words together might be informative about meaning in a way that cannot be detected from either word alone (Section 1.5.5).

We can also think about interaction information in terms of **conditional mutual information**, the expected mutual information between two variables A and B conditional on a third variable C. Conditional mutual information is:

$$\begin{aligned} I(A;B|C) &\equiv \mathop{\mathbb{E}}_{c\sim C} \mathop{\mathbb{E}}_{a,b\sim A,B|C} \left[\log \frac{p_{A,B|C}(a,b|c)}{p_{A|C}(a|c)p_{B|C}(b|c)}\right] \\ &= I(A;B) + I(A;B;C) \\ I(A;B;C) &= I(A;B|C) - I(A;B). \end{aligned}$$

For example, in the XOR case, the mutual information $I(A;B)$ was 0, but the conditional mutual information $I(A;B|C)$ was 1, because knowing C makes B become predictable from A. Thus the interaction information $I(A;B;C) = I(A;B|C) - I(A;B) = 1 - 0 = 1$.

In general, for a set α of n random variables, interaction information is (Tin, 1962; Jakulin and Bratko, 2003):

$$I(\alpha_1;...;\alpha_n) \equiv -\sum_{\beta\subseteq\alpha}(-1)^{n-|\beta|}H(\beta). \tag{1.5}$$

Interaction information must be nonnegative for even n, and it can be positive or negative for odd n. Negative interaction information corresponds to a case where viewing n variables together makes them appear to have *less* shared bits than viewing all the sets of $n-1$ variables. Positive interaction information means that viewing n variables together makes them appear to have *more* shared bits than when viewing all the sets of $n-1$ variables, as in the XOR example above.

1.3.4 Cross entropy and KL divergence

Next I introduce notions related to cross entropy. I will introduce a novel notion of cross information, which will be crucial in the description of communication under information processing constraints (Section 1.5).

Cross entropy measures the expected number of bits that are needed to encode samples from a distribution P, using a code that was optimized for another distribution Q. It is defined as (Cover and Thomas, 2006):[3]

$$H(Q \to P) \equiv \mathop{\mathbb{E}}_{x \sim P} \left[-\log p_Q(x) \right]. \tag{1.6}$$

Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) represents the *extra* bits of information that are required to encode samples from P in a code optimized for distribution Q:[4]

$$\begin{aligned} D_{\mathrm{KL}}(Q \to P) &\equiv \mathop{\mathbb{E}}_{x \sim P} \left[\frac{\log p_P(x)}{\log p_Q(x)} \right] \\ &= H(Q \to P) - H(P). \end{aligned}$$

Cross Information

I introduce here a notion of **cross information**, the mutual information shared between two jointly distributed random variables L and M conditional on the encoder knowing M and

[3] Cross entropy is usually written with the notation $H(P, Q)$. I use the notation in Equation 1.6 because (1) it is not ambiguous with $H(X, Y)$ meaning the joint entropy of X and Y; and (2) it makes clear that the distribution Q is meant to be approximating P. My notation for KL divergence has a similar rationale.

[4] More commonly notated as $D_{\mathrm{KL}}(P||Q)$ or $\mathrm{KL}(P||Q)$.

an approximate conditional distribution $\mathcal{L}' = L'|M$:

$$I(L' \to L; M) \equiv \mathop{\mathbb{E}}_{m\sim M, u\sim\mathcal{L}(m)} \left[\log \frac{p_{\mathcal{L}'}(u|m)}{p_{L'}(u)}\right] \tag{1.7}$$

$$= \mathop{\mathbb{E}}_{m\sim M, u\sim\mathcal{L}(m)} \left[\log \frac{p_{\mathcal{L}'}(u|m)p_L(u)p_{\mathcal{L}}(u|m)}{p_{L'}(u)p_L(u)p_{\mathcal{L}}(u|m)}\right]$$

$$= \mathop{\mathbb{E}}_{m\sim M, u\sim\mathcal{L}(m)} \left[\log \frac{p_{\mathcal{L}}(u|m)p_L(u)p_{\mathcal{L}}(u|m)}{p_L(u)p_{L'}(u)p_{\mathcal{L}'}(u|m)}\right]$$

$$= I(L; M) + D_{\mathrm{KL}}(L' \to L) - D_{\mathrm{KL}}(L' \to L|M), \tag{1.8}$$

with the **conditional KL divergence** $D_{\mathrm{KL}}(L' \to L|M)$ defined as:

$$D_{\mathrm{KL}}(L' \to L|M) \equiv \mathop{\mathbb{E}}_{m\sim M} \left[D_{\mathrm{KL}}(L'|m \to L|m)\right].$$

Conditional KL divergence must exceed KL divergence (**conditioning increases divergence**; $D_{\mathrm{KL}}(L' \to L|M) \geq D_{\mathrm{KL}}(L' \to L)$) (Gray, 1990, Chapter 5) (Polyanskiy and Wu, 2016), therefore:

$$I(L' \to L; M) \leq I(L; M).$$

This inequality expresses the fact that when meaning is transmitted using the distribution $\mathcal{L}$ and received by an agent that can only interpret it using $\mathcal{L}'$, less information is communicated than when the message is received by an agent that can use the true $\mathcal{L}$. Thus, a person who knows a language well will get more information about meaning from it than a person who knows the language less well.

The notion of cross information will play a key role in the definition of language utility under information processing constraints in Section 1.5.4. Information processing constraints mean that an agent is effectively decoding an utterance using a distribution $\mathcal{L}'$ which is different from the distribution $\mathcal{L}$ under which it was encoded. Thus cross information provides a way to quantify information loss due to information processing constraints.

1.3.5 Natural language vs. digital codes

Taking an information theoretic approach to linguistic communication suggests natural languages might share properties with the optimal codes studied in coding theory (MacWilliams and Sloane, 1981; Blahut, 1983; van Lint, 1999). A quick perusal of that literature will make it clear that there are major differences. The codes designed in these fields are not subject to human constraints such as limited memory and incremental planning capacity. I will call them digital codes, because they are intended for use by computers and other digital devices, as opposed to human codes (natural languages). Below, I give some examples that show that natural language has major differences from digital codes.

Digital codes are often **prefix-free**. This means that in an efficient code for independent identically distributed (iid) samples, the code will assign each sample a codeword in $\{0, 1\}^*$ with the constraint that for all codewords w, there is no other codeword w' such that w starts with w'. This constraint allows codewords for samples to be concatenated together as a stream without delimiters, and enables the stream as a whole to achieve its minimum possible expected length. If we take the words of a language to be similar to codewords for iid samples (a risky move), then we might wonder if words are prefix-free. But no known language has a prefix-free lexicon, although there is some evidence that word beginnings are more distinctive than word endings (King and Wedel, 2017).

The whole idea that natural language words might correspond to codewords reveals another way in which natural language differs from digital codes. Words as they appear in sentences are not probabilistically independent; if a sentence contains the verb *eat* then it is more likely than chance to contain the noun *food*. The occurrence of these words is correlated, so it is suboptimal to write them with context-invariant forms: because *food* is predictable from *eat*, it should be possible to write *food* in some shorter form. Again, natural languages show a tendency in this direction: Mahowald et al. (2013) show that speakers prefer reduced wordforms in contexts that make a word predictable, and Piantadosi et al. (2011) show that lengths of words are correlated with their average predictability in context. But in the vast majority of cases, words have the same form no matter what context they appear in, despite their probabilistic dependence on context, and this is plainly suboptimal

(Mandelbrot, 1953, p. 494). Pate (2017) shows formally that this constraint means that natural languages cannot reach their best possible efficiency. There must be some pressure in natural language, absent in digital codes, which makes wordforms consistent in context.

Words are also problematic because they are contiguous. When I use the word *dog*, the three phonemes that make it up all appear adjacent to each other. Words are concatenated together, rather than interleaved or combining according to some process that would make their forms context dependent. Jackendoff (2002) describes the concatenation of words as the "absolute universal bare minimum" of human languages (cf. Haspelmath, 2011). In general, more and more abstract levels of linguistic structure (morphemes, phrases, discourses) show less and less contiguity with increasing abstractness. Morphemes are mostly contiguous, except for occasional disruption by infixes and circumfixes. Phrases are contiguous inasmuch as a language is context-free (see Section 3.1.2), but discontiguous constituents exist (Weir, 1988; Joshi et al., 1991) (exemplified, for example, by *wh*-movement), and they are very common in some languages (Hale, 1983; Austin and Bresnan, 1996). Discourses are highly discontiguous, as shown in Wolf and Gibson (2005).

But in optimal codes used for communication in the face of noise, contiguity is not a desirable property. Suppose that a message is affected by noise before being received, such that some characters are erased or deleted. In order to make the message robust to this noise, digital codes use a method known as **block coding**. In block coding, a set of samples from a probability distribution is encoded into a string in $\{0, 1\}^*$, and then that string is segmented into blocks of k bits. Each block is then encoded into a string of length $n > k$ with some redundancy, such that a receiver receiving a noisy version of the code can correct errors. This procedure approaches the theoretical limits of efficiency for robust codes. Crucially, the block boundaries are independent of codewords, such that individual codewords are not necessarily represented by contiguous bits in the final resulting string. That is, the final resulting string has nothing at all comparable to the contiguous words of natural language; the bits of information about these words are distributed through the block.

Contiguous words are problematic even under a noise model which is highly plausible for human linguistic communication. Suppose noise typically affects contiguous parts of

an utterance; for example, a speaker is talking when a car goes by so that the listener misses a contiguous subsequence of the utterance. In the presence of this kind of noise, it would be optimal to spread the bits of information from each codeword out evenly throughout an utterance. Some natural language grammatical devices go in this direction: for example, we can see grammatical gender and agreement systems as error-correcting bits that spread information away from nouns (Futrell, 2010; Dye et al., ress). But nothing goes as far as would be optimal unless there were some other pressure strongly favoring contiguity and context-independence of words.

The examples in this section show that if we view natural language as an optimal efficient code, then it must operate under very different constraints from the optimal digial codes studied in coding theory. In particular, it must have pressures favoring context-invariance and contiguity of linguistic forms that have to do with particular dimensions of meaning (i.e., words). Previous work on the emergence of words as a linguistic unit (e.g., Nowak et al., 1999; Plotkin and Nowak, 2001; Tria et al., 2012; Spike et al., 2016) does not directly address the issue of contiguity. The pressures in favor of contiguity and context-invariance might have to do with both online processing and learnability.

To summarize, while languages have some features which point in the direction of the kinds of codes and constraints studied in coding theory, they have fundamental properties that digital codes do not. The goal of this work is in part to explain these discrepancies in an information theoretic framework, by considering constraints on memory and planning that apply when humans communicate using language but not when computers communicate using digital codes.

1.4 Utility of a language for ideal agents

Here I will define the utility of a language in the case of ideal agents, who know and agree on the language perfectly and who are able to use the language to encode and decode meaning with optimal efficiency. The utility function ultimately developed in this section is essentially the same as the one proposed on independent grounds for natural communication systems by Ferrer i Cancho and Solé (2003, Eq. 9), with further development in

Ferrer i Cancho (2005) and Ferrer i Cancho and Díaz-Guilera (2007). This function is a distillation of information theoretic notions of ideal codes; it does not include the influence of information processing factors. In the following section, I will show how to augment this utility function to account for these factors.

1.4.1 Reward and cost

The utility of a language $\mathcal{L}$ with respect to meaning distribution M is:

$$U_M(\mathcal{L}) \equiv kR_M(\mathcal{L}) - C_M(\mathcal{L}), \tag{1.9}$$

where R is a reward function, C is a cost function, and k is a constant stating the relative importance of reward vs. cost and converting reward into the same units as cost.

I will argue below for the following concrete expression for language utility:

$$U_M(\mathcal{L}) = kI(L;M) - H(L). \tag{1.10}$$

Reward

I define reward as the mutual information of $L = \mathcal{L}(M)$ and M:

$$R_M(\mathcal{L}) = I(L;M). \tag{1.11}$$

This term describes a pressure to maximize the information contained in L about M. It has its maximum at $I(L;M) = H(M)$, when the language as a code conveys all possible information about M. Such a code enables maximal coordination between agents. The minimum of $R_M(\mathcal{L})$ is 0.

The notion of reward here is similar to the concept of **channel capacity** (Shannon, 1948), except that channel capacity is computed according to the best possible source distribution, whereas reward here is calculated relative to a fixed source distribution M. Reward also corresponds to the notion of **reconstruction error** which is given as the communicative reward function in Regier et al. (2015). Simultaneous minimization of reconstruction

error and formal simplicity has been argued to explain category systems across languages and semantic domains. In that work, a speaker is assumed to choose an utterance to minimize a listener's reconstruction error for the speaker's intended meaning. Reconstruction error for a meaning $m \sim M$ and an utterance u is:

$$\begin{aligned} D_{\text{KL}}(M|u \to \delta_m) &= \log \frac{1}{p(m|u)} \\ &= h(m|u). \end{aligned}$$

Thus a speaker minimizes the listener's surprisal of the intended meaning given the utterance, as in RSA models (Goodman and Stuhlmüller, 2013). The expectation of this surprisal for a whole linguistic system is $H(M|L) = H(M) - I(L;M)$. Thus, when we minimize this expression with respect to the conditional distribution of utterances given meanings $\mathcal{L} = L|M$, it is the same as maximizing $I(L;M)$. So our maximizing our proposed reward function is the same as minimizing reconstruction error, a previously proposed cost function.

Cost

The cost function for a language requires somewhat more subtlety. Producing and comprehending utterances requires some effort. This effort involves many factors: the search time required to plan an effective utterance; the energy required to move one's articulators to produce speech; the attention required to focus on perceptual input in order to determine meaning; etc. Very generally, I propose that the cost of a language is the expected cost of its utterances:

$$C_M(\mathcal{L}) = \mathop{\mathbb{E}}_{m \sim M} \mathop{\mathbb{E}}_{u \sim \mathcal{L}(m)} \left[C_M(u) \right].$$

We can then consider lower bounds on the cost required to produce an utterance. The quantity $h_L(u)$ represents the number of bits required in an efficient encoding of an utterance u with respect to the language as a distribution L. Imagine that you must retrieve an utterance (or a word, or some other unit) from a store of utterances, and you can do so by making a series of binary cuts of the set of all utterances until you narrow down on the

correct one. $h_L(u)$ is the expected number of binary cuts you will have to make. In general, the surprisal of a value represents the number of *decisions* that have to be made to retrieve or produce that value.

The number of decisions provides a lower bound on effort required to produce or retrieve a value. In the most general sense, imagine a probabilistic program for generating samples from L conditional on M. The generation of any particular u will require at least $h_L(u)$ steps, corresponding to decisions that have to be made about the utterance, or to bits of meaning that have to be read and encoded into the utterance. So as a lower bound, we can expect that the effort required to produce u is a linear function of $h_L(u)$ (Brillouin, 1953, 1956). In accordance with this idea, there is evidence for decision making cost in humans being proportional to surprisal: in human psychometric data, the time taken to perform a task appears to be (approximately) linear in the number of decisions required, a generalization known as Hick's Law (Merkel, 1885; Hick, 1952; Hyman, 1953; Welford, 1960; Smith, 1968; Teichner and Krebs, 1974; Luce, 1986) (cf. Luce, 2003; Schneider and Anderson, 2011; Pavão et al., 2016).

As a measure of cost, the surprisal of an utterance $h_L(u)$ is also well justified on empirical grounds from the side of comprehension, according to well-motivated and well-supported theories of comprehension difficulty. According to Surprisal Theory (Hale, 2001; Levy, 2008a), the effort required to comprehend a word in context is proportional to the surprisal of the word in context. This theory has been validated as a predictor of reading times (Smith and Levy, 2013).

The expected value of $h_L(u)$ is $H(L)$, so I propose the following expression as a lower bound for cost:

$$C_M(\mathcal{L}) = H(L). \tag{1.12}$$

It is also possible to interpret $H(L)$ as representing the complexity of a language, which is related to its learnability. Using $H(L)$ as a cost function means that our utility function is **entropy-regularized** (Grandvalet and Bengio, 2005), meaning that complex languages are penalized. This interpretation of entropy as language complexity is appealing because it means that the overall utility function for languages contains a term for maximizing

informativity and minimizing complexity, the two criteria that have been observed to result in the emergence of natural-language-like codes in laboratory settings (Kirby et al., 2015).

In contrast with the definition here of utterance cost as surprisal, a common notion of message cost in information theory is message length. In an efficient code, expected message length corresponds to the entropy over messages (Shannon, 1948), as in Equation 1.12. It is very likely that the expected length of the actual symbol sequences making up utterances is a great deal higher than the entropic lower bound, because of the context-invariance of wordforms and other linguistic units (as discussed in Pate (2017); see also Section 1.3.5). Therefore, if we see utterance length as the true cost, then Equation 1.12 is only (proportional to) a lower bound on it.

When defining the language cost function, we run the risk of letting language cost become a dumping ground for arbitrary constraints. If arbitrary constraints can be encoded into language cost, then it would be possible to reproduce any desired pattern over resulting languages, rendering the theory unfalsifiable. To mitigate this risk, the expression for language cost should be maximally generic and/or justifiable on empirical grounds. In the framework I am setting up here, I am opting for the highly generic route, of defining cost as only the entropy of the language as a distribution. In doing so, I am dealing only with an absolute lower bound on language cost.

In Section 1.5, my goal will to incorporate information processing constraints into the utility function. It may be tempting to do so by putting these constraints in the cost function, penalizing utterances that we believe may be hard to plan, produce, and comprehend. I will not do it that way, because information processing constraints do not only affect language cost. They also degrade information transmission (the reward term), and thus require a different approach.

1.4.2 Combined utility function

Combining Equations 1.11 and 1.12, we get the utility function:

$$U_M(\mathcal{L}) = kI(L;M) - H(L). \tag{1.10}$$

Since $H(L)$ is only a lower bound on cost, this function is actually only an upper bound on utility. However, in the present work I will use it as the full utility function. Future work should explore more detailed cost functions.

The scaling factor k determines the relative importance of reward as opposed to cost. When $k \leq 1$, the reward of speaking does not overcome the cost of speaking, thus I will assume $k > 1$.

The utility function developed so far is essentially the same as the one provided by Ferrer i Cancho and Solé (2003), and studied subsequently in Ferrer i Cancho (2005); Ferrer i Cancho and Díaz-Guilera (2007). Subsequent work has offered explanations for a number of linguistic phenomena in terms of the maximization of this function, including synonymy avoidance (Clark, 1987; Ferrer i Cancho, 2017), Zipf's Law of meaning frequencies (Zipf, 1945; Ferrer i Cancho, 2016), and most prominently, Zipf's Law of word frequencies (Zipf, 1949; Ferrer i Cancho, 2005). See Salge et al. (2015) for an alternative utility function with similar aims.

While previous work has made use of this utility function to explain frequency distributions and properties of words, this thesis ultimately aims to derive properties of natural language syntax. For reasons explained in more depth in Section 5.1, the current form of the utility function does not make interesting predictions about syntax. However, in Section 1.5, I will develop an extension of the utility function to incorporate information processing constraints, taking it beyond previous work and allowing it to derive information locality and other ordering constraints on syntax.

1.4.3 No free variation

As an example of what we can conclude given the utility function in Equation 1.10, here I show that an optimal language is deterministic, meaning that every meaning in M is expressed by only one utterance, which has conditional probability 1. I do not wish to claim that real natural languages are deterministic, only that communicative utility pushes languages away from free variation.

We can rearrange Equation 1.10 to expose the conditional entropy of utterances given

meanings, $H(L|M)$:

$$\begin{aligned} U_M(\mathcal{L}) &= kI(L;M) - H(L) \\ &= kI(L;M) - H(L|M) - I(L;M) \\ &= (k-1)I(L;M) - H(L|M). \end{aligned} \tag{1.13}$$

Now $H(L|M) = 0$ when utterance is a deterministic function of meaning. If the space of languages as codes is unrestricted, then for every code $\mathcal{L}$ achieving some $I(L;M)$, there is another code achieving the same $I(L;M)$ but with $H(L|M) = 0$, thus attaining equal or higher utility.

Therefore fully optimal languages will not have free variation. **Free variation** denotes variation in utterances that is uncorrelated with meaning of any kind; the term is most common in phonology (Clark et al., 2007). The notion has been criticized on the grounds that apparently meaningless variation often contains social signalling information (Meyerhoff, 2006).

The interpretation of the argument against free variation here depends on the interpretation of the language cost function. Free variation means that there are bits of information in the language that are useless: they do not communicate any meaning. Producing these bits requires inherent effort. The presence of these bits in the language also will make message lengths needlessly long on average, because valuable real estate in the space of short utterances will be taken up by multiple variants of utterances for frequent meanings. If we interpret the language cost $H(L)$ as the complexity of the language, then free variation represents additional complexity which results in no benefit.

While I have shown that fully optimal languages will not have free variation, this result may not hold if the space of possible languages (or practically usable languages, as discussed in Section 1.5) is limited.

For a derivation of the converse claim—that optimal languages according to this utility function have a deterministic mapping from *utterances* to *meanings* (no ambiguity)—see Ferrer i Cancho (2017). Nevertheless, all known natural languages have ambiguity at the level of the utterance; such ambiguity can be explained in this framework by considering

the presence of outside, extralinguistic information that reduces uncertainty about meanings given utterances (Piantadosi et al., 2012).

1.5 Utility of a language under information processing constraints

So far I have built up an information-theoretic language for describing natural languages as codes, and I have argued that ideal communication systems maximize a utility function where reward is the quantity of information transmitted and cost is the effort required to send and receive messages. This utility function assumed that agents can encode and decode utterances with perfect efficacy and efficiency. But in real human communication, there are information processing constraints that affect our ability to use language optimally. For example, short-term memory constraints mean that when we are understanding one part of an utterance, we may have forgotten the exact form of the preceding parts of the utterance. Now I address the question of how these information processing constraints should be included in the utility function for language.

In this section I will develop a theory of communicative utility under information processing constraints. These constraints reduce the extent to which a language can convey information about meaning in practice, and they increase the cost of producing and comprehending utterances. I will focus on constraints introduced by incrementality and limited memory, which appear to be major constraints for humans. I show that memory constraints induce locality constraints: languages convey less information and are harder to process when utterance elements that depend on each other—either in terms of their distribution or in terms of their interpretation—are far from each other.

Next I will formalize this theory by generalizing the utility function for languages. The basic idea for the formalization is that, when we consider the utility of a language as a code $\mathcal{L}$, we should think of the producer and comprehender as encoding and decoding meaning using not $\mathcal{L}$, but rather using distorted languages $\mathcal{L}'$ which reflect their information processing constraints. We favor languages $\mathcal{L}$ which enable efficient communication even

when distorted by information processing constraints.

1.5.1 Incrementality

The primary constraint on human language processing, at least on the comprehension side, is incrementality. In spoken language—the dominant modality for human language for the vast majority of its time of existence—one hears an utterance once, incrementally, and cannot go back to listen to parts of it again (except at high cost, by asking the speaker to repeat herself).[5]

Utterances are sequences of symbols in time, and humans perceive them transiently, with very limited memory. These sensory facts put major constraints on what information processing can be done with the signal (Christiansen and Chater, 2016). In particular, comprehension must be done incrementally, with as much processing being done on the currently present signal as possible before more signal is received.

Next I formalize the notion of incrementality a bit. Consider an utterance as a sequence of linguistic units $\mathbf{w} = \{w_i\}_{i=1}^{n}, w_i \in W$. As a comprehender perceives this utterance, he encodes its meaning into some representation $c \in C$. Incrementality means that the comprehender encodes the meaning of the utterance $\mathbf{w}$ by successively applying some function $f : W \times C \rightarrow C$ which takes a currently perceived linguistic element w and integrates it into the current context representation c to produce a new context representation c'. For example, using f, a sequence of three linguistic elements w_1, w_2, w_3 is encoded as:

$$\text{enc}(w_1, w_2, w_3) = c_{\text{final}} = f(f(f(c_{\text{initial}}, w_1), w_2), w_3).$$

[5] In reading, one can look back to previous parts of an utterance, and this ability might explain differences in language structure between spoken and written texts. In particular, in reading, memory constraints are relaxed, so information locality effects should have less influence.

In general, a sequence $\mathbf{w}$ is encoded as:

$$\text{enc}(\mathbf{w}) = c_{\text{final}} = f(...(f(f(c_{\text{initial}}, w_1), w_2), ...), w_n) \tag{1.14}$$

$$= c_{\text{initial}} \textcircled{f} w_1 \textcircled{f} w_2 \textcircled{f} ... \textcircled{f} w_n, \tag{1.15}$$

where $c \textcircled{f} w$ means $f(c, w)$. I have motivated the definition of the representation-updating function f solely from the definition of incrementality. It is worth noting that f is exactly the function learned by recurrent neural networks, currently the best-performing language comprehension model at many tasks including language modeling (Jozefowicz et al., 2016) and translation (Wu et al., 2016; Johnson et al., 2016). f may be probabilistic, in which case the encoding function returns a probability distribution over possible representations.

Now I will show how the incremental encoding function of Equation 1.14 fits in with the theoretical framework I have been developing. A rational speaker who wishes the listener to achieve a representation c_{target} should plan and produce an utterance $\mathbf{w}$ such that the resulting distribution C_{final} over context representations is maximally close to c_{target}, minimizing the following function representing reconstruction error:

$$D_{\text{KL}}(\text{enc}(\mathbf{w}) \rightarrow \delta_{c_{\text{target}}}) = h(c_{\text{target}} | C_{\text{final}}). \tag{1.16}$$

The speaker's behavior here is a language as a code $\mathcal{L}$. Minimizing expected reconstruction error over possible meanings M is the same as maximizing language reward $I(L; M)$, as in Equation 1.10, for the reasons discussed in Section 1.4.1. Therefore the notion of incremental coding fits nicely into the proposed language utility framework.

Within this notation for thinking about incrementality, it is highly intuitive that comprehension effort appears to be a function of the surprisal of a word in context. Surprisal is the upper bound on how many bits will be written onto a representation c given observation of a word w. To see this, in $f(c, w)$, let c be a representation of some distribution the listener cares about, such as the posterior on intended meaning M, and let f perform a Bayesian update of c given new evidence w. Now $h(w|c)$ is the total bits of information in

w given knowledge of c, and as such it is the upper bound on the evidence w can provide about any third variable M given c (i.e., $I(W; M|c) \leq H(W|c)$). So at most $h(w|c)$ bits of information will be encoded onto c as a result of reading w. To explaining the observed surprisal effect, we only need the additional postulate that that the time taken to encode k bits is linear in k.

On its own, the notion of incrementality does not make any predictions about what kinds of sequences are easier to produce or comprehend. However, incrementality does make such predictions under further assumptions about memory, and limitations on what can be encoded in the incremental representation c.

1.5.2 Memory constraints

In this section, I argue that memory constraints affecting the incremental representation of an utterance result in processing cost and inaccuracy, and thus affect the utility of a language for communication. After this, I will show that these memory constraints give rise to locality constraints, such that groups of linguistic elements that are dependent should be placed near to each other in linear order to avoid processing cost and inaccuracy.

In incremental comprehension, at time t the comprehender has a representation c_t, which contains information about some variable that the comprehender is interested in, such as the speaker's intended meaning M. In order to influence the final representation c_{final}, the bits in c_t must be maintained in all the intermediate representations from c_t to c_{final}. But if memory is limited or faulty, then the relevant bits in c_t might become degraded in various ways through the timecourse of the sentence, such that some of the bits do not make it all the way to c_{final}. Here I show how this kind of memory constraint gives rise to processing cost, such that optimal languages have word orders that minimize this cost.

The logic for how memory constraints affect communicative efficiency is as follows. Suppose c_t is noisy with respect to the true sequence of linguistic units $w_{1:t}$ that gave rise to it: that is, a reconstruction of $w_{1:t}$ from c_t is noisy. This noisiness could be an inherent property of the memory in which c_t is stored, or it may be the result of there being a limit on how many bits can be stored in memory at a time, such that some bits of c_t might have to be

thrown out. Then if the word at time $t+1$ depends statistically on the exact words in $w_{1:t}$, the surprisal of that word given the context representation $h(w_{t+1}|c_t)$ might be greater than its surprisal given the true context, $h(w_{t+1}|w_{1:t})$. The extra bits of surprisal in $h(w_{t+1}|c)$ would represent *excess cost* beyond what would have been required if c_t had provided an exact representation of context.

A noisy representation c_t might not only cause excess processing cost, it might result in inaccuracy in comprehension. Suppose that the interpretation of a word w_{t+1} with respect to meaning depends on a previous context word $w_i, i < t+1$, but that c_t is noisy, such that w_i cannot be reconstructed from it. In that case, w_{t+1} might be interpreted incorrectly. Thus memory constraints cause c_t to be potentially noisy as a representation of $w_{1:t}$, which in turn creates potential for inaccuracy and duplicated effort in comprehension.

1.5.3 Locality constraints from memory constraints

Now let's make a further assumption that information about context becomes increasingly noisy the longer it has been kept in memory.[6] In that case, the excess processing cost due to memory limitations *increases when elements that predict each other are far apart.* To see this, consider the case where the surprisal (processing cost) of a word given the encoded context $h(w_{t+1}|c_t)$ was greater than its surprisal given its true context $h(w_{t+1}|w_{1:t})$. As c_t becomes noisier and noisier as a representation of the true context $w_{1:t}$, the discrepancy between these surprisals must increase on average. The true context $w_{1:t}$ contains information that lowers the surprisal of w_{t+1}, but if the representation of context does not contain the relevant bits, then they are not present to lower the surprisal of w_{t+1} conditional on the context representation. So the surprisal of w_{t+1} will be higher on average as c_t gets noisier as a representation of the true context. Thus we expect *more* processing cost and inaccuracy as linguistic elements that predict each other get farther apart. By similar logic, there should be inaccuracy in comprehension when words that depend on each other for their interpretation are far apart. This is the general idea of information locality, discussed in depth in Chapter 4.

[6] An increasing noise rate with time is unavoidable on average, as it is a natural consequence of the Data Processing Inequality (Cover and Thomas, 2006).

There is ample evidence for locality effects on human language processing difficulty. The most commonly discussed kind of locality is **dependency locality**, which denotes an increase in processing complexity when words that are syntactically dependent are far apart (Gibson, 1998, 2000; Grodner and Gibson, 2005; Demberg and Keller, 2008; Husain et al., 2015; Shain et al., 2016).

Locality in Languages

As argued above, for reasons of accuracy and efficiency, memory limitations which alter c_t from its ideal form should reduce the utility of a language. Supposing memory limitations are fixed at some level, we can compare languages based on how much inefficiency and inaccuracy is introduced by the memory limitations. In general, memory constraints militate against long-distance context-dependence in interpretation and syntactic distribution.

We saw that processing cost increases when linguistic elements that predict each other are far apart. By similar logic, inaccuracy increases when linguistic elements that depend on each other for interpretation are far apart. Thus, the best languages under memory constraints *are those where related words are close*: those that predict each other and those that depend on each other for interpretation. This is the idea of information locality as a constraint on languages.

Dependency locality—the idea that syntactic words in dependencies should be close in word order—is one kind of information locality constraint. In previous work, it has been argued to explain many syntactic universals of language such as Greenbergian word order correlations (Greenberg, 1963; Hawkins, 1994) as well as exceptions to these (Dryer, 1992; Gildea and Temperley, 2010), and also ordering preferences of constituents with regard to length (Behaghel, 1932; Yamashita and Chang, 2001; Wasow, 2002; Hawkins, 2004), and projectivity or context-freeness (Ferrer i Cancho, 2006). In Chapter 3, I provide detailed and large-scale corpus evidence across many languages that both grammar and usage are affected by dependency locality, in that they place syntactically related words close in linear order, beyond what would be expected from well-motivated baselines.

1.5.4 Formalization

Here I provide a formalization of the logic by which memory constraints affect the utility function for languages. I augment the utility function from Equation 1.10 to account for information processing limitations on the parts of the speaker and listener. Then I show how to derive from the augmented utility function the central predictions from Section 1.5.2, that memory constraints lead to inaccuracy and processing inefficiency. The resulting augmented utility function thus favors languages that exhibit information locality, as discussed in Section 1.5.3, or other properties that result in processing efficiency in general.

I formalize the notion of information processing difficulty using the concepts of cross entropy and cross information (introduced in Section 1.3.4). Language utility ends up splitting into different forms for the speaker and listener, a split which is not necessary for the more basic utility function in Equation 1.10 (Ferrer i Cancho and Díaz-Guilera, 2007). I leave for future work to explore in detail the extent to which these functions differ in behavior, and whether the form of languages is better explained by listener or speaker utility.

The idea behind the formalization is the following. Suppose we want to evaluate the utility of some hypothetical language as a code $\mathcal{L}$ with respect to meanings M. We do so by considering a speaker and listener who do not actually encode and decode utterances using $\mathcal{L}$, but rather using distorted languages $\mathcal{L}_s$ (for the speaker) and $\mathcal{L}_l$ (for the listener). These languages describe the behavior of agents who know the language $\mathcal{L}$, but are constrained by information processing constraints in how they apply this knowledge. An agent's distorted language represents the agent's *best possible behavior* given that they know $\mathcal{L}$ but have to approximate it under information processing constraints.

A language $\mathcal{L}$ may seem to have high utility when we consider it in the ideal case, but it might not yield efficient communication when evaluted under the best possible behavior of the speaker and listener under information processing constraints. This is the means by which information processing constraints are incorported into the utility function.

Listener's Utility

Imagine an agent is speaking a language $\mathcal{L}$ to another agent, who processes language as if it came from a related language, $\mathcal{L}_l$. The listener may well know that $\mathcal{L}$ is the source language, but in the course of incremental encoding he can only work relative to $\mathcal{L}_l$. The utility of the pair of languages $(\mathcal{L}, \mathcal{L}_l)$ for the *listener* is:

$$\begin{aligned} U^l_M(\mathcal{L}, \mathcal{L}_l) \equiv & \quad kI(L_l \to L; M) - H(L_l \to L) \qquad (1.17) \\ = & \quad kI(L; M) - kD_{\text{KL}}(L_l \to L|M) + kD_{\text{KL}}(L_l \to L) \\ & - H(L) - D_{\text{KL}}(L_l \to L) \\ = & \quad \underbrace{U_M(\mathcal{L})}_{\text{utility without processing constraints}} \quad \underbrace{- kD_{\text{KL}}(L_l \to L|M) + (k-1)D_{\text{KL}}(L_l \to L)}_{\text{utility loss due to processing constraints}}, \qquad (1.18) \end{aligned}$$

where $I(L_l \to L; M)$ is cross information (bits of meaning encoded using $\mathcal{L}$ decoded successfully using $\mathcal{L}_l$), and $H(L_l \to L)$ is cross entropy (the cost of processing samples from L as if they came from L_l). (See Section 1.3.4 for the properties of cross entropy and cross information.)

We model information processing constraints by supposing that the listener processes samples from L as if they came from some other distribution, L_l. Let's think about how this would play out in the case of memory constraints. In Section 1.5.2, I showed how a lossy memory representation c_t causes a linguistic unit w_{t+1} to have on average higher surprisal $h(w_{t+1}|c_t)$ than $h(w_{t+1}|w_{1:t})$, resulting in excess processing cost. The expected surprisal of the distribution over words W_{t+1} conditional on a true generating context, $w_{1:t}$, is $H(W_{t+1}|w_{1:t})$. Now the expected surprisal of W_{t+1} conditional on a context representation c_t, when W_{t+1} *was actually* generated conditional on $w_{1:t}$, is a cross entropy $H(W_{t+1}|c_t \to W_{t+1}|w_{1:t}) = H(W_{t+1}|w_{1:t}) + D_{\text{KL}}(W_{t+1}|c_t \to W_{t+1}|w_{1:t})$.

We can see the conditional distribution $W_{t+1}|w_{1:t}$ as defining a language L. The key idea here is that the distribution $W_{t+1}|c_t$ *defines a new language* L_l, a distortion of L where symbols are generated sequentially conditional on a lossy representation of their context, rather than conditional on their true context. This is the meaning of L_l in Equation 1.17,

and the justification for the term $H(L_l \rightarrow L)$ as processing cost under information processing limitations. The logic for the cross information term is similar: information processing constraints implicitly define a new language L_l, under which samples from L are interpreted.

Treating information processing constraints as cross entropies guarantees that these constraints lead to higher cost and lower reward. In the case of cost, higher cost is guaranteed because $H(L_l \rightarrow L) \geq H(L)$. In the case of reward, an example is instructive: see Section 1.5.5.

Finally, I consider how $\mathcal{L}_l$ is chosen based on $\mathcal{L}$. The listener wants to maximize the utility function in Equation 1.17, and finds the best $\mathcal{L}_l$ for that purpose:

$$\mathcal{L}_l = \underset{\mathcal{L}_l}{\operatorname{argmin}}\, kD_{\mathrm{KL}}(L_l \rightarrow L|M) - (k-1)D_{\mathrm{KL}}(L_l \rightarrow L). \tag{1.19}$$

This equation expresses the idea that the information processing characteristics of the listener are shaped by a desire to maximize language utility. It has a minimum when $\mathcal{L}_l = \mathcal{L}$, that is, when the listener can perfectly approximate the language of the interlocutor. But under cognitive limitations, it may not be possible to achieve $\mathcal{L}_l = \mathcal{L}$, resulting in processing-based cost and inaccuracy.

Equation 1.19 expresses that information processing (but not the space of possible information processing algorithms) is shaped to some extent by the language being spoken; for example, if memory capacity is limited to b bits, then following Equation 1.19 would lead a listener to save only those bits in memory that have the highest contributions to utility, thus performing lossy compression. See Section 5.4.4 for more discussion of this idea.

Speaker's Utility

The expression for the speaker's utility under information processing constraints is similar to that of the listener. We assume the speaker knows a language $\mathcal{L}$ and believes that her interlocutors follow it, but speaks herself according to a distorted distribution $\mathcal{L}_s$ integrating

information processing constraints:

$$U^s_M(\mathcal{L}, \mathcal{L}_s) = kI(L \rightarrow L_s; M) - H(L_s). \tag{1.20}$$

This is identical to the listener's utility Equation 1.17, except that (1) the "known" language $\mathcal{L}$ is now the approximating distribution in the mutual information term, and (2) the cost of generation is not a cross entropy, but rather the plain entropy of the distorted distribution L_s. Difference (1) reflects the fact that the producer speaks in a certain way $\mathcal{L}_s$, but the listener will interpret it according to a different distribution $\mathcal{L}$. Difference (2) reflects the fact that the producer's cost has to do with what she actually produces; whereas the comprehender's cost has to do with what he receives, which is out of his control. These asymmetries have the effect that the speaker's utility does not reduce as nicely as the listener's utility in Equation 1.18.

For the speaker, the primary constraints that might give rise to $\mathcal{L}_s \neq \mathcal{L}$ are constraints on incremental planning of utterances (Lashley, 1951; MacDonald, 1999; MacDonald, 2013). These constraints have been advanced as an explanation for "easy-first" ordering preferences in language, whereby words that are easier to produce in a context are produced early in an utterance (Bock, 1982; Levelt, 1982; Bock and Warren, 1985; Chang, 2009; Tanaka et al., 2011). I will not discuss these phenomena in detail in this thesis, but I note that they can be accommodated in this framework in the speaker's utility. In general, we can see the speaker as having a noisy sampler for sequences $\mathcal{L}$, which produces the distribution $\mathcal{L}_s$. See Section 5.4 for more detailed discussion of this point.

Given the utility in Equation 1.20, the speaker's best $\mathcal{L}_s$ is simply

$$\mathcal{L}_s = \operatorname*{argmax}_{\mathcal{L}_s} U^s_M(\mathcal{L}, \mathcal{L}_s); \tag{1.21}$$

no terms can be removed because they all contain L_s.

Remarks

The expression for listener's utility reveals an interesting case where language as a distribution L appears to matter separately from language as a code for meaning $\mathcal{L}$. From the listener's perspective, ideal languages must produce distributions over utterances that can be well-approximated. To see this, we use the fact that $D_{\mathrm{KL}}(L_l \to L|M) > D_{\mathrm{KL}}(L_l \to L)$ (*conditioning increases divergence*) to write an upper bound on the listener's utility:

$$\begin{aligned} U_M^l(\mathcal{L}, \mathcal{L}_l) &= U_M(\mathcal{L}) - kD_{\mathrm{KL}}(L_l \to L|M) + (k-1)D_{\mathrm{KL}}(L_l \to L) && (1.18) \\ &\leq U_M(\mathcal{L}) - kD_{\mathrm{KL}}(L_l \to L) + (k+1)D_{\mathrm{KL}}(L_l \to L) \\ &= U_M(\mathcal{L}) - D_{\mathrm{KL}}(L_l \to L). && (1.22) \end{aligned}$$

It follows that the listener's utility must be less than $U_M(\mathcal{L}) - D_{\mathrm{KL}}(L_l \to L)$, hence the utility of the language is affected by the ability of the listener to approximate the language *as a distribution* without regard for meaning. This result justifies the study of efficiency in languages based solely on the language as a distribution, without regard to unobservable meaning.

From the perspective of reducing the ill effects of information processing constraints, the best language is one where it is possible to achieve $\mathcal{L}_s = \mathcal{L}_l$. In that case, the speaker and listener utilities simplify to the ideal utility, and there is no excess cost or inaccuracy due to information processing constraints. Yet all observed languages are subject to inaccuracy in comprehension and production. This happens because achieving $\mathcal{L}_s = \mathcal{L}_l$ may come at the cost of overall combined utility for the speaker and listener. That is, it may be that there is a language pair $(\mathcal{L}_s, \mathcal{L}_l)$ which achieves high combined utility for the speaker and listener with some excess processing cost, but this level of utility cannot be achieved with zero excess processing cost. Those languages that can be spoken error-free might be too simplistic to convey much information.

W_1	W_2	M
a	a	0
a	b	1
b	a	1
b	b	0

Table 1.2: An example language $\mathcal{L}$ generating ordered pairs (W_1, W_2) conditional on M. For meaning 0, the language generates aa or bb with equal probability. For 1, it generates ab or ba with equal probability.

1.5.5 Example: Context-dependence in interpretation

Here I will work out an example, demonstrating the idea that memory constraints disfavor context-dependence in interpretation, as claimed in Section 1.5.3. The basic idea is just that if the interpretation of a word depends on context, and context has been forgotten, then the word will likely be interpreted incorrectly.

Suppose that a language $\mathcal{L}$ for meanings M has utterances that always consist of two symbols in order, W_1 and W_2. For the sake of the example, let M be a Bernoulli variable generating 0 or 1 with equal probability. Furthermore let W_1 and W_2 range over the alphabet $\{a, b\}$. For meaning 0, the language generates aa or bb with equal probability. For 1, it generates ab or ba with equal probability. The specification of $\mathcal{L}$ is shown in Table 1.2. (This example may be familiar from Section 1.3.3.)

Now let us consider how $\mathcal{L}$ would be understood by a comprehender with extremely limited memory for wordforms. The comprehender reads W_1 and then W_2. Suppose that the comprehender's representation of context contains no information at all about words before the one being currently perceived. The comprehender may have a perfect representation of *meaning* as inferred from previous words, but does not remember the wordforms themselves. So after reading W_1, the comprehender's distribution over possible meanings will be 0 with probability $\frac{1}{2}$ and 1 with probability $\frac{1}{2}$. That is, reading the first word provided no useful information about meaning at all.

Now when the comprehender reads W_2, he is interpreting it in isolation. So while the distribution over words at this point in the true language $\mathcal{L}$ is $W_2|M, W_1$, we would model the comprehender's encoding distribution $\mathcal{L}_l$ at this point as $W_2|M$, where M is the comprehender's inferred distribution over meaning given the first word, representing

the fact that the comprehender does not remember wordforms. Furthermore, M is not informative about W_2 in isolation—for any given meaning, W_2 could be `a` or `b` with equal probability. So the listener's distribution over the second word after hearing the first word is just the unigram distribution over the second word, W_2.

In that case, the cross information of M and L under L_l is:

$$\begin{aligned}
I(L_l \to L; M) &= \mathop{\mathbb{E}}_{m\sim M} \mathop{\mathbb{E}}_{w_1,w_2\sim\mathcal{L}(m)} \left[\log \frac{p_{\mathcal{L}_l}(w_1,w_2|m)}{p_{\mathcal{L}_l}(w_1,w_2)}\right] \\
&= \mathop{\mathbb{E}}_{m\sim M} \mathop{\mathbb{E}}_{w_1,w_2\sim\mathcal{L}(m)} \left[\log \frac{p_{\mathcal{L}}(w_1|m)p_{\mathcal{L}}(w_2|m)}{p_{\mathcal{L}}(w_1)p_{\mathcal{L}}(w_2)}\right] \\
&= \mathop{\mathbb{E}}_{m\sim M} \mathop{\mathbb{E}}_{w_1,w_2\sim\mathcal{L}(m)} \left[\log \frac{p_{\mathcal{L}}(w_1|m)p_{\mathcal{L}}(w_2|m)p_{\mathcal{L}}(w_1,w_2|m)p_{\mathcal{L}}(w_1,w_2)}{p_{\mathcal{L}}(w_1)p_{\mathcal{L}}(w_2)p_{\mathcal{L}}(w_1,w_2|m)p_{\mathcal{L}}(w_1,w_2)}\right] \\
&= I(W_1,W_2;M) + I(W_1;W_2) - I(W_1;W_2|M) \\
&= \underbrace{I(L;M)}_{\text{information transmitted in ideal case}} - \underbrace{I(W_1;W_2;M)}_{\text{information loss due to memory constraints}} .
\end{aligned}$$

Thus, because of the comprehender's memory limitations, a quantity of information equal to $I(W_1;W_2;M)$ was not received. In this case of this example language, that $I(W_1;W_2;M) = 1$ bit was the entire relevant information about meaning, so the language would have at best 0 utility for the listener. (For the speaker, under these memory constraints it would simply be impossible to produce informative samples from $\mathcal{L}$.)

Thus, in the face of memory limitations, *linguistic elements should be context-independent in their interpretation.* In a way, this example already contains the ideas of information locality, domain minimization, etc., which this thesis will be focusing on.

1.5.6 Example: Context-dependence in form

As a second example, I consider a case where processing constraints lead to additional processing cost, but not to inaccuracy in information transmission. This case arises due to context-dependence in utterance form: the form of one word depends on the form of another word in a way that does not contribute to meaning. Consider a language on the alphabet $\{\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d}\}$ specified by the function $\mathcal{L}$ shown in Table 1.3.

Let us consider the listener's utility for this language under an incremental processing

W_1	W_2	M
a	a	0
b	b	0
c	c	1
d	d	1

Table 1.3: An example language $\mathcal{L}$ generating ordered pairs (W_1, W_2) conditional on M. All pairs (W_1, W_2) are generated with uniform probability conditional on M.

W_1	W_2	M
a	a	0
a	b	0
b	a	0
b	b	0
c	c	1
c	d	1
d	c	1
d	d	1

Table 1.4: $\mathcal{L}_l$ implied by Table 1.3 under a processing model where the comprehender has no memory for wordforms.

model where, after each word, the listener remembers an inferred meaning, but does not remember the exact wordform that was used. That is, after hearing a as W_1, the listener infers that the meaning is 0, but then does not know whether the next word will be a or b. Thus the listener's information processing constraints imply an approximating language $\mathcal{L}_l$ as shown in Table 1.4.

The excess cost of processing this language ($D_{\text{KL}}(L_l \to L)$, Equation 1.17) comes out to 1 bit:

$$\begin{aligned} D_{\text{KL}}(L_l \to L) &= \mathop{\mathbb{E}}_{m\sim M} \mathop{\mathbb{E}}_{w_1,w_2\sim\mathcal{L}(m)} \left[\log \frac{\mathbb{E}_{m'\sim M}\left[p_{\mathcal{L}}(w_1, w_2|m')\right]}{\mathbb{E}_{m'\sim M}\left[p_{\mathcal{L}_l}(w_1, w_2|m')\right]}\right] \\ &= \mathop{\mathbb{E}}_{m\sim M} \mathop{\mathbb{E}}_{w_1,w_2\sim\mathcal{L}(m)} \left[\log \frac{\frac{1}{2}\cdot\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2}}{\frac{1}{2}\cdot\frac{1}{4} + \frac{1}{2}\cdot\frac{1}{4}}\right] \\ &= \mathop{\mathbb{E}}_{m\sim M} \mathop{\mathbb{E}}_{w_1,w_2\sim\mathcal{L}(m)} \left[\log 2\right] \\ &= 1 \text{ bit.} \end{aligned}$$

However, the language does not induce any information loss under memory constraints.

The listener's information reward is $I(L_l \rightarrow L; M) = I(L; M) - D_{\text{KL}}(L_l \rightarrow L|M) + D_{\text{KL}}(L_l \rightarrow L)$. As argued above, the unconditional KL divergence term is equal 1 bit. To calculate *conditional* KL divergence, note that conditional on any value $m \sim M$, the distribution $\mathcal{L}_l(m)$ is the product of the marginals for $W_1, W_2|m$. Now mutual information is equal to the KL divergence from the product of the marginals to the joint distribution, so it follows that $D_{\text{KL}}(L_l \rightarrow L|M) = I(W_1; W_2|M) = 1$ bit. Thus the communicative reward $I(L_l \rightarrow L; M) = I(L; M) + 1 - 1 = I(L; M)$.

This redundant language is suboptimal because, under memory constraints, it creates extra processing work. The choice of `aa` or `bb` to express meaning `0` is essentially noise; and the correlation of W_1 and W_2 creates correlated noise. This example shows that correlated noise is suboptimal in languages: it creates extra processing effort, because it takes extra resources to predict specific wordforms which have no utility for communication.[7] Note that, for example, if the language in Table 1.4 were the true language, there would be no excess processing cost, because there would be no need to remember the form of W_1 to predict W_2.

1.6 Summary and roadmap

In this introduction I have provided a formal framework in which we can view natural languages as local maxima in a utility function defined by communicative efficiency under information processing constraints. The bulk of the thesis will consist of in-depth empirical and theoretical studies of the specific ideas brought up here, without explicitly situating those studies in the formal framework.

Chapter 2 is a case study on the concept of analyzing syntax quantitatively using the information theoretic concepts developed here. I study variation in word order conditional on dependency structure, and examine theoretical issues and also practical issues that arise in trying to do this with current data and statistical methods. The basic idea here is to take the unordered dependency tree structure as a partial representation of the meaning M, and think about word orders as codes $\mathcal{L}$ for M. I propose the entropy of word orders

[7] However, it has utility in the case of communication over a noisy channel, because it adds redundancy.

conditional on unordered dependency trees $H(L|M)$ as the central measure of word order freedom; in a communication theoretic framework, this quantity represents the maximal information that word order in a language can convey beyond what it is conveying about predicate-argument structure. I show in corpora that in languages where subject and object can be distinguished easily given morphology, the order of these words is more variable. This work was published as Futrell et al. (2015c, DepLing).

Chapter 3 is a detailed empirical study of dependency length minimization in crosslinguistic dependency corpora. I compare observed dependency length to expected dependency length under independently motivated constraints, such as projectivity and consistency in head direction. This work was published as Futrell et al. (2015b, PNAS). I also address the question of whether dependency length minimization affects grammar or usage or both, by comparing observed dependency length to dependency length under possible *grammatical* reorderings of dependency trees per language. This comparison is accomplished by developing a probabilistic model of word orders conditional on unordered dependency trees; a paper describing this model was published as Futrell and Gibson (2015, EMNLP).

Chapter 4 proposes a new theory of human sentence processing difficulty, **noisy-context surprisal**, that reconciles approaches based on memory (Gibson, 1998; Lewis and Vasishth, 2005) with those based on probabilistic expectations (Hale, 2001; Levy, 2008a), and provides a model of **structural forgetting**, a phenomenon involving interactions of these two factors (Gibson and Thomas, 1999; Vasishth et al., 2010; Frank et al., 2016). In addition I show that noisy-context surprisal derives information locality effects, and provide evidence for information locality from crosslinguistic corpora. I argue that dependency locality (thus dependency length minimization) and information locality are linked under the hypothesis that syntactic dependencies correspond to word pairs with high mutual information; I provide evidence that this is true, and I speculate on the theoretical justification for why this is true. This work was published as Futrell and Levy (2017, EACL).

Chapter 5 discusses extensions and future directions, including connections to natural language processing, and concludes.

Chapter 2

Case Study in Quantitative Syntax: Word Order Freedom

2.1 Introduction

Comparative cross-linguistic research on the quantitative properties of natural languages has typically focused on measures that can be extracted from unannotated or shallowly annotated text. For example, probably the most intensively studied quantitative properties of language are Zipf's findings about the power law distribution of word frequencies (Zipf, 1949). However, the properties of languages that can be quantified from raw text are relatively shallow, and are not straightforwardly related to higher-level properties of languages such as their morphology and syntax. As a result, there has been relatively little large-scale comparative work on quantitative properties of natural language *syntax*.

In recent years it has become possible to bridge that gap thanks to the availability of large dependency treebanks for many languages and the development of standardized annotation schemes (de Marneffe et al., 2014; Nivre, 2015; Nivre et al., 2015). These resources make it possible to perform direct comparisons of quantitative properties of dependency trees. Previous work using dependency corpora to study crosslinguistic syntactic phenomena includes Liu (2010), who quantifies the frequency of right- and left-branching in dependency corpora, and Kuhlmann (2013), who quantifies the frequency with which natural language dependency trees deviate from projectivity. Other work has studied graph-theoretic properties of dependency trees in the context of language classification (Liu and Li, 2010; Abramov and Mehler, 2011).

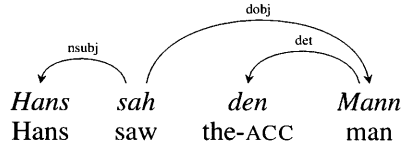
Here we study a particular quantitative property of language syntax: word order freedom. We focus on developing linguistically interpretable measures, as close as possible to an intuitive, relatively theory-neutral idea of what word order freedom means. In doing so, a number of methodological issues and questions arise. What quantitative measures map most cleanly onto the concept of word order freedom? Is it feasible to estimate the proposed measure given limited corpus size? Which corpus annotation style—e.g., content-head dependencies or dependencies where function words are heads—best facilitates crosslinguistic comparison? In this work, we argue for a set of methodological decisions which we believe balance the interests of linguistic interpretability, stability with respect to corpus size, and comparability across languages.

We also present results of our measures as applied to 34 languages and discuss their linguistic significance. In particular, we find that languages with quantitatively large freedom in their ordering of subject and object all have nominative/accusative case marking, but that languages with such case marking do not necessarily have much word order freedom. This asymmetric relationship has been suggested in the typological literature (Kiparsky, 1997), but this is the first work to verify it quantitatively. We also discuss some of the exceptions to this generalization in the light of recent work on information-theoretic properties of different word orders (Gibson et al., 2013).

2.2 Word order and the notion of dependency

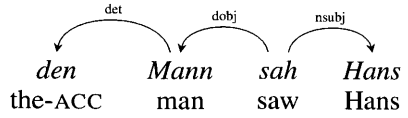
We define **word order freedom** as the extent to which the same word or constituent in the same form can appear in multiple positions while retaining the same predicate-argument structure and preserving grammaticality. For example, the sentence pair (1a-b) provides an example of word order freedom in German, while sentence pair (2a-b) provides an example of a lack of word order freedom in English. However, the sentences (2a) and (2c) do *not* provide an instance of word order freedom in English by our definition, since the agent and patient appear in different syntactic forms in (2c) compared to (2a). We provide dependency syntax analyses of these sentences below.

(1a)



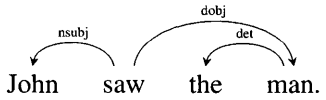
Meaning: "Hans saw the man."

(1b)

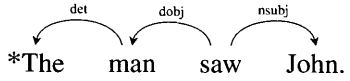


Meaning: "Hans saw the man."

(2a)

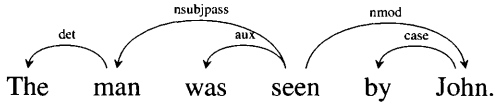


(2b)



Cannot mean: "John saw the man."

(2c)



In the typological literature, this phenomenon has also been called *word order flexibility*, *pragmatic word order*, and a lack of *word order rigidity* (Givón, 1992). These last two terms reflect the fact that word order freedom does not mean that that word order is random. When word order is "free", speakers might order words to convey non-propositional aspects of their intent. For example, a speaker might place certain words earlier in a sentence in order to convey that those words refer to old information (Ferreira and Yoshita, 2003); a speaker might order words according to how accessible they are psycholinguistically (Chang, 2009); etc. In English, word order is used to convey whether an expression is a question or a statement. Word order may be predictable given these goals, but here we are interested only in the extent to which word order is conditioned on the predicate-argument structure of an utterance.

In a dependency grammar framework, we can conceptualize word order freedom as variability in the linear order of words given an unordered dependency graph with labelled edges. For example, both sentences (1a) and (1b) are linearizations of the unordered dependency graph in Figure 2-1.

The dependency formalism also gives us a framework for a functional perspective on why word order freedom exists and under what conditions it might arise. In general, the task of understanding the propositional meaning of a sentence requires identifying which

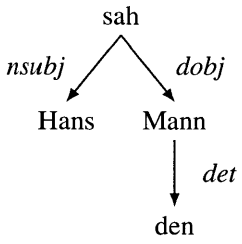


Figure 2-1: Unordered dependency graph representing a class of German sentences.

words are linked to other words, and what the relation types of those links are. The dependency formalism directly encodes a subset of these links, with the additional assumption that links are always between exactly two explicit words. Therefore, we can roughly view an utterance as an attempt by a language producer to serialize a dependency graph such that a comprehender can recover it. The producer will want to choose a serialization which is efficient to produce and which will allow the comprehender to recover the structure robustly. That is, the utterance must be informative about which pairs of words are linked in a dependency, and what the relation types of those links are.

Here we focus on the communication of relation types. In the English and German examples above, the relation types to be conveyed are *nsubj* and *dobj* in the notation of the Universal Dependencies project (Nivre et al., 2015). For the task of communicating the relation type between a head and dependent, natural languages seem to adopt two non-exclusive solutions: either the order of the head, the dependent, and the dependent's sisters is informative about relation type (a word order code), or the wordform of the head or dependent is informative about relation type (Nichols, 1986) (a case-marking code). Considerations of robustness and efficiency lead to a prediction of a tradeoff between these options. If a language uses case-marking to convey relation type, then word order can be repurposed to efficiently convey other, potentially non-propositional aspects of meaning. On the other hand, if a language uses inflexible word order to convey relation type, then it would be inefficient to also include case marking. However, some word order codes are less robust to noise than others (Gibson et al., 2013; Futrell et al., 2015a), so certain rigid word orders might still require case-marking to maintain robustness. Similarly, some case-marking systems might be more or less robust, and so require rigid word order.

The idea that word order freedom is related to the prevalence of morphological marking is an old one (Sapir, 1921). A persistent generalization in the typological literature is that while word order freedom implies the existence of morphological marking, morphological marking does not imply the existence of word order freedom (Kiparsky, 1997; McFadden, 2003). These generalizations have been made primarily on the basis of native speaker intuitions and analyses of small datasets. Such data is problematic for measures such as word order freedom, since languages may vary quantitatively in how much variability they have, and it is not clear where to discretize this variability in order to form the categories "free word order" and "fixed word order". In order to test the reality of these generalizations, and to explore explanatory hypotheses for crosslinguistic variation, it is necessary to quantify the degree of word order freedom in a language.

2.3 Entropy measures

Our basic idea is to measure the extent to which the linear order of words is determined by the unordered dependency graph of a sentence. A natural way to quantify this is **conditional entropy**:

$$H(X|C) = \sum_{c \in C} p_C(c) \sum_{x \in X} p_{X|C}(x|c) \log p_{X|C}(x|c), \tag{2.1}$$

which is the expected conditional uncertainty about a discrete random variable X, which we call the **dependent variable**, conditioned on another discrete random variable C, which we call the **conditioning variable**. In our case, the "perfect" measure of word order freedom would be the conditional entropy of sequences of words given unordered dependency graphs. Directly measuring this quantity is impractical for a number of reasons, so we will explore a number of entropy measures over partial information about dependency trees.

Using a conditional entropy measure with dependency corpora requires us to decide on three parameters: (1) the method of estimating entropy from observed joint counts of X and C, (2) the information contained in the dependent variable X, and (3) the information contained in the conditioning variable C. The two major factors in deciding these parameters are avoiding data sparsity and retaining linguistic interpretability. In this section we discuss the detailed considerations that must go into these decisions.

2.3.1 Estimating entropy

The simplest way to estimate entropy given joint counts is through maximum likelihood estimation. However, maximum likelihood estimates of entropy are known to be biased and highly sensitive to sample size (Miller, 1955). The bias issues arise because the entropy of a distribution is highly sensitive to the shape of its tail, and it is difficult to estimate the tail of a distribution given a small sample size. As a result, entropy is systematically underestimated. These issues are exacerbated when applying entropy measures to natural language data, because of the especially long-tailed frequency distribution of sentences and words.

The bias issue is especially acute when doing crosslinguistic comparison with dependency corpora because the corpora available vary hugely in their sample size, from 1017 sentences of Irish to 82,451 sentences of Czech. An entropy difference between one language and another might be the result of sample size differences, rather than a real linguistic difference.

We address this issue in two ways: first, we estimate entropy using the bootstrap estimator of DeDeo et al. (2013), and apply the estimator to equally sized subcorpora across languages[1]. Second, we choose dependent and conditioning variables to minimize data sparsity and avoid long tails. In particular, we avoid entropy measures where the conditioning variable involves wordforms or lemmas. We evaluate the effects of data sparsity on our measures in Section 2.4.

2.3.2 Local subtrees

In order to cope with data sparsity and long-tailed distributions, the dependent and conditioning variables must have manageable numbers of possible values. This means that we cannot compute something like the entropy over full sentences given full dependency graphs, as these joint counts would be incredibly sparse, even if we include only part of speech information about words.

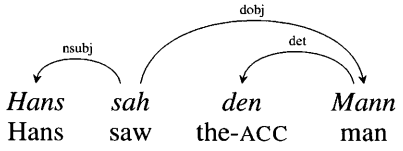
[1] At a high level, the bootstrap algorithm works by measuring entropy in the whole sample and in subsamples and uses these estimates to attempt to correct bias in the whole sample. We refer the reader to DeDeo et al. (2013) for details.

We suggest computing conditional entropy only on **local subtrees**: just subtrees consisting of a head and its immediate dependents. We conjecture that most word order and morphological rules can be stated in terms of heads and their dependents, or in terms of sisters of the same head. For example, almost all agreement phenomena in natural language involve heads and their immediate dependents Corbett (2006). Prominent and successful generative models of dependency structure such as the Dependency Model with Valence Klein and Manning (2004) assume that dependency trees are generated recursively by generating these local subtrees.

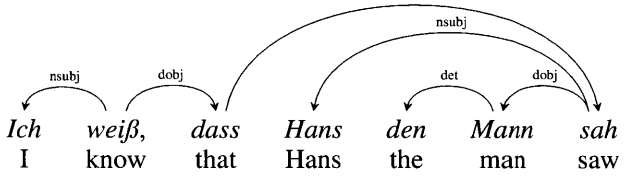
There are two shortcomings to working only with local subtrees; here we discuss how to deal with them.

First, there are certain word order phenomena which appear variable given only local subtree structure, but which are in fact deterministic given dependency structure beyond local subtrees. The extent to which this is true depends on the specifics of the dependency formalism. For example, in German, the position of the verb depends on clause type. In a subordinate clause with a complementizer, the verb must appear after all of its dependents (V-final order). Otherwise, the verb must appear after exactly one of its dependents (V2 order). If we analyze complementizers as heading their verbs, as in (3a), then the local subtree of the verb *sah* does not include information about whether the verb is in a subordinate clause or not.

(3a)



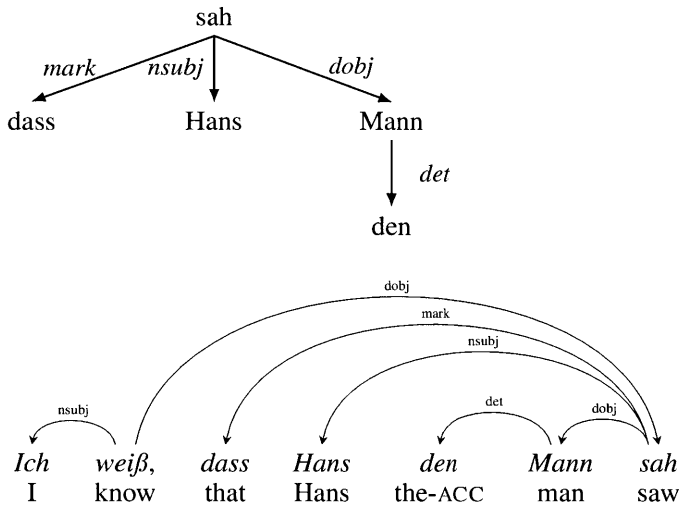
(3b)



As a result, if we measure the entropy of the order of verbal dependents conditioned

on the local subtree structure, then we will erroneously conclude that German is highly variable, since the order is either V2 or V-final and there is nothing in the local subtree to predict which one is appropriate. However, if we analyze complementizers as the dependent of their verb (as in the Universal Dependencies style, (3c)), then the conditional entropy of the verb position given local subtree structure is small. This is because the position of the verb is fully predicted by the presence in the local subtree of a *mark* relation whose dependent is *dass*, *weil*, etc.

(3c)



We deal with this issue by preferring annotation styles under which the determinants of the order of a local subtree are present in that subtree. This often means using the content-head dependency style, as in this example. When we condition on the local subtree structure and find the conditional entropy of word orders, we call this measure **Relation Order Entropy**, since we are getting the order with which relation types are expressed in a local subtree.

The second issue with looking only at local subtrees is that we miss certain word order variability associated with nonprojectivity, such as scrambling.

For example, in German subordinate clauses, the following orders are both grammatical:

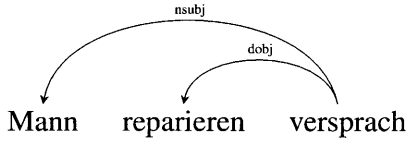
(4a) ... dass der Mann den Kühlschrank zu reparieren versprach

(4b) ... dass den Kühlschrank der Mann zu reparieren versprach

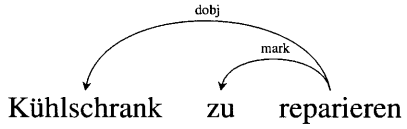
"... that the man promised to repair the refrigerator."

These sentences are both linearizations of the same overall unordered tree. However, both of them correspond to the same linearizations of the two local subtrees:

(5a)



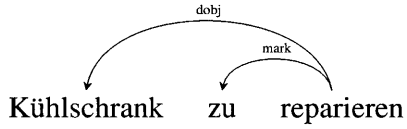
(5b)



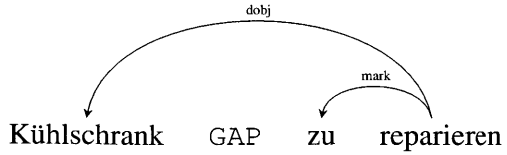
So a conditional entropy measure that only looks at the word order of local subtrees would miss this variability.

We can incorporate these nonprojectively free word orders into our entropy measures in two ways. First, we could get entropy over orders conditioned on subtrees beyond immediate subtrees. For example we could look at counts of linearizations of large subtrees. However, the height of the subtrees we would need to condition on to capture all nonprojective phenomena is potentially unbounded. Second, we could incorporate "gaps" in our representation of the order of words under a head. For example, the linearization of (5b) for the linearization in (4a) would be (6a), and for (4b) would be (6b):

(6a)



(6b)



Then taking the entropy of the orders of dependents would incorporate order freedom introduced by nonprojectivity. An array of words containing a GAP can be thought of as two **blocks** in the sense of Kuhlmann (2013); these are equivalent to **components** in the production rules of a linear context-free rewriting system. Measuring the entropy over arrays containing GAPs is the same as measuring the freedom of words to appear in different orders and to be split into different blocks, or the entropy over rule expansions containing the same words in a probabilistic mildly nonprojective dependency grammar.

For simplicity in the present work, we ignore nonprojectivity.

2.3.3 Dependency direction

Another option for dealing with data sparsity is to get conditional entropy measures over even less dependency structure. In particular we consider the case of entropy measures conditioned only on a dependent, its head, and the relation type to its head, where the dependent measure is simply whether the head is to the left or right of the dependent. This measure potentially suffers much less from data sparsity issues, since the set of possible heads and dependents in a corpus is much smaller than the set of possible local subtrees. But in restricting our attention only to head direction, we miss the ability to measure any word order freedom among sister dependents. This measure also has the disadvantage that it can miss the kind of conditioning information present in local subtrees, as described in Section 2.3.2.

When we condition only on simple dependencies, we call this measure **Head Direction Entropy**.

2.3.4 Conditioning variables

So far we have discussed our decision to use conditional entropy measures over local subtrees or single dependencies. In this setting, the conditioning variable is the unordered local subtree or dependency, and the dependent variable is the linear order of words. We now turn to the question of what information should be contained in the conditioning variable: whether it should be the full unordered tree, or just the structure of the tree, or the structure

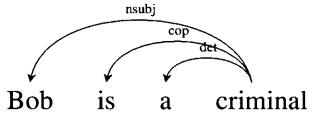
of the tree plus part-of-speech (POS) tags and relation types, etc.

In Section 2.3.1 we argued that we should not condition on the wordforms or lemmas due to sparsity issues. The remaining kinds of information available in corpora are the tree topology, POS tags, and relation types. Many corpora also include annotation for morphological features, but this is not reliably present.

Without conditioning on relation types, our entropy measures become much less linguistically useful. Much linguistic work has to do with the relationship between word order and grammatical relations; without including dependency relation types, very little information about universal grammatical relations is available. For example, if we did not condition on dependency relation types, it would be impossible to identify verbal subjects and objects or to quantify how informative word order is about these relations crosslinguistically. So we always include dependency relation type in conditioning variables.

The remaining questions are whether to include the POS tags of heads and of each dependent. Some annotation decisions in the Universal Dependencies and Stanford Dependencies argue for including POS information of heads. For example, the Universal Dependencies annotation for copular sentences has the predicate noun as the head, with the subject noun as a dependent of type *nsubj*, as in example (7):

(7)



This has the effect that the linguistic meaning of the *nsubj* relation encodes one syntactic relation when its head is a verb, and another syntactic relation when its head is a noun. So we should include POS information about heads when possible.

There are also linguistic reasons for including the POS of dependents in the conditioning variable. Word order often depends on part of speech; for example, in Romance languages, the standard order in the main clause is Subject-Verb-Object if the object is a noun but Subject-Object-Verb if the object is a pronoun. Not including POS tags in the conditioning variable would lead to misleadingly high word order freedom numbers for these clauses in these languages.

Therefore, when possible, our conditioning variables include the POS tags of heads and dependents in addition to dependency relation types.

2.3.5 Annotation style and crosslinguistic comparability

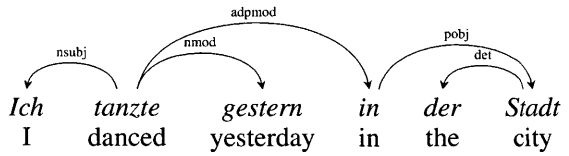
We have discussed issues involving entropy estimation and the choice of conditioning and dependent variables. Here we discuss another dimension of choices: what dependency annotation scheme to use.

Since the informativity of dependency trees about syntax and semantics affects our word order freedom measures, it is important to ensure that dependency trees across different corpora convey the same information. Certain annotation styles might allow unordered local subtrees to convey more information in one language than in another. To ensure comparability, we should use those annotation styles which are most consistent across languages regarding how much information they give about words in local subtrees, even if this means choosing annotation schemes which are less informative overall. We give examples below.

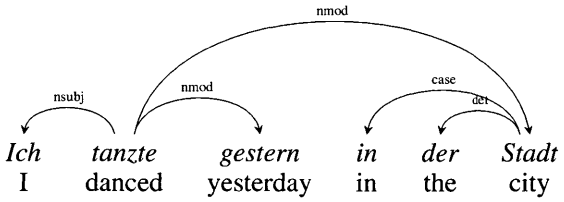
In many cases, dependency annotation schemes where function words are heads provide more information about syntactic and semantic relations, so such annotation schemes lead to lower estimates of word order freedom. For example, consider the ordering of German verbal adjuncts. The usual order is time adjuncts followed by place adjuncts. Time is often expressed by a bare noun such as *gestern* "yesterday", while place is often expressed with an adpositional phrase.

We will consider how our measures will behave for these constructions given function-word-head dependencies, and given content-head dependencies. Given function-word-head dependencies as in (8a), these two adjuncts will appear with relations *nmod* and *adpmod* in the local subtree rooted by the verb *tanzte*; their order will be highly predictable given these relation types inasmuch as time adjuncts are usually expressed as bare nouns and place adjuncts are usually expressed as adpositional phrases. On the other hand, given content-head dependencies as in (8b), the adjuncts will appear in the local subtree as *nmod* and *nmod*, and their order will appear free.

(8a)



(8b)



However, function-word-head dependencies do not provide the same amount of information from language to language, because languages differ in how often they use adpositions as opposed to case marking. In the German example, function-word-head dependencies allowed us to distinguish time adjuncts from place adjuncts because place adjuncts usually appear as adpositional phrases while time adjuncts often appear as noun phrases. But in a language which uses case-marked noun phrases for such adjuncts, such as Finnish, the function-word-head dependencies would not provide this information. Therefore, even if (say) Finnish and German had the same degree of freedom in their ordering of place adjuncts and time adjuncts, we would estimate more word order freedom in Finnish and less in German. However, using content-head dependencies, we get the same amount of information in both languages. Therefore, we prefer content-head dependencies for our measures.

Following similar reasoning, we decide to use only the universal POS tags and relation types in our corpora, and not finer-grained language-specific tags.

Using content-head dependencies while conditioning only on local subtrees overestimates word order freedom compared to function-word-head dependencies. At first glance, the content-head dependency annotation seems inappropriate for a typological study, because it clashes with standard linguistic analyses where function words such as adpositions and complementizers (and, in some analyses, even determiners (Abney, 1987)) are

heads, rather than dependents. However, content-head dependencies provide more consistent measures across languages. Therefore we present results from our measures applied to content-head dependencies.

2.3.6 Summary of parameters of entropy measures

We have discussed a number of parameters which go into the construction of a conditional entropy measure of word order freedom. They are:

1. Annotation style: function words as heads or content words as heads.
2. Whether we measure entropy of linearizations of local subtrees (*Relation Order Entropy*) or of simple dependencies (*Head Direction Entropy*).
3. What information we include in the conditioning variable: relation types, head and dependent POS, head and dependent wordforms, etc.
4. Whether to measure entropy over all dependents, or only over some subset of interest, such as subjects or objects.

The decisions for these parameters are dictated by balancing data sparsity and linguistic interpretability. We have argued that we should use content-head dependencies, and never include wordforms or lemmas in the conditioning variables. Furthermore, we have argued that it is generally better to include part-of-speech information in the conditioning variable, but that this may have to be relaxed to cope with data sparsity. The decisions about whether to condition on local subtrees or on simple dependencies, and whether to restrict attention to a particular subset of dependencies, depends on the particular question of interest.

2.3.7 Entropy measures as upper bounds on word order freedom

We initially defined an ideal measure, the entropy of word orders given full unordered dependency trees. We argued that we would have to back away from this measure by looking only at the conditional entropy of orders of local subtrees, and furthermore that we should only condition on the parts of speech and relation types in the local subtree. Here we argue that these steps away from the ideal measure mean that the resulting measures can only be interpreted as upper bounds on word order freedom.

With each step away from the ideal measure, we also move the *interpretation* of the measures away from the idealized notion of word order freedom. With each kind of information we remove from the independent variable, we allow instances where the word order of a phrase might in fact be fully deterministic given that missing information, but where we will erroneously measure high word order freedom. For example, in German, the order of verbal adjuncts is usually time before place. However, in a dependency treebank, these relations are all *nmod*. By considering only the ordering of dependents with respect to their relation types and parts of speech, we miss the extent to which these dependents *do* have a deterministic order determined by their semantics. Thus, we tend to overestimate true word order freedom.

On the other hand, the conditional entropy approach do not in principle *under*estimate word order freedom as we have defined it. The conditioning information present in a dependency tree represents only semantic and syntactic relations, and we are explicitly interested in word order variability beyond what can be explained by these factors. Therefore, our word order freedom measures constitute upper bounds on the true word order freedom in a language.

Underestimation can arise due to data sparsity issues and bias issues in entropy estimators. For this reason, it is important to ensure that our measures are stable with respect to sample size, lest our upper bound become a lower bound on an upper bound.

The tightness of the upper bound on word order freedom depends on the informativity of the relation types and parts of speech included in a measure. For example, if we use a system of relation types which subdivides *nmod* relations into categories like *nmod:tmod* for time phrases, then we would not overestimate the word order freedom of German verbal adjuncts. As another example, to achieve a tighter bound for a limited aspect of word order freedom at the cost of empirical coverage, we might restrict ourselves to relation types such as *nsubj* and *dobj*, which are highly informative about their meanings.

2.4 Applying the measures

Here we give the results of applying some of the measures discussed in Section 2.3 to dependency corpora. We use the dependency corpora of the HamleDT 2.0 (Zeman et al., 2012; Rosa et al., 2014) and Universal Dependencies 1.0 (Nivre et al., 2015). All punctuation and dependencies with relation type *punct* are removed. We only examine sentences with a single root. We exclude corpora with less than 1000 such sentences. Annotation was normalized to content-head format when necessary. Combined this gives us dependency corpora of 34 languages in a fairly standardized format.

In order to evaluate the stability of our measures with respect to sample size, we measure all entropies using the bootstrap estimator of DeDeo et al. (2013). We report the mean results from applying our measures to subcorpora of 1000 sentences for each corpus. We also report results from applying measures to the full corpus, so that the difference between the full corpus and the subcorpora can be compared, and the effect of data sparsity evaluated.

2.4.1 Head Direction Entropy

Head direction entropy, defined and motivated in Section 2.3.3, is the conditional entropy of whether a head is to the right or left of a dependent, conditioned on relation type and part of speech of head and dependent. This measure can reflect either consistency in head direction conditioned on relation type, or consistency in head direction *overall*. Results from this measure are shown in Figure 2-2. As can be seen, the measure gives similar results when applied to subcorpora as when applied to full corpora, indicating that this is measure is not unduly affected by differences in sample size.

We find considerable variability in word order freedom with respect to head direction. In languages such as Korean, Telugu, Irish, and English, we find that head direction is nearly deterministic. On the other hand, in Slavic languages and in Latin and Ancient Greek we find great variability. The fact that entropy measures on subcorpora of 1000 sentences do not diverge greatly from entropy measures on full corpora indicates that this measure is stable with respect to sample size.

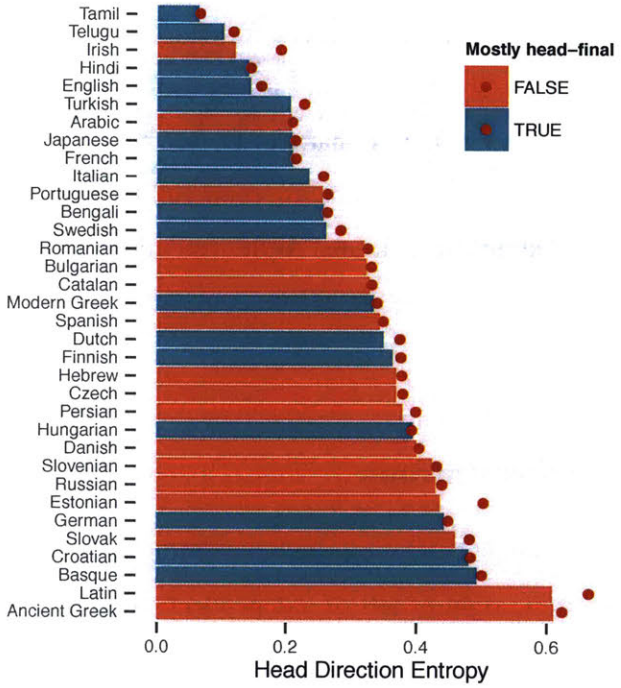


Figure 2-2: Head direction entropy in 34 languages. The bar represents the average magnitude of head direction entropy estimated from subcorpora of 1000 sentences; the red dot represents head direction entropy estimated from the whole corpus.

We find a potential relationship between predominant head direction and word order freedom in head direction. Figure 2-2 is coded according to whether languages have more than 50% head-final dependencies or not. The results suggest that languages which have highly predictable head direction might tend to be mostly head-final languages.

The results here also have bearing on appropriate generative models for grammar induction. Common generative models, such as DMV, use separate multinomial models for left and right dependents of a head. Our results suggest that for some languages there should be some sharing between these distributions.

2.4.2 Relation Order Entropy

Relation order entropy (Section 2.3.2) is the conditional entropy of the order of words in a local subtree, conditioned on the tree structure, relation types, and parts of speech. Figure 2-3 shows relation order entropy for our corpora. As can be seen, this measure is highly sensitive to sample size: for corpora with a medium sample size, such as English (16535 sentences), there is a moderate difference between the results from subcorpora and the results from the full corpus. For other languages with comparable size, such as Spanish (15906 sentences), there is a larger difference. In the case of languages with small corpora such as Bengali (1114 sentences), their true relation order entropy is almost certainly higher than measured.

While relation order entropy is the most easily interpretable and general measure of word order freedom, it does not seem to be workable given current corpora and methods. In further experiments, we found that removing POS tags from the conditioning variable does not reduce the instability of this measure.

2.4.3 Relation Order Entropy of subjects and objects

We can alleviate the data sparsity issues of relation order entropy by restricting our attention to a few relations of interest. For example, the position of subject and object in the main clause has long been of interest to typologists (Greenberg, 1963), (cf. (Dryer, 1992)). In Figure 2-4 we present relation order entropy of subject and object for local subtrees con-

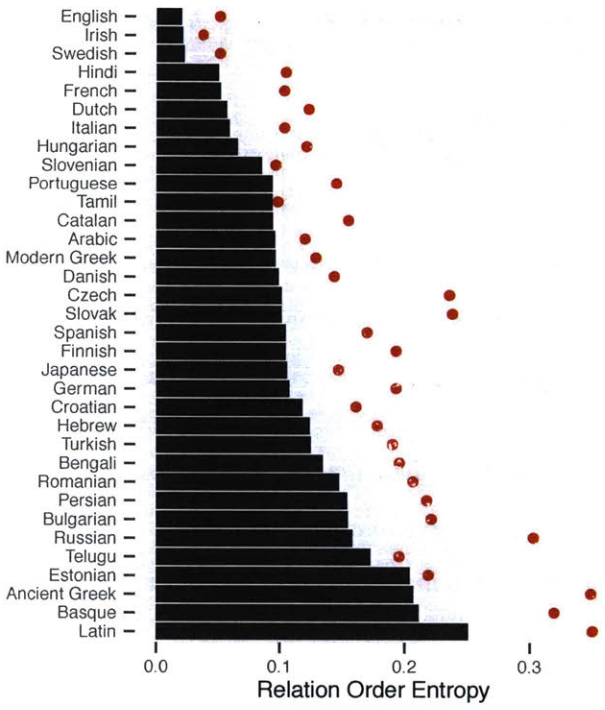


Figure 2-3: Relation order entropy in 34 languages. The bar represents the average magnitude of relation order entropy estimated from subcorpora of 1000 sentences; the red dot represents relation order entropy estimated from the whole corpus.

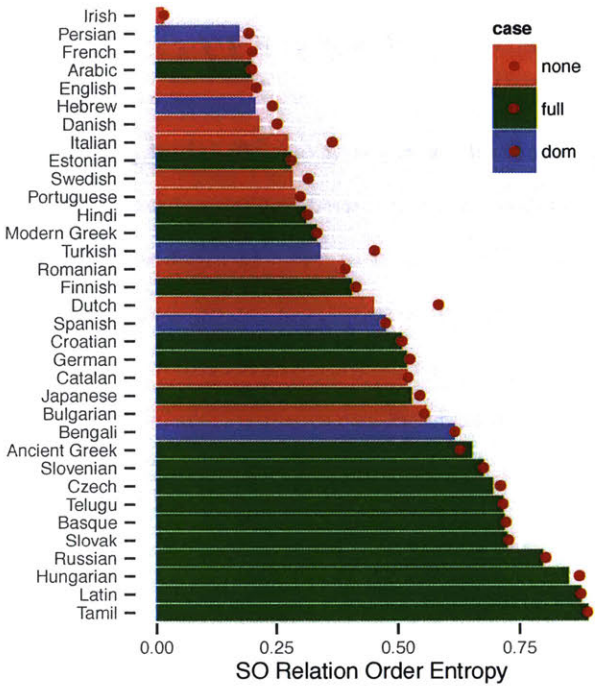


Figure 2-4: Relation order entropy for subject and object in 34 languages. Language names are annotated with corpus size in number of sentences. Bars are colored depending on the nominative-accusative case marking system type for each language. "Full" means fully present case marking in at least one paradigm. "dom" means Differential Object Marking.

taining relations of type *nsubj* and *dobj* (*obj* in the case of HamleDT corpora), conditioned on the parts of speech for these dependents.

The languages Figure 2-4 are colored according to their nominative-accusative[2] case marking on nouns. We consider a language to have full case marking if it makes a consistent morphological distinction between subject and object in at least one paradigm. If the distinction is only present conditional on animacy or definiteness, we mark the language as DOM for Differential Object Marking (Aissen, 2003).

The figure reveals a relationship between morphology and this particular aspect of word order freedom. Languages with relation order entropy above .625 all have relevant case marking, so it seems word order freedom in this domain implies the presence of case marking. However, case marking does not imply rigid word order; several languages in the

[2]Or ergative-absolutive in the case of Basque and the Hindi past tense.

sample have rigid word order while still having case marking. Our result is a quantitative sharpening of the pattern claimed in Kiparsky (1997).

Interestingly, many of the exceptional languages—those with case marking and rigid word order—are languages with verb-final or verb-initial orders. In our sample, Persian, Hindi, and Turkish are case-marking verb-final languages where we measure low levels of freedom in the order of subject and object. Modern Standard Arabic is (partly) verb-initial and case-marking (although case marking is rarely pronounced or explicitly written in modern Arabic). This finding is in line with recent work (Gibson et al., 2013; Futrell et al., 2015a) which has suggested that verb-final and verb-initial orders without case marking do not allow robust communication in a noisy channel, and so should be dispreferred.

2.5 Conclusion

We have presented a set of interrelated methodological and linguistic issues that arise as part of quantifying word order freedom in dependency corpora. We have shown that conditional entropy measures can be used to get reliable estimates of variability in head direction and in ordering relations for certain restricted relation types. We have argued that such measures constitute upper bounds on word order freedom. Further, we have demonstrated a simple relationship between morphological case marking and word order freedom in the domain of subjects and objects, providing to our knowledge the first large-scale quantitative validation of the old intuition that languages with free word order must have case marking.

Chapter 3

Large-scale Evidence for Dependency Length Minimization

3.1 Introduction

Finding explanations for the observed variation in human languages is the primary goal of linguistics, and promises to shed light on the nature of human cognition.[1] One particularly attractive set of explanations is functional in nature, holding that language universals are grounded in the known properties of human information processing (Haspelmath, 2008; Jaeger and Tily, 2011). The idea is that grammars of languages have evolved so that language users can communicate using sentences that are relatively easy to produce and comprehend. Within the space of functional explanations, a promising hypothesis is dependency length minimization (DLM). The aim of this paper is to provide corpus evidence from over 40 languages for DLM as a universal pressure affecting both grammar and usage. Section 3.2 covers results that were previously published in Futrell et al. (2015b), though it is largely rewritten. Sections 3.3 and 3.4 are new material except for Section 3.3.1 which was published in Futrell and Gibson (2015).

3.1.1 Background

This study is about dependency length: the distances between linguistic heads and dependents. The notions of head and dependent can be defined on top of most syntactic formalisms. Nearly all theories of syntax include some notion of headedness, the idea that the behavior of a constituent can be understood primarily with reference to one distinguished word, the **head** (Bloomfield, 1933; Tesnière, 1959; Hays, 1964; Bresnan, 1982; Hudson, 1990b; Pollard and Sag, 1987; Mel'čuk, 1988; Corbett et al., 1993). For example, the syntactic behavior of a noun phrase is determined primarily by the head noun in the phrase. Headedness in syntax is also known as **endocentricity**. A **dependent** is a word that modifies a head, and a **dependency** is the relationship between a head and a dependent.

While most constituents appear to be endocentric, not all syntactic formalisms posit a head for all phrases. In these formalisms, some constructions forming constituents are **exocentric**, having no head. For example, it is notoriously difficult to assign a head to a phrase

[1] Code for replicating the results in this section can be found online at `http://github.com/Futrell/cliqs`.

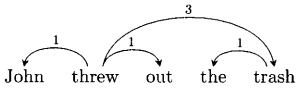
such as *Bob and Mary* (Temperley, 2005; Popel et al., 2013), with different dependency formalisms choosing different means (Tesnière, 1959; Mel'čuk, 1988), and some introducing elements of phrase structure formalisms especially for thus purpose Hudson (1990a). In formalisms such as Minimalism, all phrases have heads, but these heads may be silent elements (Adger, 2003). Thus while the notions of head and dependent exist in most formalisms, they are not always present or straightforward. Nevertheless, most phrases are uncontroversially endocentric.

Taking the notion of endocentricity to it slogical conclusion, **dependency grammar** posits that syntax can be fully described solely in terms of relationships among heads, without a further notion of constituent or other higher-order groupings of words (Tesnière, 1959; Hays, 1964; Hudson, 1990b; Mel'čuk, 1988; Sleator and Temperley, 1991). In dependency grammar, the correct syntactic analysis takes the form of a tree or directed graph linking heads to their dependents. If all phrases are endocentric, then constituency grammars and dependency grammars can be freely converted one to the other. But if endocentricity is not universal, then it is likely that dependency formalisms will miss some syntactic constraints that can be expressed in constituency grammars.

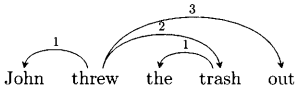
In this work, while we use a dependency formalism, we do not wish to claim that dependency grammar is the only correct description of syntax, or that a dependency tree encapsulates all the syntactic information that there is to know about a sentence. We only wish to claim that dependency trees represent an important and large subset of that information. We describe syntax in terms of dependency trees here for two reasons: simplicity and convenience. With regard to simplicity, dependency trees are simple to reason about, and to formulate algorithms over, while providing a decent description of syntax. With regard to convenience, large-scale corpora are available with dependency annotation, because it is easier to perform this annotation in a consistent way across languages than to use a constituency annotation (Nivre, 2005).

Examples of dependency trees are given in Figure 3-1. The verb *throw* in Sentence C is the head of two nouns that modify it, *John*—its subject—and *trash*—its object. Subject and object relations are kinds of dependency relations.

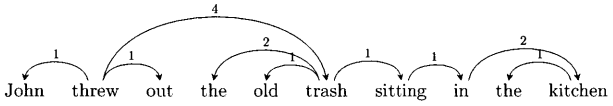
Another way to think about dependency is to note that heads and dependents are words



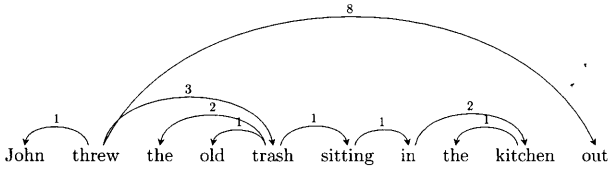
Sentence A: Total dependency length = 6



Sentence B: Total dependency length = 7



Sentence C: Total dependency length = 14



Sentence D: Total dependency length = 20

Figure 3-1: Four sentences along with their dependency representations. The number over each arc represents the length of the dependency in words. The total dependency length is given below each sentence. Sentences **A** and **B** have the same semantics, and either word order is acceptable in English; English speakers typically do not find one more natural than the other. Sentences **C** and **D** also both have the same semantics, but English speakers typically find **C** more natural than **D**.

which must be linked together in order to understand a sentence, to a first approximation. For example, in order to correctly understand Sentence C in Figure 3-1, a comprehender must determine that a relationship of adjectival modification exists between the words *old* and *trash*, and not between, say, the words *old* and *kitchen*. In typical dependency analyses, objects of prepositions (*him* in *for him*) depend on their prepositions, articles depend on the nouns they modify, and so on[2].

The **dependency length minimization** (DLM) hypothesis is that language users prefer word orders which minimize dependency length. The hypothesis makes two broad predictions. First, when the grammar of a language provides multiple ways to express an idea, language users will prefer the expression with the shortest dependency length. Second, grammars should facilitate the production of short dependencies by not enforcing word orders with long dependencies (Rijkhoff, 1990; Hawkins, 1990).

Explanations for *why* language users would prefer short dependencies are various, but they all involve the idea that short dependencies are easier or more efficient to produce and comprehend than long dependencies (Hawkins, 1994; Gibson, 1998). The difficulty of long dependencies emerges naturally in many models of human language processing. For example, in a left-corner parser or generator, dependency length corresponds to a timespan over which a head or dependent must be held in a memory store (Abney and Johnson, 1991; Gibson, 1991; Resnik, 1992); since storing items in memory may be difficult or error-prone, short dependencies would be easier and more efficient to produce and parse according to this model. In support of this idea, comprehension and production difficulty have been observed at the sites of long dependencies (Gibson, 1998; Grodner and Gibson, 2005; Demberg and Keller, 2008; Shain et al., 2016) (cf. Gennary and MacDonald, 2008).

In terms of the framework from the introduction, if dependencies represent instances of context-dependence in semantic interpretation, then they are undesirable under memory constraints for reasons discussed in Section 1.5.5 above.

If language users are motivated by avoiding difficulty, then they should avoid long dependencies. Furthermore, if languages have evolved to support easy communication, then

[2]Most aspects of dependency analysis are generally agreed upon, although the analysis of certain relations has been in dispute, primarily those relations involving function words such as prepositions, determiners, and conjunctions.

they should not enforce word orders that create long dependencies. The DLM hypothesis thus provides a link between language structure and efficiency through the idea that speakers and languages find ways to express meaning while avoiding structures which are difficult to produce and comprehend.

3.1.2 Four predictions of Dependency Length Minimization

Over the last quarter century, researchers have proposed DLM-based explanations of some of the most pervasive properties of word order in languages. We can see the word order in a sentence as a particular *linearization* of a dependency graph, where a linearization is an arrangement of the words of the dependency graph in a certain linear order. For instance, Sentences **A** and **B** in Figure 3-1 are two linearizations of the same graph.

The predictions of DLM as a theory of linearization are potentially complex. We must consider, for a dependency tree, what is the linearization of the tree that minimizes dependency length (Harper, 1964; Iordanskii, 1974; Chung, 1984). However, four generalizations about minimal dependency length linearizations have emerged which can guide predictions about word order:

1. **Projectivity.** In a minimal dependency length linearization, when dependency arcs are drawn above a sentence, the lines rarely cross (Ferrer i Cancho, 2006). An example is shown in Figure 3-2.

2. **Head direction consistency.** In low-arity dependency trees, consistency in head direction should be preferred. This prediction is motivated by examples in Figure 3-3.

3. **Ordered nesting.** When a head has multiple dependents, they should be arranged in order of decreasing length before a head, and increasing length after a head. The preference for sentence C over sentence D in Figure 3-1 is an example of this principle.

4. **Mixed branching.** In high-arity trees, where one dependent is much shorter than the another, the short dependent should be on the opposite side of the head from the long

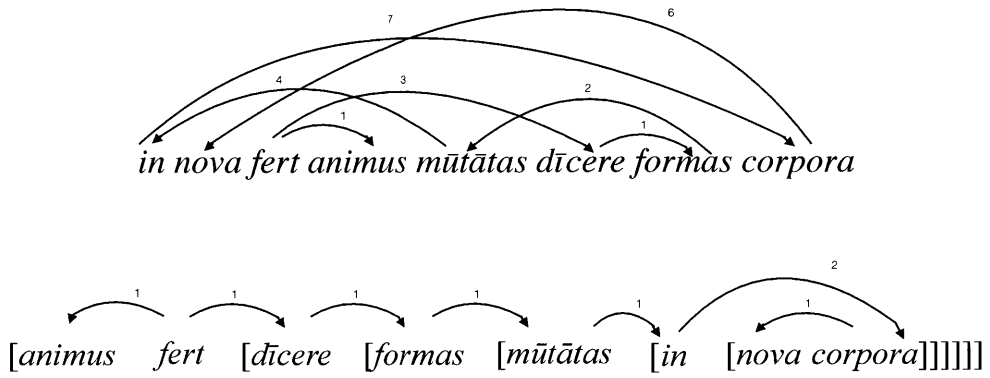


Figure 3-2: On average, projective linearizations have shorter dependency length than non-projective ones. This example shows a line from Ovid in its original word order, compared with a projective linearization of the same tree. Dependency length for the projective linearization is substantially shorter.

one Gildea and Temperley (2007); Temperley (2007, 2008).

The prediction that linearizations are projective is borne out pervasively across languages, to the extent that projectivity has often been incorporated as an explicit constraint on dependency representations (Gaifman, 1965; Mel'čuk, 1988). Nevertheless, there are compelling examples where dependencies seem to cross (Bresnan et al., 1982; Joshi, 1990; Kuhlmann and Nivre, 2006; Chen-Main and Joshi, 2010). Ferrer i Cancho (2006) argues that this ubiquitous property of languages arises from DLM, because orders that minimize dependency length have a small number of crossing dependencies on average.

Also, the second of these generalizations can explain a pervasive word order universal. Greenberg (1963) found striking correlations between different ordering constraints in languages, such that languages tend to be consistent in whether heads come before depen-

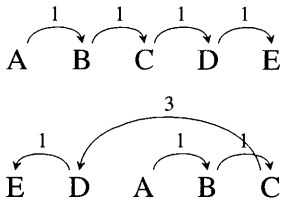


Figure 3-3: An example of how DLM prefers linearizations with consistent head direction for low-arity trees. Dependency length (number of words from head to dependent) is drawn over each arc. The first linearization has longer sum dependency length than the second.

dents or vice versa (Lehmann, 1973; Vennemann, 1974; Radford, 1997). This pattern bears out the head direction consistency prediction Hawkins (1994). Furthermore, exceptions to it are typically for single-word phrases, thus bearing out the mixed branching prediction (Dryer, 1992; Temperley, 2007).

Minimal dependency length has also been widely assumed as a reliable generalization in the field of natural language processing. For example, most state-of-the-art models for natural language grammar induction incorporate a bias toward positing short dependencies, and their performance is greatly improved by this assumption (Klein and Manning, 2004; Smith and Eisner, 2006; Noji et al., 2016). The influential grammar induction model of Klein and Manning (2004) only achieved results above the random baseline after incorporating an assumption of minimal dependency length. Seminal parsing algorithms also incorporate this assumption (Sleator and Temperley, 1991; Collins, 2003; Eisner and Smith, 2005).

3.1.3 Evidence for Dependency Length Minimization

While DLM *can* explain high-level syntactic generalizations about languages, it has not yet been shown conclusively to be the correct explanation. Other independently-motivated explanations exist for many of the linguistic properties attributed to DLM. For example, head direction consistency could be motivated by simplicity in grammmars, and projectivity could be motivated by time complexity in parsing. Furthermore, the argument for DLM as a pressure in usage is limited to a few languages, mostly English. If it turns out that there is not a universal usage preference for sentences with short dependencies, then that would

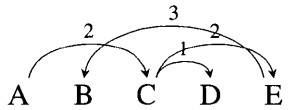
weaken the case that DLM is the correct explanation of grammatical universals.

With regard to the DLM predictions about grammar, of the four predictions of DLM, consistent head direction and projectivity have the most validation. In addition, Hawkins (2014) has argued that crosslinguistic grammars also show the preference for ordered nesting. On the usage side, evidence for all four of the DLM predictions as production preferences has been provided in detailed corpus studies that investigate these predictions explicitly. Such studies exist for English, German, and Romance languages (Hawkins, 1994; Wasow, 2002; Temperley, 2007; Gulordava and Merlo, 2015b). Tily (2010) also shows that dependency length becomes more minimized over time in historical corpora of English. In addition, Yamashita and Chang (2001) gives experimental evidence for a nested ordering (long-before-short order) preference in Japanese usage.

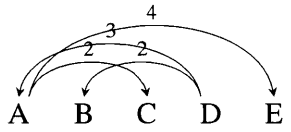
In addition to corpus studies that explicitly test the four predictions of DLM, a number of corpus studies have attempted to show very general evidence for DLM in grammar and/or usage by comparing observed dependency length to random baselines, representing a hypothetical state of language unaffected by DLM. These approaches have the advantage of being extensible in principle to any language for which a suitable corpus exists, without requiring in-depth construction-by-construction analysis.x These studies have come in two types: those that compare observed dependency length to dependency length in random trees, and those that compare to dependency length in random reorderings of observed trees. We will call these two approaches **random tree** and **random order** approaches.

Random tree approaches include Liu (2008) and Ferrer i Cancho and Liu (2014). In these approaches, observed dependency trees are compared to random dependency trees generated using various algorithms, such as Prüfer codes (Prüfer, 1918). In these approaches the dependency length in a sentence such as Sentence A of Figure 3-1 is compared to dependency length in random trees of the same length, as shown in Figure 3-4. Using this approach, Liu (2008) finds that dependency length in real trees of 20 languages is shorter than dependency length in random trees. Subsequent work has focused on finding random tree generation algorithms which produce dependency length distributions similar to natural language, as a way of explaining these distributions (Lu et al., 2016).

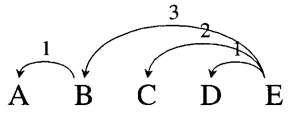
The comparison to random trees tells us that dependency length minimization exists in



Total dependency length = 8



Total dependency length = 11



Total dependency length = 7

Figure 3-4: Some random trees based on the sentence in Figure 3-1 according to random tree baseline used in Liu (2008).

a very general sense, but not that it has an effect on word orders. The result is compatible with many mechanisms to reduce dependency length. For example, language users might structure discourse (splitting ideas into multiple sentences), drop optional syntactic elements such as pronouns, or choose favorable word orders. A speaker has many degrees of freedom in choosing what sentences to use, and could be using these to reduce dependency length without DLM affecting word order at all.

The other strand of large-scale corpus work on DLM compares to random order baselines. In these baselines, the observed *unordered* dependency tree structures are held constant, and the word order is allowed to vary according to constraints under study. The result tells us the extent to which DLM affects word order, in addition to how it can affect tree structure and content expressed. Both the content expressed in a sentence and word order preferences contribute to the dependency length of the sentence. Comparing to random tree baselines, we see that dependency length is shorter than what we would expect if both of these factors are allowed to vary. Comparing to random order baselines, we are holding tree structure—a proxy for content expressed in a sentence—fixed, and letting order vary. This approach allows us to isolate the effect of DLM on word order, and give evidence for DLM as a pressure affecting word order specifically.

In the random order approach, most prominently, Gildea and Temperley (2007, 2010) compare observed dependency length to random projective reorderings of trees, answering the question of whether the observed dependency length in real sentences can be explained by projectivity alone. They find that dependency length is shorter than expected from the random projective baseline in English and German, and the result is replicated in Park and Levy (2009). Yet while they find a statistically significant DLM effect in both languages, the authors find that DLM is much weaker in German. In suggestive related work, Noji and Miyao (2014) show that memory usage in a specific parser is minimzed for corpora of 18 languages when compared to random reorderings, but they do not test the question of dependency length minimization directly. Overall, as large-scale evidence for DLM effects on word order goes, the results are mixed: English shows optimization, German only barely so.

3.1.4 Aims of this work

In this paper, we provide large-scale corpus evidence that DLM is in fact a universal pressure affecting word order in both grammar and usage across languages. In Section 3.2, we show that dependency length in corpora of dozens of languages is shorter than what we would expect from independently-motivated constraints for projectivity, consistent head direction, and word order fixedness as a function of grammatical relations. This result establishes that DLM explains a strict superset of the word order phenomena that these other constraints can explain. In Section 3.3, we argue that DLM affects both grammar and usage in these languages. To make this argument, we induce probabilistic grammars that take a dependency tree in a language and give the probability distribution over licit linearizations of that tree in the language. We then show that observed dependency lengths in sentences are shorter than random grammatical reorderings of those sentences, establishing a usage preference for short dependencies beyond what is encoded by grammar. Furthermore, we show that the random grammatical reorderings of sentences themselves have lower dependency length than the reorderings according to constraints such as projectivity. This result establishes that word order in grammars are also shaped by DLM. Finally, in Section 3.4,

we discuss observed variation in dependency length across languages. While all languages in the current sample have dependency length shorter than baselines, they vary significantly in the extent of minimization. We discuss some linguistic properties that appear to condition this variation, and speculate on how this variation can shed light on constraints other than DLM that shape natural languages.

3.2 Comparison with independently motivated baselines

Here we address the question of whether the observed dependency length in corpora of many languages can be accounted for by independently-motivated constraints for projectivity, fixed word order, and consistent head direction. If dependency length can be fully accounted for by these independent factors, then that result would diminish the evidence for DLM as an explanation of word order universals. On the other hand, if dependency length is shorter than expected from these constraints, then the word order phenomena they account for is only a subset of what DLM can explain. This result would strengthen the argument for DLM as a causal force affecting languages. The results of this section can be taken equally to mean DLM affects grammar and usage; we do not distinguish between these two here.

To address our question, we use recently-available dependency-parsed corpora of many languages (McDonald et al., 2013; Zeman et al., 2014; Nivre et al., 2015). We obtained hand-parsed or hand-corrected corpora of 37 languages, comprising 10 language families. 36 of the corpora follow widely recognized standards for dependency analysis (de Marneffe et al., 2014; Nivre et al., 2015); the remaining corpus (Mandarin Chinese) uses its own system which is nonetheless similar to the standards. The texts in the corpora are for the most part written prose from newspapers, novels, and blogs. Exceptions are the corpora of Latin and Ancient Greek, which include a great deal of poetry, and the corpus of Japanese, which consists of spoken dialogue.

In addition to the random baselines, we present an optimal baseline for the minimum possible dependency length in a projective linearization for each sentence, following the method of Gildea and Temperley (2007). This allows us to evaluate the *extent* to which

different languages minimize their dependency lengths compared to what is possible. We do not expect observed dependency lengths to be completely minimized, since there are other factors influencing grammars and language usage which might come into conflict with DLM.

3.2.1 Methods

Data

We use the dependency trees of the HamleDT 2.0, Google Universal Treebank 2.0, and Universal Dependencies 1.0 corpora (McDonald et al., 2013; Zeman et al., 2014; Nivre et al., 2015); these are projects which have aimed to harmonize details of dependency analysis between dependency corpora. In addition we include a corpus of Mandarin, the Chinese Dependency Treebank (Che et al., 2012). We normalize the corpora so that prepositional objects depend on their prepositions (where the original corpus has a *case* relation) and verbs depend on their complementizers (where the original corpus has a *mark* relation). For conjunctions, we use Stanford style. We also experimented with corpora in the original content-head format of HamleDT and Universal Dependencies; the pattern of results and their significance was the same. These results are shown in Appendix A.

Measuring dependency length

We calculate the length of a single dependency arc as the number of words between a head and a dependent, including the dependent, as in Figure 3-1. For sentences, we calculate the overall dependency length by summing the lengths of all dependency arcs. We do not count any nodes representing punctuation or "root" nodes, nor arcs between them; sentences that are not singly rooted after removal punctuation are excluded.

Fixed Word Order Random Baseline

Fixed word order random linearizations are generated according to the following procedure per sentence. Assign each relation type a random weight in $[-1, 1]$. Starting at the root node, collect the head word and its dependents and order them by their weight, with the

head receiving weight 0. Then repeat the process for each dependent, keeping the same weights. This creates consistency in word order with respect to relation types.

This linearization scheme can capture many aspects of fixed order in languages, but cannot capture all of them; for example, linearization order in German depends on whether a verb is in a subordinate clause or not. The fixed linearization scheme is also inaccurate in that it produces entirely deterministic orders. In contrast, many languages permit the speaker a great deal of freedom in choosing word order. However, creating a linearization model that can handle all possible syntactic phenomena is beyond the scope of this paper.

Generalized Additive Models

For the figures, we present fits from Generalized Additive Models predicting dependency length from sentence length using cubic splines as a basis function. This provides a line which is relatively close to the data for visualization.

Regression Models

For hypothesis testing and comparison of effect sizes, we use regression models fit to data from each language independently. For these regressions, we only consider sentences with length < 100 words. For each sentence s in a corpus, we have $N + 1$ datapoints: 1 for the observed dependency length of the sentence, and $N = 100$ for the dependency lengths of the random linearizations of the sentence's dependency tree. We fit a mixed-effects regression model (Gelman and Hill, 2007) with the following equation, with coefficients β representing fixed effects and coefficients S representing random effects by sentence:

$$\hat{y}_i = \beta_0 + S_0 + \beta_1 l_s^2 + (\beta_2 + S_2) r_i + \beta_3 r_i l_s^2 + \epsilon_i \tag{3.1}$$

where $\hat{y}_i$ is the estimated total dependency length of datapoint i, β_0 is the intercept, l_s^2 is the squared length of sentence s in words, r_i is an indicator variable with value 1 if datapoint i is a random linearization and 0 if it is an observed linearization, and m_i is an indicator variable with value 1 if datapoint i is a minimal linearization and 0 if it is an observer linearization. We use l_s^2 rather than l_s because we found that a model using

squared sentence length provides a better fit to the data for $33/37$ languages, as measured by AIC and BIC; the pattern and significance of the results are the same for a model using plain sentence length rather than squared sentence length. The coefficient β_3 determines the extent to which dependency length of observed sentences grows more slowly with sentence length than dependency length of randomly linearized sentences. This growth rate is the variable of interest for DLM; summary measures which are not a function of length fall prey to inaccuracy due to mixing dependencies of different lengths (Ferrer i Cancho and Liu, 2014). For significance testing comparing the real dependencies and random baselines, we performed a likelihood ratio test comparing models with and without β_3. We fit the model using the `lme4` package in R (Bates et al., 2015).

3.2.2 Results

Free Word Order Baseline Our first baseline is fully random projective linearizations of dependency trees. Random projective linearizations are generated according to the following procedure, from Gildea and Temperley (2007), a method similar to one developed by Hawkins (1998). Starting at the root node of a dependency tree, collect the head word and its dependents and order them randomly. Then repeat the process for each dependent. For each sentence in our corpora, we compare real dependency lengths to dependency lengths from 100 random linearizations produced using this algorithm. Note that the 100 random linearization all have the same underlying dependency structure as the original sentence, just with a potentially different linear order. Under this procedure, the random linearizations do not obey any particular word order rules: there is no consistency in whether subjects precede or follow verbs, for example. In that sense, these baselines may most closely resemble a free word order language as opposed to a language like English, in which the order of words in sentences are relatively fixed.

Figure 3-5 shows observed and random dependency lengths for sentences of length 1–50. As the figure shows, all languages have average dependency lengths shorter than the random baseline, especially for longer sentences. To test the significance of the effect, for each language, we fit regression models predicting dependency length as a function of

sentence length. The models show a significant effect where the dependency length of real sentences grows more slowly than the dependency length of baseline sentences ($p < 0.0001$ for each language).

Figure 3-6 shows histograms of observed and random dependency lengths for sentences of length 12, the shortest sentence length to show a significant effect in all languages ($p < 0.01$ for Latin, $p < 0.001$ for Telugu, and $p < 0.0001$ for all others, by Stouffer's method). In languages for which we have sufficient data, there is a significant DLM effect for all longer dependency lengths.

Fixed Word Order Baseline The first baseline ignores a major common property of languages: that word order is typically fixed for certain dependency types. For example, in English, the order of certain dependents of the verb is mostly fixed: the subject of the verb almost always comes before it, and the object of a verb almost always comes after. We capture this aspect of language by introducing a new baseline. In this baseline, the relative ordering of the dependents of a head is fixed given the *relation types* of the dependencies (subject, object, prepositional object, etc.). For each sentence, we choose a random ordering of dependency types, and linearize the sentence consistently according to that order. We perform this procedure 100 times to generate 100 random linearizations per sentence.

Figure 3-7 shows observed dependency lengths compared to the random fixed-order baselines. The results are similar to the comparison with the free-word-order baselines in that all languages have dependencies shorter than chance, especially for longer sentences. We find that this random baseline is more conservative than the free-word-order baseline in that the average dependency lengths of the fixed word order random baselines are shorter than those of the free word order random baselines (with significance $p < 0.0001$ by a t-test in each language). For this baseline, the DLM effect as measured in the regression model is significant at $p < 0.0001$ in all languages except Telugu, a small corpus lacking long sentences, where $p = 0.15$.

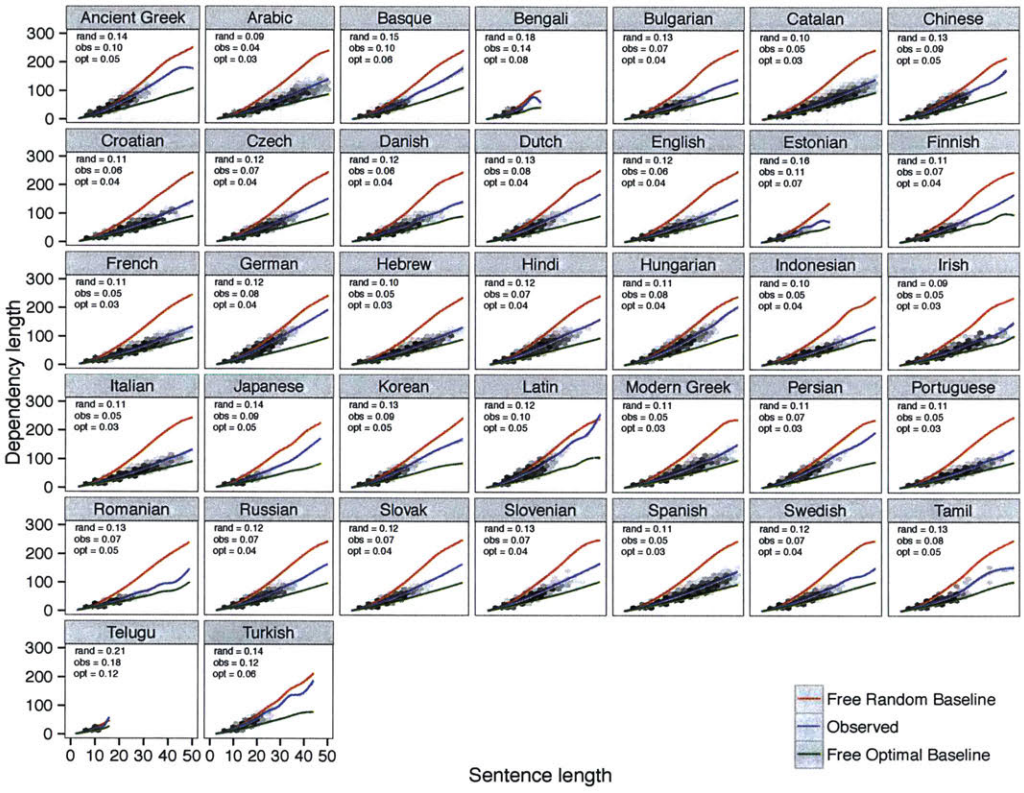


Figure 3-5: Random **Free Word Order** baseline dependency lengths, observed dependency lengths, and optimal dependency lengths for sentences of length 1–50. The blue line shows observed dependency length, the red line shows average dependency length for the random Free Word Order baseline, and the green line shows average dependency length for the optimal baseline. The density of observed dependency lengths is shown in black. The lines in this figure are fit using a generalized additive model. We also give the slopes of dependency length as a function of squared sentence length, as estimated from a mixed-effects regression model. `rand` is the slope of the random baseline. `obs` is the slope of the observed dependency lengths. `opt` is the slope of the optimal baseline. Due to varying sizes of the corpora, some languages (such as Telugu) do not have attested sentences at all sentence lengths.

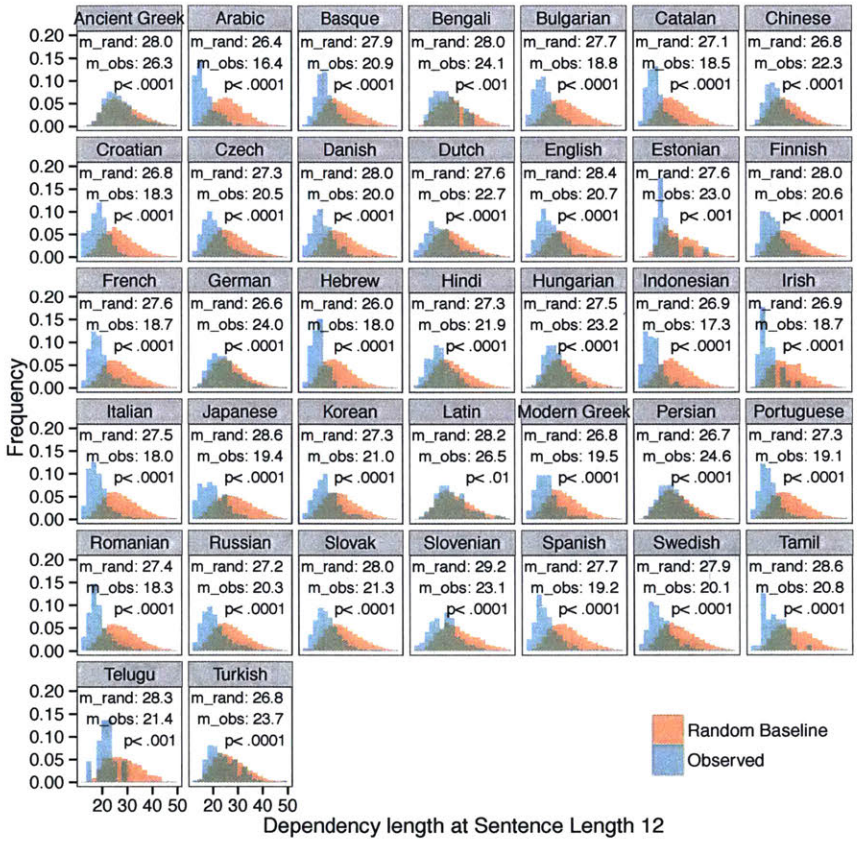


Figure 3-6: Histograms of observed dependency lengths and Free Word Order random baseline dependency lengths for sentences of length 12. `m_rand` is the mean of the free word order random baseline dependency lengths; `m_obs` is the mean of observed dependency lengths. We show p values from Stouffer's Z-transform test comparing observed dependency lengths to the dependency lengths of the corresponding random linearizations.

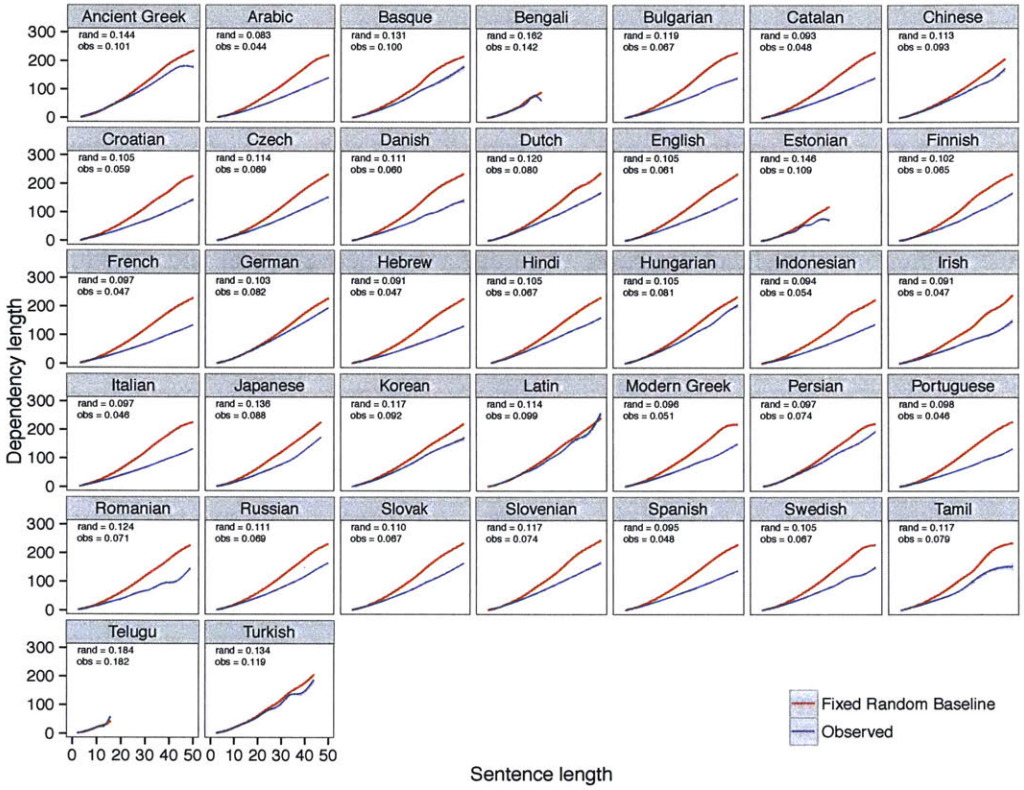


Figure 3-7: Real dependency lengths as a function of sentence length (blue), compared to the **Fixed Word Order** Random baseline (red). GAM fits are shown. `rand` and `obs` are the slopes for random baseline and observed dependency length as a function of squared sentence length, as in Figure 3-5.

3.2.3 Consistent Head Direction Baseline

There is the possibility that our findings actually reflect independently-motivated consistency in head order rather than DLM per se. Here we test this idea by comparing languages to random and optimal baselines where head direction is fixed for all relation types. In this case, the only way that dependency length can be minimized is by choosing an optimal ordering of the dependents of a single head; this is accomplished by ordering constituents from short to long in the case of a head-initial language, or from long to short in the case of a head-final language.

Figure 3-8 shows real dependency lengths compared to the consistent-head-direction baselines. We find that all languages have shorter dependencies than we would expect by chance given consistent head direction. The difference between real and random slopes is significant at $p < 0.001$ for all languages. The baseline is especially interesting in the case of the overwhelmingly head-final languages in our sample, such as Japanese, Korean, Turkish, Telugu, Tamil, and Hindi. For these languages, which are similar to the baselines in the consistency of their head direction, the fact that they have dependency lengths shorter than the random baseline indicates that they accomplish dependency length minimization through long-before-short order.

3.2.4 Fixed Head Position Baseline

To what extent is DLM accomplished by choosing an optimal position of the head relative to its dependents, and to what extent is it accomplished by choosing an optimal ordering of the dependents? To address this question, we compare real dependency lengths to random and optimal baselines where the position of the head and the direction of each dependent with respect to the head is fixed at the observed values. For example, given an observed head H with left dependents A, B, C, and right dependents D, E, F, we consider random orderings such as $[C, A, B, H, E, F, D]$, $[A, C, B, H, D, F, E]$, etc., where A, B, C and D, E, F are shuffled but maintain their direction with respect to the head.

Figure 3-9 shows real dependency lengths compared to the random and optimal fixed-head-position baselines. We find that all languages have dependency lengths shorter than

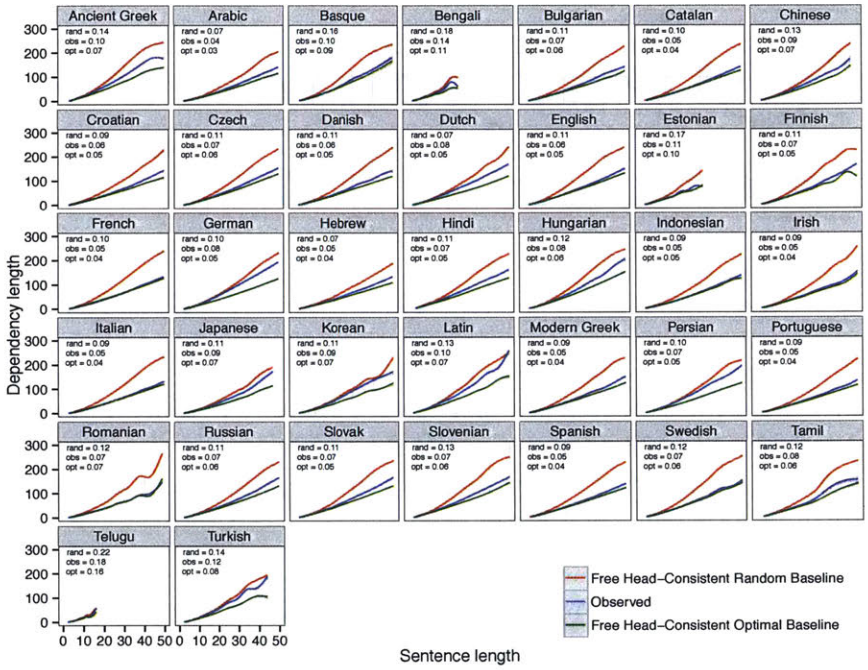


Figure 3-8: Real dependency lengths as a function of sentence length (blue), compared to the **Consistent Head Direction Free Word Order** Random baseline (red), and the **Consistent Head Direction Free Word Order** Optimal baseline (green). GAM fits are shown. `rand`, `obs`, and `opt` are the slopes for random, observed, and optimal dependency length as a function of squared sentence length, as in Figure 3 in the main text.

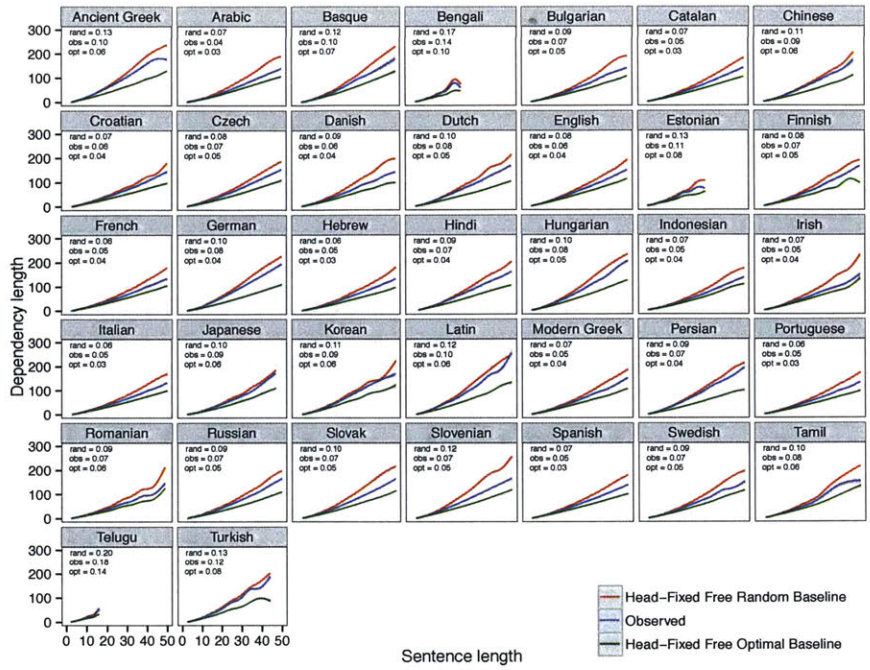


Figure 3-9: Real dependency lengths as a function of sentence length (blue), compared to the **Head-Fixed Free Word Order** Random baseline (red) and the **Head-Fixed Free Word Order** Optimal baseline (green). GAM fits are shown. `rand`, `obs`, and `opt` are the slopes for random, observed, and optimal dependency length as a function of squared sentence length, as in Figure 3 in the main text.

this baseline. The difference between real and random slopes is significant at $p < 0.001$ for all languages. The finding suggests that given a fixed head position, the ordering of dependents of the head is optimized across all languages, i.e. there is long-before-short order before heads and short-before-long order after heads.

3.2.5 Discussion

While there has previously been convincing behavioral and computational evidence for the avoidance of long dependencies, the evidence presented here is the strongest large-scale cross-linguistic support for the dependency length minimization as a universal phenomenon, across languages and language families.

Figure 3-5 also reveals that, while observed dependency lengths are always shorter than the random baselines, they are also longer than the minimal baselines (though some

languages such as Indonesian come quite close). In part, this is due to the unrealistic nature of the optimal baseline. In particular, that baseline does not have any consistency in word order[3].

In general, we believe dependency length should not be fully minimized because of other factors and desiderata influencing languages which may conflict with DLM. For example, linearizations should allow the underlying dependency structure to be recovered incrementally, in order to allow incremental understanding of utterances. In a sequence of two words A and B, when the comprehender receives B, it would be desirable to be able to determine immediately and correctly whether A is the head of B, B is the head of A, or A and B are both dependents of some as-yet-unheard word. If the order of dependents around a head is determined only by minimizing dependency length, then there is no guarantee that word orders will facilitate correct incremental inference. More generally, it has been argued that linearizations should allow the comprehender to quickly identify the syntactic and semantic properties of each word (see Hawkins (2014) for detailed discussion of the interaction of this principle with DLM). The interactions of DLM with these and other desiderata for languages are the subject of ongoing research.

The results presented here also show great variance in the effect size of DLM across languages. For example, the head-final languages such as Japanese, Korean, and Turkish show much less minimization than more head-initial languages such as Italian, Indonesian, and Irish, which are apparently highly optimized. In concordance with previous work, we find that German is among the languages with the longest dependency length. This variance is discussed more thoroughly in Section 3.4.

This work has shown that the preference for short dependencies is a widespread phenomenon that not confined to the limited languages and constructions previously studied. Therefore, it lends support to DLM-based explanations for language universals. Inasmuch as DLM can be attributed to minimizing the effort involved in language production and comprehension, this work joins previous work showing how aspects of natural language can be explained by considerations of efficiency (Zipf, 1949; Jaeger, 2006; Piantadosi et al.,

[3]See Gildea and Temperley (2010) for attempts to develop approximately optimal baselines which address this issue.

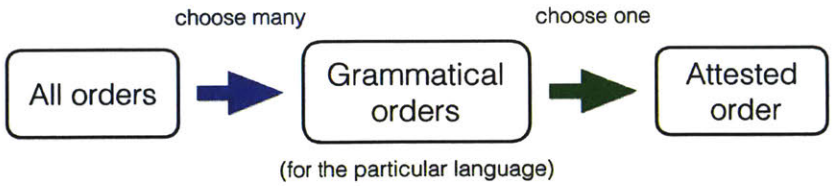


Figure 3-10: Schematic for how grammar and usage relate to linearizations. Grammar selects a set of licit linearizations from the logically possible ones; usage selects one linearization from the grammatically licit ones.

2011; Fedzechkina et al., 2012; Kemp and Regier, 2012; Hawkins, 2014).

3.3 Grammar and usage

The results above showed that dependency length in corpora is robustly shorter than we would expect from independently motivated general constraints alone. However, the observed minimization could be accomplished by two different mechanisms: it could be that grammars are optimized so that (when expressing common meanings) dependency length is minimal, or it could be that language users simply have a production preference for short dependencies (Rajkumar et al., 2016), without DLM affecting grammars per say.

Figure 3-10 shows how both grammar and usage can result in an observed DLM preference in word order. Among all the logically possible word orders for a tree with a particular meaning, the grammar of a language selects a set (or a probability distribution) of permitted orders. Then from that set, the language user selects one order to use. These two selections are two places where DLM can have an effect. For example, if the grammar only permits harmonic word orders, then the average utterance will come out with lower dependency length when compared with a grammar that enforces antiharmonic word orders (different branching directions for all dependency types). On the usage side, the grammar may permit either harmonic or antiharmonic orders, and the language user chooses the harmonic ones.

In terms of causal attribution, grammar and usage cannot be separated with certainty. Even if we observe complete consistency in word order per dependency tree, this consistency could logically be attributed to usage preferences. Nevertheless, it is possible to try to

estimate what grammar looks like from a corpus—the space of how trees can be linearized in general—and determine how that relates to the observed linearization of any particular tree.

The goal of this section is to develop probabilistic models of word order in grammar and use them to argue that DLM affects both grammar and usage preferences. The logic is as follows. We will take observed dependency trees and compare their dependency length to random reorderings *according to the probabilistic model of the grammar.* This work is essentially an attempt to automate the approach of Rajkumar et al. (2016), who compare dependency length in real utterances to dependency length in alternative grammatical utterances generated by hand. If the observed sentences have shorter dependency length than random grammatically possible reorerings, then this is evidence that language users are choosing particular utterances to minimize dependency length. Also, if the *distribution of grammatical reorderings* has lower dependency length than the random baselines from Section 3.2, then that is evidence that the grammar itself is affected by DLM.

In what follows, I will first describe the method for developing probabilistic models of word order conditional on dependency trees (Section 3.3.1). The methods and issues here closely parallel those discussed in Chapter 2.[4] Then I show results of comparing real dependency length to these random grammatical reorderings (Section 3.3.2).

3.3.1 Generative models for dependency tree linearization

We explore generative models for producing linearizations of unordered labeled syntactic dependency trees. This specific task has attracted attention in recent years (Filippova and Strube, 2009; He et al., 2009; Belz et al., 2011; Bohnet et al., 2012; Zhang, 2013) because it forms a useful part of a natural language generation pipeline, especially in machine translation (Chang and Toutanova, 2007) and summarization (Barzilay and McKeown, 2005). Closely related tasks are generation of sentences given CCG parses (White and Rajkumar, 2012), bags of words (Liu et al., 2015), and semantic graphs (Braune et al., 2014).

[4]We use these dependency models because they allow us to closely control the information that goes into linearization: off-the-shelf systems often include many opaque features and may built in DLM as a preference covertly.

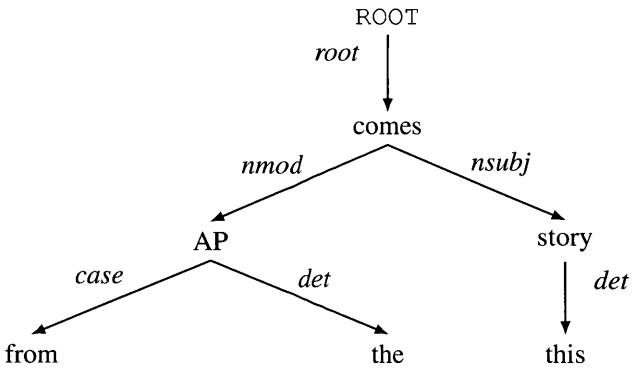
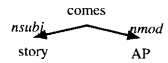


Figure 3-11: Example unordered dependency tree. Possible linearizations include (1) *This story comes from the AP* and (2) *From the AP comes this story*. Order 2 is the original order in the corpus, but order 1 is much more likely under our models.

Here we focus narrowly on testing probabilistic generative models for dependency tree linearization. In contrast, the approach in most previous work is to apply a variety of scoring functions to trees and linearizations and search for an optimally-scoring tree among some set. The probabilistic linearization models we investigate are derived from generative models for dependency trees (Eisner, 1996), as most commonly used in unsupervised grammar induction (Klein and Manning, 2004; Gelling et al., 2012). Generative dependency models have typically been evaluated in a parsing task (Eisner, 1997). Here, we are interested in the inverse task: inferring a distribution over linear orders given unordered dependency trees.

This is the first work to consider generative dependency models from the perspective of word ordering. The results can potentially shed light on how ordering constraints are best represented in such models. In addition, the use of probabilistic models means that we can easily define well-motivated normalized probability distributions over orders of dependency trees. These distributions are useful for answering scientific questions about crosslinguistic word order in quantitative linguistics, where obtaining robust estimates has proven challenging due to data sparsity (Futrell et al., 2015c).

We investigate head-outward projective generative dependency models. In these models, an ordered dependency tree is generated by the following kind of procedure. Given a head node, we use some generative process G to generate a depth-1 subtree rooted in that

head node. Then we apply the procedure recursively to each of the dependent nodes. By applying the procedure starting at a ROOT node, we generate a dependency tree. For example, to generate the dependency tree in Figure 3-11 from the node *comes* down, we take the head *comes* and generate the subtree 

, then we take the head *story* and generate 

, and so on. In this work, we experiment with different specific generative processes G which generate a local subtree conditioned on a head.

Model types

Here we describe some possible generative processes G which generate subtrees conditioned on a head. These models contain progressively more information about ordering relations among sister dependents.

A common starting point for G is **Eisner Model C** (Eisner, 1996). In this model, dependents on one side of the head are generated by repeatedly sampling from a categorical distribution until a special stop-symbol is generated. The model only captures the propensity of dependents to appear on the left or right of the head, and does not capture any order constraints between sister dependents on one side of the head.

We consider a generalization of Eisner Model C which we call **Dependent N-gram** models. In a Dependent N-gram model, we generate dependents on each side the head by sampling a *sequence* of dependents from an N-gram model. Each dependent is generated conditional on the $N-1$ previously generated dependents from the head outwards. We have two separate N-gram sequence distributions for left and right dependents. Eisner Model C can be seen as a Dependent N-gram model with $N = 1$.

We also consider a model which can capture many more ordering relations among sister dependents: given a head h, sample a subtree whose head is h from a Categorical distribution over subtrees. We call this the **Observed Orders** model because in practice we are simply sampling one of the observed orders from the training data. This generative process has the capacity to capture the most ordering relations between sister dependents.

Distributions over permutations of dependents

We have discussed generative models for ordered dependency trees. Here we discuss how to use them to make generative models for word orders conditional on unordered dependency trees.

Suppose we have a generative process G for dependency trees which takes a head h and generates a sequence of dependents $\mathbf{w}_l$ to the left of h and a sequence of dependents $\mathbf{w}_r$ to the right of h. Let $\mathbf{w}$ denote the pair $(\mathbf{w}_l, \mathbf{w}_r)$, which we call the **configuration** of dependents. To get the probability of some $\mathbf{w}$ given an unordered subtree u, we want to calculate the probability of $\mathbf{w}$ given that G has generated the particular multiset $\mathbf{W}$ of dependents corresponding to u. To do this, we calculate:

$$\begin{aligned} p(\mathbf{w}|\mathbf{W}) &= \frac{p(\mathbf{w}, \mathbf{W})}{p(\mathbf{W})} \\ &= \frac{p(\mathbf{w})}{Z}, \end{aligned} \tag{3.2}$$

where

$$Z = \sum_{\mathbf{w}' \in \mathcal{W}} p(\mathbf{w}') \tag{3.3}$$

and $\mathcal{W}$ is the set of all possible configurations $(\mathbf{w}_l, \mathbf{w}_r)$ compatible with multiset $\mathbf{W}$. That is, $\mathcal{W}$ is the set of pairs of permutations of multisets $\mathbf{W}_l$ and $\mathbf{W}_r$ for all possible partitions of $\mathbf{W}$ into $\mathbf{W}_l$ and $\mathbf{W}_r$. The generative dependency model gives us the probability $p(\mathbf{w})$.

It remains to calculate the normalizing constant Z, the sum of probabilities of possible configurations. For the Observed Orders model, Z is the sum of probabilities of subtrees with the same dependents as subtree u. For the Dependent N-gram models of order N, we calculate Z using a dynamic programming algorithm, presented in Algorithm 1 as memoized recursive functions. When $N = 1$ (Eisner Model C), Z is more simply:

$$\begin{aligned} Z_{\text{emc}} = p_L(\texttt{stop}) p_R(\texttt{stop}) \sum_{(\mathbf{W}_l, \mathbf{W}_r) \in \textsc{parts}(\mathbf{W})} |\mathbf{W}_l|! |\mathbf{W}_r|! \\ \prod_{w \in \mathbf{W}_l} p_L(w) \prod_{w \in \mathbf{W}_r} p_R(w), \end{aligned}$$

where PARTS($\mathbf{W}$) is the set of all partitions of multiset $\mathbf{W}$ into two multisets $\mathbf{W}_l$ and $\mathbf{W}_r$, p_L is the probability mass function for a dependent to the left of the head, p_R is the function for a dependent to the right, and `stop` is a special symbol in the support of p_L and p_R which indicates that generation of dependents should halt. The probability mass functions may be conditional on the head h. These methods for calculating Z make it possible to transform a generative dependency model into a model of dependency tree *ordering* conditional on local subtree structure.

Algorithm 1 Compute the sum of probabilities of all configurations of dependents $\mathbf{W}$ under a Dependent N-gram model with two component N-gram models of order N: p_R for sequences to the right of the head and p_L for sequences to the left.

memoized function RIGHT_NORM($\mathbf{r}$, $\mathbf{c}$)
  if $|\mathbf{r}| = 0$ **then**
    return $p_R(\texttt{stop} \mid \mathbf{c})$
  end if
  $Z \leftarrow 0$
  for $i = 1 : |\mathbf{r}|$ **do**
    $\mathbf{r}' \leftarrow$ elements of $\mathbf{r}$ except the ith
    $\mathbf{c}' \leftarrow$ append r_i to $\mathbf{c}$ then truncate to length $N - 1$
    $Z \leftarrow Z + p_R(r_i|\mathbf{c}) \times$ RIGHT_NORM($\mathbf{r}', \mathbf{c}'$)
  end for
  return Z
end memoized function
memoized function LEFT_NORM($\mathbf{r}$, $\mathbf{c}$)
  $Z \leftarrow p_L(\texttt{stop} \mid \mathbf{c}) \times$ RIGHT_NORM($[\texttt{start}], \mathbf{r}$)
  for $i = 1 : |\mathbf{r}|$ **do**
    $\mathbf{r}' \leftarrow$ elements of $\mathbf{r}$ except the ith
    $\mathbf{c}' \leftarrow$ append r_i to $\mathbf{c}$ then truncate to length $N - 1$
    $Z \leftarrow Z + p_L(r_i|\mathbf{c}) \times$ LEFT_NORM($\mathbf{r}', \mathbf{c}'$)
  end for
  return Z
end memoized function
Result is LEFT_NORM($\mathbf{W}$, $[\texttt{start}]$)

Labelling

The previous section discussed the question of the structure of the generative process for dependency trees. Here we discuss an orthogonal modeling question, which we call **labelling**: what information about the *labels* on dependency tree nodes and edges should be included in our models. Dependency tree nodes are labeled with wordforms, lemmas, and parts-of-

speech (POS) tags; and dependency tree edges are labeled with relation types. A model might generate orders of dependents conditioned on all of these labels, or a subset of them. Decisions can be conditioned on the head of a phrase; when so, they can be conditioned on the wordform, POS, etc. For example, a generative dependnecy model might generate (relation type, dependent POS tag) tuples conditioned on the POS tag of the head of the phrase. When we use such a model for dependency linearization, we would say the model's labelling is relation type, dependent POS, and head POS. In this study, we avoid including wordforms or lemmas in the labelling, to avoid data sparsity issues.

Model estimation and smoothing

In order to alleviate data sparsity in fitting our models, we adopt two smoothing methods from the language modelling literature.

All categorical distributions are estimated using add-k smoothing where $k = 0.01$. For the Dependent N-gram models, this means adding k pseudocounts for each possible dependent in each context. For the Observed Orders model, this means adding k pseudocounts for each possible permutation of the head and its dependents.

We also experiment with combining our models into mixture distributions. This can be viewed as a kind of back-off smoothing (Katz, 1987), where the Observed Orders model is the model with the most context, and Dependent N-grams and Eisner Model C are backoff distributions with successively less context. Similarly, models with less information in the labelling can serve as backoff distributions for models with more information in the labelling. For example, a model which is conditioned on the POS of the head can be backed off to a model which does not condition on the head at all. We find optimal mixture weights using the Baum-Welch algorithm tuned on a held-out development set.

Evaluation

Here we empirically evaluate some options for model type and model labelling as described above. We are interested in how many of the possible orders of a sentence our model can generate (recall), and in how many of our generated orders really are acceptable (precision). As a recall-like measure, we quantify the probability of the word orders of held-out

Labelling	Model	Basque	Czech	English	Finnish	French	German	Hebrew	Indonesian	Persian	Spanish	Swedish
HDR	oo	-6.83	-7.58	-5.23	-7.35	-10.86	-8.36	-9.74	-8.99	-10.39	-11.31	-8.83
	n1	-6.12	-8.97	-5.08	-7.15	-11.54	-9.81	-9.63	-8.68	-10.63	-13.19	-8.37
	n2	-4.86	-6.35	-2.87	-5.30	-6.86	-6.60	-5.91	-5.98	-5.54	-7.47	-4.92
	n3	-5.92	-6.59	-3.13	-5.68	-7.34	-7.02	-6.81	-6.69	-6.49	-8.06	-5.68
	n123	-4.58	-6.18	-2.60	-5.11	-6.67	-6.19	-5.77	-5.73	-5.51	-7.36	-4.72
	oo+n123	-4.52	-5.95	-2.57	-5.04	-6.58	-5.92	-5.68	-5.68	-5.47	-7.27	-4.68
HDR+R	oo	-5.56	-6.78	-3.94	-6.25	-9.63	-7.42	-7.95	-7.51	-9.19	-9.54	-7.28
	n1	-6.08	-8.97	-5.07	-7.16	-11.54	-9.79	-9.58	-8.67	-10.62	-13.17	-8.35
	n2	-4.49	-6.31	-2.62	-5.17	-6.79	-6.34	-5.62	-5.67	-5.42	-7.40	-4.67
	n3	-4.86	-6.41	-2.61	-5.20	-7.08	-6.43	-6.07	-6.02	-6.04	-7.70	-5.02
	n123	-4.41	-6.15	-2.48	-5.01	-6.59	-5.99	-5.54	-5.53	-5.42	-7.29	-4.53
	oo+n123	**-4.29**	**-5.84**	**-2.44**	**-4.88**	**-6.50**	**-5.74**	**-5.40**	**-5.47**	**-5.38**	**-7.09**	**-4.46**

Table 3.1: Average log likelihood of word order per sentence in test set under various models. Under "Labelling", **HDR** means conditioning on Head POS, Dependent POS, and Relation Type, and **R** means conditioning on Relation Type alone (see Section 3.3.1). Under "Model", **oo** is the Observed Orders model, **n1** is the Dependent 1-gram model (Eisner Model C), **n2** is the Dependent 2-gram model, and **n3** is the Dependent 3-gram model (see Section 3.3.1). In both columns, x+y means a mixture of model x and model y; **n123** means n1+n2+n3.

test sentences. Low probabilities assigned to held-out sentences indicate that there are possible orders which our model is missing. As a precision-like measure, we get human acceptability ratings for sentence reorderings generated by our model.

We carry out our evaluations using the dependency corpora of the Universal Dependencies project (v1.1) (Agić et al., 2015), with the train/dev/test splits provided in that dataset. We remove nodes and edges dealing with punctuation. Due to space constraints, we only present results from 11 languages here.

Test-Set Probability Here we calculate average probabilities of word orders per sentence in the test set. This number can be interpreted as the (negative) average amount of information contained in the word order of a sentence beyond information about dependency relations.

The results for selected languages are shown in Table 3.1. The biggest gains come from using Dependent N-gram models with $N > 1$, and from backing off the model labelling. The Observed Orders model does poorly on its own, likely due to data sparsity; its performance is much improved when backing off from conditioning on the head. Eisner Model C (n1) also performs poorly, likely because it cannot represent any ordering constraints among sister dependents. The fact it helps to back off to distributions not conditioned on the head suggests that there are commonalities among distributions of dependents of different heads, which could be exploited in further generative dependency models.

Human Evaluation We collected human ratings for sentence reorderings sampled from the English models from 54 native American English speakers on Amazon Mechanical Turk. We randomly selected a set of 90 sentences from the test set of the English Universal Dependencies corpus. We generated a reordering of each sentence according to each of 12 model configurations in Table 3.1. Each participant saw an original sentence and a reordering of it, and was asked to rate how natural each version of the sentence sounded, on a scale of 1 to 5. The order of presentation of the original and reordered forms was randomized, so that participants were not aware of which form was the original and which was a reordering. Each participant rated 56 sentence pairs. Participants were also asked

Labelling	Model	Acceptability	Same Meaning
HDR	oo	2.92	0.58
	n1	2.06	0.44
	n2	3.42	0.78
	n3	3.48	**0.85**
	n123	**3.56**	0.79
	oo+n123	3.45	0.75
HDR+R	oo	3.11	0.72
	n1	2.11	0.49
	n2	3.32	0.80
	n3	3.52	0.77
	n123	3.31	0.76
	oo+n123	3.43	0.80

Table 3.2: Mean acceptability rating out of 5, and proportion of reordered sentences with the same meaning as the original, for English models. Labels as in Table 3.1.

whether the two sentences in a pair meant the same thing, with "can't tell" as a possible answer. This part of the evaluation also helps answer the scientific question of how well dependency trees with relation types encode a meaning invariant to word order.

Table 3.2 shows average human acceptability ratings for reorderings, and the proportion of sentence pairs judged to mean the same thing. The original sentences have an average acceptability rating of 4.48/5. The very best performing models are those which do not back off to a distribution not conditioned on the head. However, in the case of the Observed Orders and other sparse models, we see consistent improvement from this backoff.

Figure 3-12 shows the acceptability ratings (out of 5) plotted against test set probability. We see that the models which yield poor test set probability also have poor acceptability ratings.

Comparison with other systems Previous work has focused on the ability to correctly reconstruct the word order of an observed dependency tree. Our goal is to explicitly model a distribution over possible orders, rather than to recover a single correct order, because many orders are often possible, and the particulator order that a dependency tree originally appeared in might not be the most natural. For example, our models typically reorder the sentence "From the AP comes this story" (in Figure 3-11) as "This story comes from

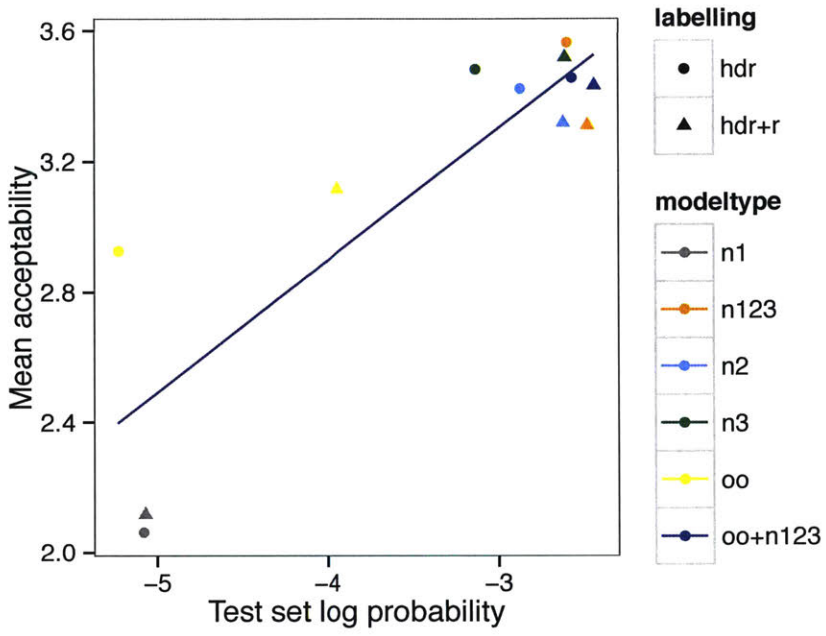


Figure 3-12: Comparison of test set probability (Table 3.1) and acceptability ratings (Table 3.2) for English across models. A least-squares linear regression line is shown. Labels as in Table 3.1.

the AP"; the second order is arguably more natural, though the first is idiomatic for this particular phrase. So we do not believe that BLEU scores and other metrics of similarity to a "correct" ordering are particularly relevant for our task.

Previous work uses BLEU scores (Papineni et al., 2002) and human ratings to evaluate generation of word orders. To provide some comparability with previous work, we report BLEU scores on the 2011 Shared Task data here. The systems reported in Belz et al. (2011) achieve BLEU scores ranging from 23 to 89 for English; subsequent work achieves BLEU scores of 91.6 on the same data (Bohnet et al., 2012). Drawing the highest-probability orderings from our models, we achieve a top BLEU score of 57.7 using the model configuration hdr/oo. Curiously, hdr/oo is typically the worst model configuration in the test set probability evaluation (Section 3.3.1). The BLEU performance is in the middle range of the Shared Task systems. The human evaluation of our models is more optimistic: the best score for Meaning Similarity in the Shared Task was 84/100 (Bohnet et al., 2011), while sentences ordered according to our models were judged to have the same meaning as

the original in 85% of cases (Table 3.2), though these figures are based on different data. These comparisons suggest that these generative models do not provide state-of-the-art performance, but do capture some of the same information as previous models.

Discussion Overall, the most effective models are the Dependent N-gram models. The naive approach to modeling order relations among sister dependents, as embodied in the Observed Orders model, does not generalize well. The result suggests that models like the Dependent N-gram model might be effective as generative dependency models.

3.3.2 Dependency length in grammatical baselines

Here I will compare dependency length with three random baselines generated using the methods described above. The first is the random grammatical baseline that simply orders the immediate dependents of a head uniformly at random from among the observed orders, as defined only by dependency relation types (the **licit** baseline). This baseline is included for those who do not believe in probabilistic grammar; it also has the largest variance of any of the baselines. The second is the random baseline that scored highest on the "same meaning" evaluation for English (the **same meaning** baseline). The third is the random baseline that scored highest on perplexity for all languages (the **best perplexity** baseline). I will compare these random reorderings to the real observed dependency length of sentences, and to the **free projective** baseline from Section 3.2.

Linguistic interpretation of baselines

Before launching into the results, some discussion is in order on the specific linguistic interpretation of the random baselines.

The main constraint in the reordering models of Section 3.3.1 (and the word order freedom models in Chapter 2) was that order is only computed related to the *immediate dependents of a head.* This is done in order to alleviate data sparsity in model estimation, but it puts limits on what ordering constraints can be represented by the model. In particular, it means that ordering constraints that involve heads and their grandchildren, or any other relationship going beyond direct head-dependent relationships and sibling relationships, are

not represented. We also assume linearizations are projective.

As such, the conservative interpretation of the results in this section is that we find dependency length minimization beyond what would be expected *from only (1) projectivity and (2) the ordering constraints among heads and immediate dependents.* The constraint that order can only be constrained among immediate dependents of a head, and that word order is projective, corresponds to the assumption that language follows a context-free grammar. The results show conservatively that whatever constraints exist beyond what can be expressed in a context-free formalism, they serve to lower dependency length beyond what would be expected from projective dependency-local constraints alone. Nevertheless, because we believe a majority of word order constraints can be represented in a context-free framework, we will interpret the observed minimization of dependency length beyond the grammatical baselines as (noisy) evidence for DLM in usage.

Another issue that arises is that the estimates of grammatical reorderings can themselves be affected by usage preferences. If people have strong preferences for DLM in usage, this will be reflected in all their utterances, and our estimates of grammatical orders are calcualted from those utterances. Perhaps some orders are grammatically possible, but people never say them because of their bad dependency length properties; we would miss these orders in our baselines.

Our grammatical baselines really reflect the expected behavior at the level of the phrase. We believe this provides a useful estimate of grammatical constraints, but for those who do not take this as indicative of grammatical constraints, our results still show that people minimize dependency length in usage beyond what would be expected based on their behavior at the level of single phrases.

Results

Figure 3-13 shows real dependency length compared to the random baselines mentioned. Because the lines are all very close, Figure 3-14 shows the same figure zoomed in to sentence lengths between 15 and 30, so that the relative ordering of the different baselines is clear.

We see that the various grammatical baselines all produce linearizations with very sim-

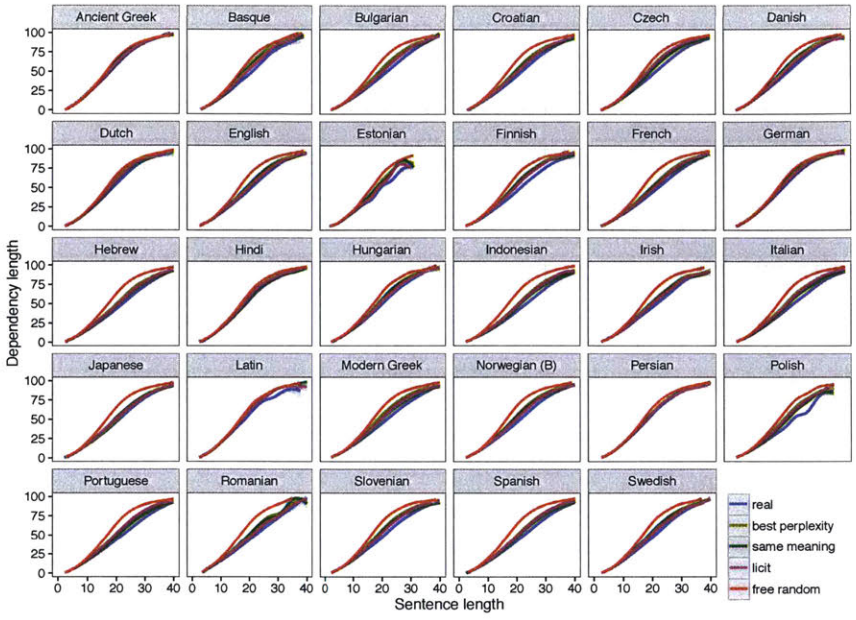


Figure 3-13: Dependency length as a function of sentence length, as estimated using cubic splies as in Section 3.2.1.

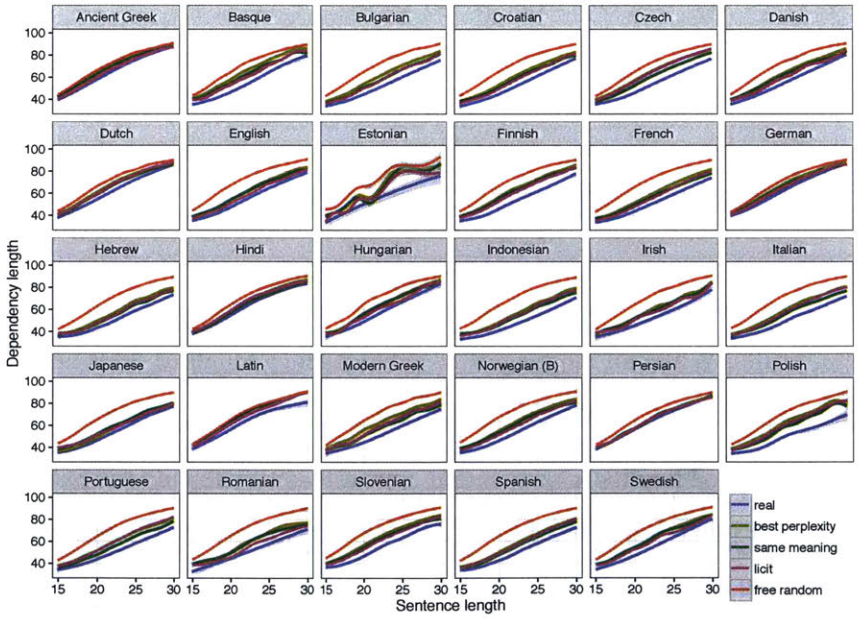


Figure 3-14: Dependency length as a function of sentence length for sentence of length 15 to 30, as estimated using cubic splies as in Section 3.2.1.

ilar dependency length. I will not attempt to draw any contrast among these baselines.

Comparing the projective baseline, the random baselines, and the observed dependency length, we see that the projective baseline has the longest dependency length, followed by the random baselines, followed by the observed dependency length. We analyzed the results statistically using the same regression methods described in Section 3.2.1. For all languages, the dependency length growth rate for all the baselines is greater than for the observed sentences at $p < 0.001$. Also, for all languages, the linearizations according to the licit baseline have lower dependency length growth rate than linearizations according to the projective baseline at $p < 0.001$ for all languages, suggesting that grammatical restrictions reduce dependency length.

The results show that grammatical orders are shorter than fully random orders, and that observed orders are shorter than grammatical orders. Thus, as a broad interpretation, we have evidence that both grammar and usage are affected by DLM. The most narrow interpretation is that people's expected ordering behavior at the level of the phrase minimizes dependency length, and that their behavior beyond what is described at the level of the phrase serves to further minimize dependency length.

3.4 Variation in dependency length

In this section, I discuss some of the variation between languages observed in all the dependency length results. While all languages have dependency length shorter than the random baselines presented, they show variance in the extent to which this is true.

Here I discuss some linguistic factors that appear to correlate with the extent of DLM. I show that languages thare more head-final have longer dependencies; languages with more word order freedom have longer dependencies; and languages with more complex morphology have more word order freedom. These results are not expected from the motivations for DLM described in Section 3.1. They were not expected, and represent explananda for any future theory of quantitative syntax.

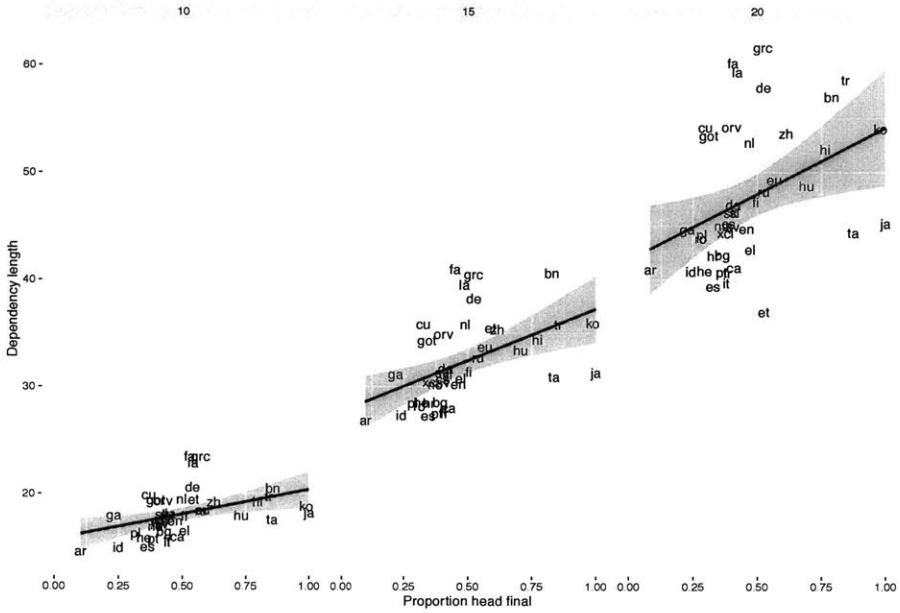


Figure 3-15: Dependency length compared to proportion of head-final dependencies for sentences of length 10, 15, 20.

N	Pearson's r	Spearman's ρ
10	.45**	.60***
15	.51***	.64***
20	.49***	.53***

Table 3.3: Pearson and Spearman correlation coefficients across languages of mean dependency length with proportion of head-final dependencies, for sentences of length N. * = significant at $p < .05$, ** = significant at $p < .01$, *** = significant at $p < .001$.

3.4.1 Head-finality

We find that languages with a larger proportion of head final dependencies tend to have longer dependencies. Figure 3-15 shows observed average dependency length at sentence lengths 10, 15, and 20 compared with the proportion of head-final dependencies at that sentence length. For example, we see that Japanese and Korean are nearly entirely head-final, while Arabic is largely head-initial. The more head-final languages have significantly longer dependencies; the correlations between dependency length and proportion of head-final dependencies are shown in Table 3.3.

There also appears to be evidence for stronger DLM in head-initial contexts than head-

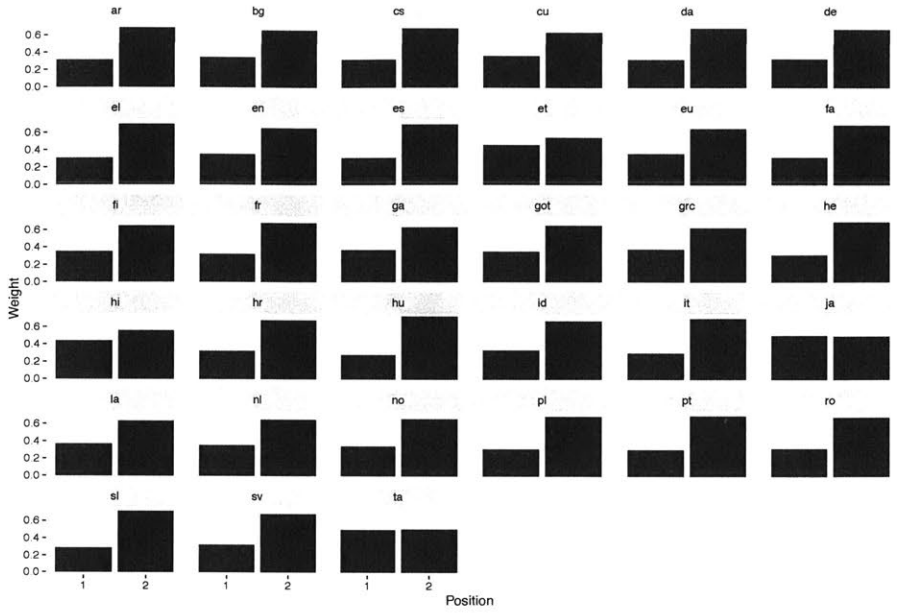


Figure 3-16: Constituent weight for two dependents to the right of a head, for heads with exactly two right dependents, as a proportion of total constituent weight.

final contexts within languages. In head-initial contexts, DLM is accomplished by ordering dependent constituents from short to long. The short-before-long preference is demonstrated in Figure 3-16, which shows the average weight (number of words) in the two phrases to the right of a head, for heads with exactly two dependents to the right, normalized by the total number of words in each context. It shows a clear preference for short constituents before long constituents. Figure 3-17 shows the same result for three constituents after the head.

Turning to head-final contexts, we should expect a long-before-short preference. Figure 3-18 shows the average weight of two dependents *before* a head. Figure 3-19 shows the result for three dependents. Here again, we often (but not universally) see a long-before-short preference. But the magnitude of the preference is much weaker than the short-before-long preference after heads.

One possible explanation for the apparent tolerance of longer dependencies before heads than after heads could be an interaction of DLM preferences with given-before-new word ordering preferences (Halliday, 1967; Birner and Ward, 2006). Given-before-new

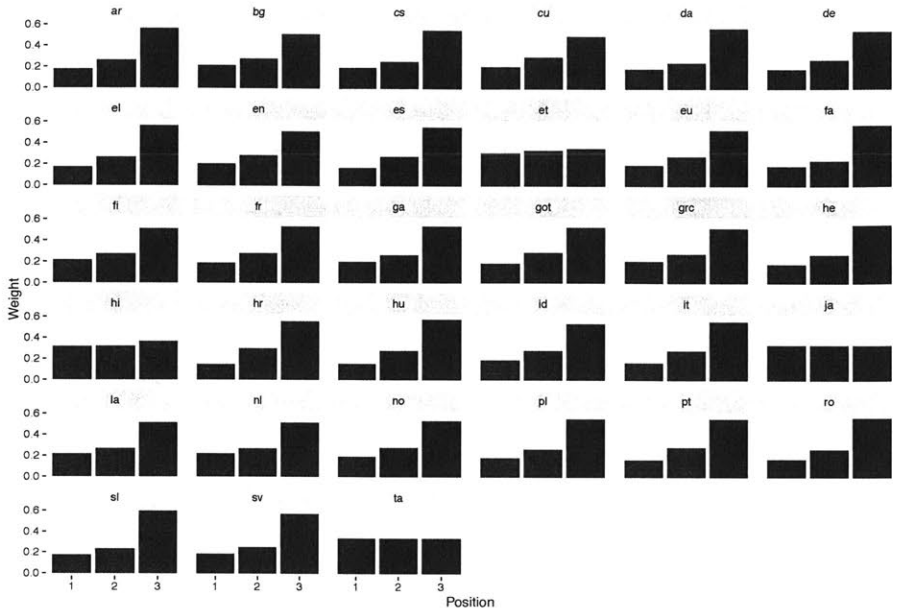


Figure 3-17: Constituent weight for three dependents to the right of a head, for heads with exactly three right dependents, as a proportion of total constituent weight.

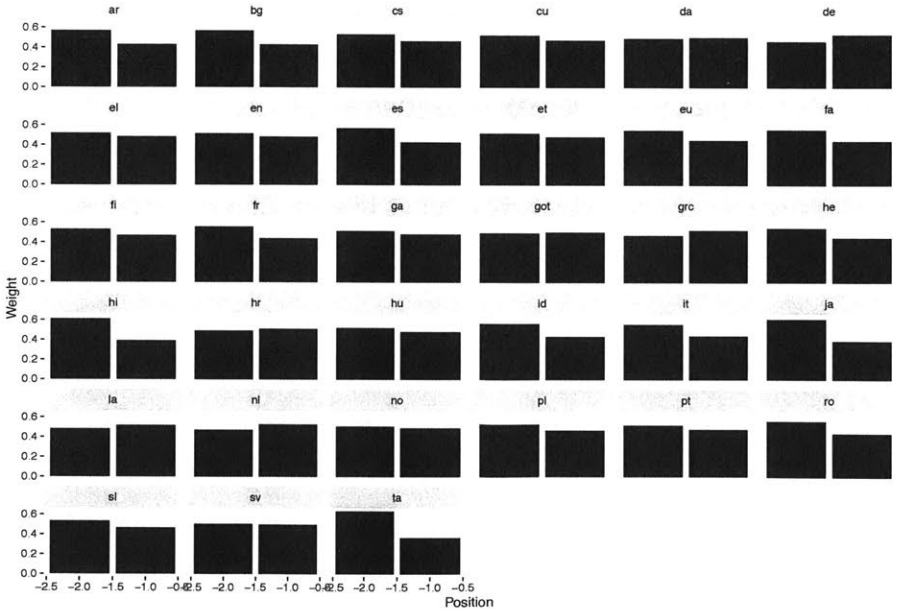


Figure 3-18: Constituent weight for two dependents to the left of a head, for heads with exactly two left dependents, as a proportion of total constituent weight.

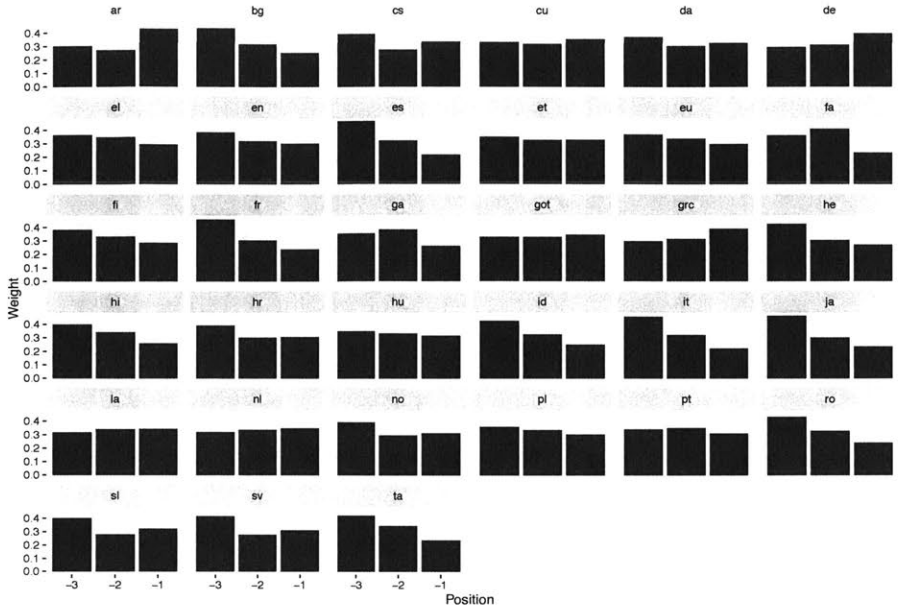


Figure 3-19: Constituent weight for three dependents to the left of a head, for heads with exactly three left dependents, as a proportion of total constituent weight.

preferences force words with more concrete, known referents, such as nouns, earlier in a sentence, thus favoring SOV word order (Gibson et al., 2013). There is also evidence that when a dependent is given, dependency locality is less important for predicting its placement (Xu and Liu, 2015). Other explanations in terms of incremental parsing may be possible.

Another explanation could have to do with morphology. Head final languages typically have richer morphology than head initial languages (Dryer, 2002). Morphology (case and/or agreement) provides informative cues about what the head of each marked word is. If we think that dependency locality effect are in part driven by inaccuracy in parsing (as argued in Vasishth et al. (2017)), then such morphology would alleviate dependency locality effects. Indeed, Ros et al. (2015) find weaker DLM preferences in morphologically rich languages. Thus head-final languages may get away with having longer dependencies than head-initial languages.

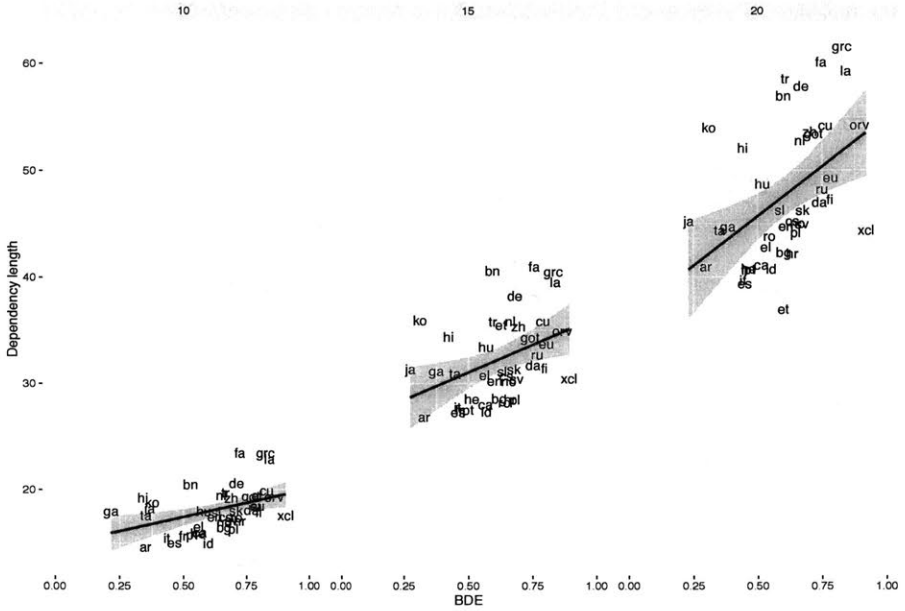


Figure 3-20: Dependency length compared to Branching Direction Entropy conditional on parts of speech (see Section 2.3.3) for sentences of length 10, 15, 20.

N	Pearson's r	Spearman's ρ
10	.38*	.41**
15	.33*	.42**
20	.64***	.58***

Table 3.4: Pearson and Spearman correlation coefficients across languages of mean dependency length with Branching Direction Entropy, for sentences of length N. * = significant at $p < .05$, ** = significant at $p < .01$, *** = significant at $p < .001$.

3.4.2 Word order freedom

We find that languages with more word order freedom have longer dependency length. Figure 3-20 shows the correlation of mean dependency length and the "Branching Direction Entropy" measure from Section 2.3.3, conditional on parts of speech. Correlations between branching direction entropy and dependency length are shown in Table 3.4.

A similar result was found in Gulordava and Merlo (2015a) for Latin and Ancient Greek. In historical dependency corpora of those languages, increasing word order rigidness over the centuries is associated with decreasing dependency length. Tily (2010) also finds a decreasing dependency length over time in English, which coincides with a loss of

word order freedom.

At first glance, this result is disturbing from the perspective of a claimed universal preference for DLM. If languages have more word order freedom, then it seems they should use that freedom in order to make their dependencies even shorter, rather than using it to make them longer. However, when we consider that languages with more free word order also often have informative morphology, we see a possible motivation for this result. As discussed above for head-final languages, there are theoretical and empirical reasons to believe that morphological richness should correlate with less pressure for DLM.

3.4.3 Morphological richness

Here we directly test the idea that more morphologically rich languages have less pressure for DLM and thus longer dependencies. We measure morphological richness using an information-theoretic measure C which gives the information content of the distributino over wordforms W beyond the information content of the distribution over their lemmas L:

$$\begin{aligned} C &= H(W|L) \\ &= H(W) - I(W;L) \\ &= H(W) - H(L). \end{aligned}$$

This measure is closely related with other proposed measures of morphological complexity (see Bentz et al. (2016) for a review). For example, the measure of "normalized frequency difference" (NFD) (Bentz et al., 2017) can be seen as a nonparametric estimate of C.

The information theoretic measure tells us directly how much information is present in morphology, but it is difficult to estimate for a number of reasons. The entropy estimation required is difficult, especially since we are operating words and lemmas, meaning we require large amounts of data for the estimates to converge (Bentz and Alikaniotis, 2016). It also requires lemma annotations, which introduces issues of how to define lemmas, and whether this might be done differently across corpora and languages.[5] As a result, although

[5]For example, lemmas in the UD corpus for Hindi are largely identical to wordforms, resulting in Hindi

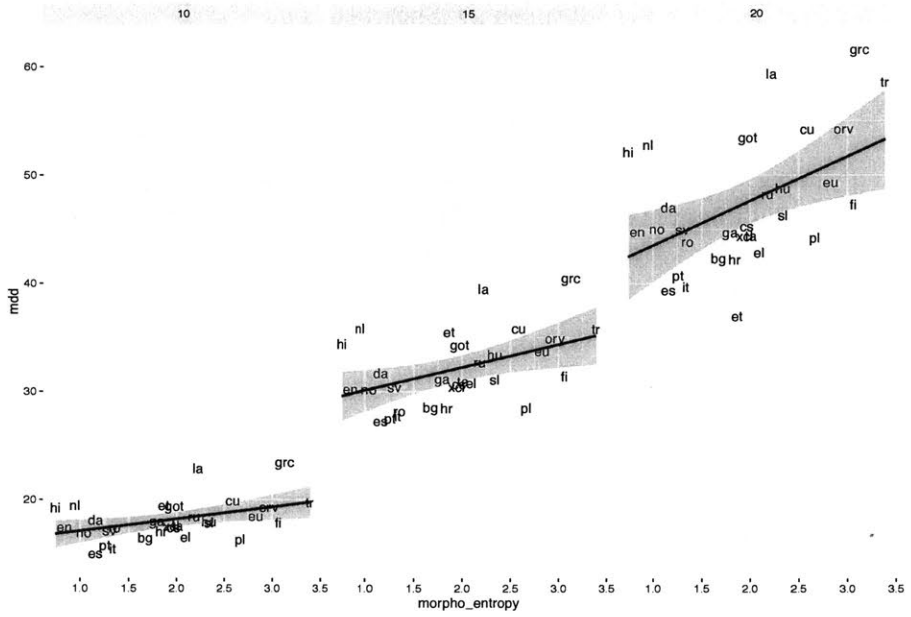


Figure 3-21: Dependency length compared to morphological information content (see text) for sentences of length 10, 15, 20.

this measure is theoretically well grounded, as an empirical measure it should be taken with a grain of salt.

Figure 3-21 shows the correlation of morphological information content with dependency length for sentence lengths 10, 15, and 20; statistics are given in Table 3.5. The entropies are calculated by the Pitman-Yor Mixture method of Archer et al. (2014).[6] As expected—though with the caveats above—we find that languages with more morphological information content have longer dependencies. This result supports the idea that morphology lessons DLM pressures by weakening dependency locality effects. The weakening of locality effects can happen if dependency locality is in part driven by inaccuracy in memory retrieval (Vasishth et al., 2017), while morphology provides cues that make memory retrieval more accurate.

having an incorrectly low estimate of morphological complexity.

[6]See Archer et al. (2013) for justification of the use of this estimator for what is essentially a mutual information estimation problem.

N	Pearson's r	Spearman's ρ
10	.43*	.42**
15	.45*	.47**
20	.49**	.47*

Table 3.5: Pearson and Spearman correlation coefficients across languages of mean dependency length with morphological information content, for sentences of length N. * = significant at $p < .05$, ** = significant at $p < .01$, *** = significant at $p < .001$.

3.5 Conclusion

I have presented evidence that DLM is a universal pressure affecting word order in both grammar and usage in corpora of over 40 languages, and shown a number of unexpected results bearing on the variance in dependency length among languages. In addition to making this scientific point, these results show the utility of the quantitative, information-theoretic approach to syntax, as many of the methods and measure from Chapter 2 proved useful here.

The approach and results here open the way for future work testing the quantitative dependency length properties of usage in more detail. For example, we may find that dependency length minimization is more operative within some constructions than others (Rajkumar et al., 2016), and seek explanations for this variance. The same approach of comparing real orders with random baselines can also be used to show other pressures affecting word orders (Gildea and Jaeger, 2015).

Chapter 4

Noisy-Context Surprisal as a Human Sentence Processing Cost Model

4.1 Introduction

Models of human sentence processing difficulty can be divided into two kinds, **expectation-based** and **memory-based**.[1] Expectation-based models predict the processing difficulty of a word from the word's surprisal given previous material in the sentence (Hale, 2001; Levy, 2008a). These models have good coverage: they can account for effects of syntactic construction frequency and resolution of ambiguity on incremental processing difficulty. Memory-based models, on the other hand, explain difficulty resulting from working memory limitations during incremental parsing (Gibson, 1998; Lewis and Vasishth, 2005); a major prediction of these models is **locality effects**, where processing a word is difficult when it is far from other words with which it must be syntactically integrated. Expectation-based models do not intrinsically capture this difficulty.

Integrating these two approaches at a high level has proven challenging. A major hurdle is that the theories are typically stated at different levels of analysis: expectation-based theories are computational-level theories (Marr, 1982) specifying what computational problem the human sentence processing system is solving—the problem of how update one's belief about a sentence given a new word—without specifying implementation details. Memory-based theories such as Lewis and Vasishth (2005) are for the most part based in mechanistic algorithmic-level theories describing the actions of a specific incremental parser.

Previous theories that capture both surprisal and locality effects have typically done so by augmenting parsing models with a special prediction-verification operation to capture surprisal effects (Demberg and Keller, 2009; Demberg et al., 2013), or by combining surprisal and memory-based cost derived from a parsing model as separate factors in a linear model (Shain et al., 2016). These models capture surprisal and locality effects at the same time, but they do not clearly capture phenomena involving the interaction of memory and probabilistic expectations such as language-dependent structural forgetting (see Section 4.3).

Here we develop a computational-level model capturing both memory and expectation effects from a single set of principles, without reference to a specific parsing algorithm. In

[1] Code for replicating the results in this section can be found online at `http://github.com/Futrell/nc-surprisal-eacl`.

our model, the processing cost of a word is a function of its surprisal given a *noisy* representation of previous context (Section 4.2). We show that the model can reproduce structural forgetting effects, including the difference between English and German (Section 4.3), a phenomenon not previously captured by memory-based or expectation-based models in isolation. We also give a derivation of the existence of locality effects in the model; these effects were previously accounted for only in mechanistic memory-based models (Section 4.4). The derivation yields a generalization of classic locality effects which we call **information locality**: sentences are predicted to be easier to process when words with high mutual information are close. We give corpus-based evidence that words in syntactic dependencies have high mutual information, meaning that classical dependency locality effects can be seen as a subset of information locality effects.

4.2 Noisy-Context Surprisal

In surprisal theory, the processing cost of a word is asserted to be proportional to extent to which one must change one's beliefs given that word (Hale, 2001; Smith and Levy, 2013). So the cost of a word is (up to proportionality):

$$C_{\text{surprisal}}(w_i|w_{1:i-1}) = -\log p_L(w_i|w_{1:i-1}), \tag{4.1}$$

where $p_L(\cdot|\cdot)$ is the conditional probability of a word in context in a probabilistic language L.

Standard surprisal assumes that the comprehender has perfect access to a representation of w_i's full context, including the words preceding it in the sentence ($w_{1:i-1}$) and also extra-sentential context (which we leave implicit). But given that human working memory is limited, the assumption of perfect access is unrealistic. We propose that processing cost at a word is better modeled as the cost of belief updates given a *noisy representation* of the previous input. The probability of a word given a noisy context is modeled as the noisy channel probability of the word, assuming that people do noisy channel inference on their context representation (Levy, 2008b; Gibson et al., 2013). Given this model, the expected

processing cost of a word is its expected surprisal over the possible noisy representations of its context.

The noisy-context surprisal processing cost function is thus:[2]

$$C(w_i|w_{1:i-1}) = \mathop{\mathbb{E}}_{V|w_{1:i-1}} \left[-\log p_L^{\text{NC}}(w_i|V)\right] \tag{4.2}$$

$$= -\sum_V p_N(V|w_{1:i-1}) \log p_L^{\text{NC}}(w_i|V) \tag{4.3}$$

where V is the noisy representation of the previous material $w_{1:i-1}$, the **noise distribution** p_N characterizes how memory of previous material may be corrupted, and $p_L^{\text{NC}}(\cdot|\cdot)$ is the noisy-channel probability of a word given a noisy context, computed via marginalization:

$$p_L^{\text{NC}}(w_i|V) = \sum_{w_{1:i-1}} p_L(w_i|w_{1:i-1}) p^{\text{NC}}(w_{1:i-1}|V)$$

with $p^{\text{NC}}(w_{1:i-1}|V)$ computed via Bayes Rule:

$$p^{\text{NC}}(w_{1:i-1}|V) \propto p_N(V|w_{1:i-1}) p_L(w_{1:i-1}).$$

Note here that w_i's cost is computed using its true identity but a noisy representation of the context: from the incremental perspective, w_i is observed now, but context is stored and retrieved in a potentially noisy storage medium. This asymmetry between noise levels for proximal versus distal input differs from the noisy-channel surprisal model of Levy (2011), and is crucial to the derivation of information locality we present in Section 4.4.

Here we use two types of noise distributions for p_N: erasure noise and deletion noise. In **erasure noise**, a symbol in the context is probabilistically erased and replaced with a special symbol E with probability e. In **deletion noise**, a symbol is erased from the sequence completely, leaving no trace. Given deletion noise, a comprehender does not know how many symbols were in the original context; with erasure noise, the comprehender knows exactly which symbols were affected by noise. In both cases, we assume that the application or non-application of noise is probabilistically independent among elements in

[2]Neglecting the implicit proportionality term in Equation 4.1.

the context. We use these concrete noise distributions for convenience, but we believe our results should generalize to larger classes of noise distributions.

4.3 Structural forgetting effects

Here we show that noisy-context surprisal as a processing cost model can reproduce effects that were not previously well-explained by either expectation-based or memory-based theories. In particular, we take up the puzzle of **structural forgetting effects**, where comprehenders seem to forget structural elements of a sentence prefix when predicting the rest of the sentence. The result is that some ungrammatical sentences have lower processing cost and higher acceptability than some complex grammatical sentences: with doubly nested relative clauses, for instance, subjects rate ungrammatical sentence (1) as more acceptable than sentence (2), forgetting about the VP predicted by the second noun (Gibson and Thomas, 1999).

(1) *The apartment$_1$ that the maid$_2$ who the cleaning service$_3$ had$_3$ sent over was$_1$ well-decorated.

(2) The apartment$_1$ that the maid$_2$ who the cleaning service$_3$ had$_3$ sent over was$_2$ cleaning every week was$_1$ well-decorated.

Vasishth et al. (2010) show this same effect in reading times at the last verb: in English native speakers are more surprised to encounter a third VP than not to. However, this effect is language-specific: the same authors find that in German, native speakers are more surprised when a third VP is missing than when it is present. Frank et al. (2016) show further that native speakers do not show the effect in Dutch, but Dutch-native L2 speakers of English do show the effect in English. The result shows that the memory resources taxed by these structures are themselves meaningfully shaped by the distributional statistics of the language.

The verb forgetting effect is a challenge for both expectation-based and memory-based models. Pure expectation-based models cannot reproduce the effect: they have no mechanism for forgetting an established VP prediction and thus they assign small or zero probability to ungrammatical sentences. On the other hand, memory-based models will have to

Rule	Probability
S $\rightarrow$ NP V	1
NP $\rightarrow$ N	$1-m$
NP $\rightarrow$ N RC	mr
NP $\rightarrow$ N PP	$m(1-r)$
PP $\rightarrow$ P NP	1
RC $\rightarrow$ C V NP	s
RC $\rightarrow$ C NP V	$1-s$

Table 4.1: Toy grammar used to demonstrate verb forgetting. Nouns are postmodified with probability m; a postmodifier is a relative clause with probability r, and a relative clause is V-initial with probability s. For practical reasons we bound nonterminal rewrites of NP at 2.

account for why the same structures are forgotten in English but not in German. Here we show that noisy-context surprisal provides the first purely computational-level account for the language-dependent verb forgetting effect. The essential mechanism is that when verb-final nested structures are more probable in a language, then they will be better preserved in a noisy memory representation.

4.3.1 Model of verb forgetting

Table 4.1 presents a toy probabilistic context-free grammar for the constructions involved in verb forgetting. The grammar generates strings over the alphabet of N (noun), V (verb), C (complementizer), P (preposition). We apply deletion noise with by-symbol deletion probability d. So for example, given a prefix NCNCNVV, the prefix can be corrupted to NCNNVV with probability proportional to d, representing one deletion. In that case a noisy-channel comprehender might incorrectly infer that the original prefix was in fact NCNPNVV, and thus fail to predict a third verb.

To illustrate that noisy surprisal can account for language-dependent verb forgetting, we show in Figure 4-1 the differences between noisy surprisal values for grammatical (V) and ungrammatical (end-of-sentence) continuations of prefixes NCNCNVV under parameter settings reflecting the difference between English and German, and compare these differences with self-paced reading times observed after the final verb by Vasishth et al. (2010). Noisy surprisal qualitatively reproduces language-dependent verb forgetting: in

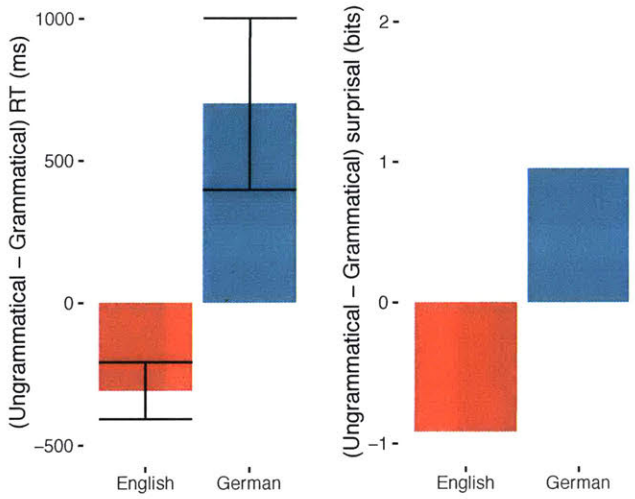


Figure 4-1: Differences in reaction times for ungrammatical continuations minus grammatical continuations, compared to noisy surprisal differences. RT data comes from self-paced reading experiments in Vasishth et al. (2010) in the post-VP region. The noisy surprisal predictions are produced with $d = .2$, $m = .5$, $r = .5$ fixed, and $s = .8$ for English and $s = 0$ for German.

English the ungrammatical continuation is higher surprisal, but in German the grammatical continuation is higher surprisal. The English–German difference in the model is entirely accounted for by the parameter s, which determines the proportion of relative clauses that are verb-initial. In English, most relative clauses are subject-extracted and those are verb-initial, so for English $s \approx .8$ (Roland et al., 2007). German, in contrast, has $s \approx 0$, since its relative clauses are obligatorily verb-final. When verb-final relative clauses have higher prior probability, a doubly-nested RC prefix NCNCVV is more likely to be preserved by a rational noisy-channel comprehender.

The results of Figure 4-1 do not speak, however, to the generality of the model's predictions regarding verb forgetting. To explore this matter, we partition the model's four-dimensional parameter space into regions distinguishing whether noisy-context surprisal is lower for (G) grammatical continuations or (U) ungrammatical continuations for (1) singly-embedded NCNV and (2) doubly-embedded NCNCNVV contexts. Figure 4-2 shows this partition for a range of r, s, m, and d. In the blue region, grammatical continuations are lower-cost than ungrammatical continuations for both singly and doubly embedded con-

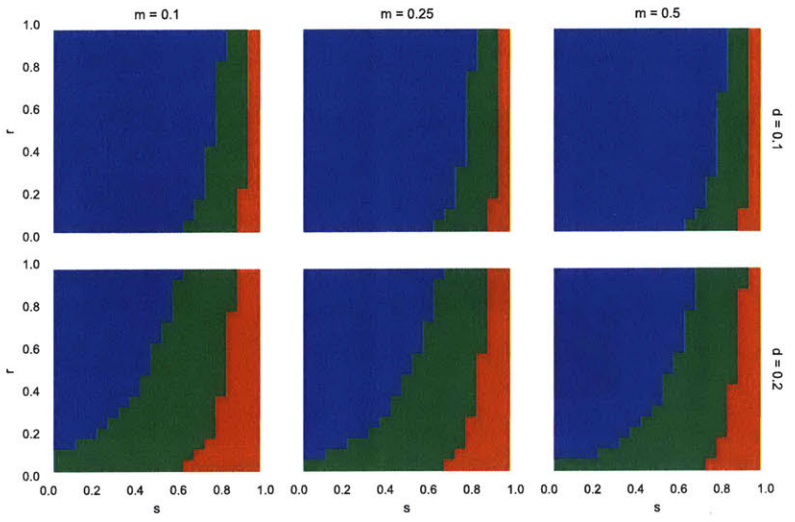


Figure 4-2: Regions of different model behavior with respect to parameters r, s, m, and d (see Table 4.1). Blue: $\mathrm{G_1G_2}$; red: $\mathrm{U_1U_2}$; green: $\mathrm{G_1U_2}$ (see text).

texts, as in German ($\mathrm{G_1G_2}$); in the red region, the ungrammatical continuation is lower-cost for both contexts ($\mathrm{U_1U_2}$). In the green region, the grammatical continuation is lower cost for single embedding, but higher cost for double embedding, as in English ($\mathrm{G_1U_2}$). No combination of parameter values instantiates $\mathrm{U_1G_2}$ (for either the depicted or other possible values of m and d). Thus both the English and German behavioral patterns are quite generally predicted by the model. Furthermore, each language's statistics place it in a region of parameter space plausibly corresponding to its behavioral pattern: the English-type forgetting effect is predicted mostly for high s, the German-type for low s.

The only previous formalized account of language-specific verb forgetting, Frank et al. (2016), showed that Simple Recurrent Networks (SRNs) trained on English and Dutch data partly reproduce the verb forgetting effect in the surprisals they assign to the final verb. Our model provides an explanation of the SRN's behavior, in that it shows how and why this behavior results from any model that predicts words given a lossily compressed representation of previous words. We do not intend it as a competing model to the SRN for this purpose: rather, we propose that noisy-context surprisal is a useful tool for reasoning about the behavior of SRNs at a high level of generalization.

4.4 Information Locality

Here we show how, given an appropriate noise distribution, noisy surprisal gives rise to locality effects. Standard locality effects are related to syntactic dependencies: the claim is that processing is difficult when the parser must make a syntactic connection with an element that has been in memory for a long time. In Section 4.4.1, we derive a more general prediction: that processing is difficult when any elements with high mutual information are far from one another. The effect arises under noisy surprisal because context elements that would have been helpful for predicting a word might have been forgotten. We call this principle **information locality**. In Section 4.4.3, we argue that words in syntactic dependencies have higher mutual information than other word pairs, which leads to a view of dependency locality effects as a special case of information locality effects.

4.4.1 Derivation of information locality

Viewing processing cost as a function of word order, noisy surprisal gives rise to the generalization that cost is minimized when elements with high mutual information are close. We show this by decomposing the noisy surprisal cost of a word into many terms of higher-order mutual information with the context, then showing that applying a certain kind of erasure noise to the context causes these terms to be downweighted based on their distance to the word. Thus the best word order puts the words that have high mutual information with a word close to that word.

Noise Distribution

Noisy surprisal gives rise to information locality under a family of noise distributions which we call **progressive erasure noise**, which is any noise function that erases discrete elements of a sequence with increasing probability the earlier those elements are in the sequence. Formally, in progressive erasure noise, the ith element in a sequence X with length $|X|$ is erased with probability proportional to some monotonically increasing function of how far left that element is in the sequence: $f(|X| - i)$. As a concrete example of progressive erasure noise, consider an exponential decay function, such that the probability that an

element i in X remains unerased is $(1-e)^{|X|-i}$ for some probability e. This exponential decay function corresponds to a noise model where the context sequence is hit with erasure noise successively as each word is processed. Any progressive erasure noise distribution suffices for the derivation here to go through.

Decomposing Surprisal Cost

In noisy surprisal theory, the cost of a word w_i in context $w_{1:i-1}$ is:

$$\begin{aligned} C(w_i|w_{1:i-1}) &= \mathop{\mathbb{E}}_{V|w_{1:i-1}}[-\log p(w_i|V)] \\ &= \mathop{\mathbb{E}}_{V|w_{1:i-1}}[h(w_i) - \text{pmi}(w_i;V)] \\ &= h(w_i) - \mathop{\mathbb{E}}_{V|w_{1:i-1}}[\text{pmi}(w_i;V)], \end{aligned} \tag{4.4}$$

where $h(\cdot)$ is surprisal (here unconditional, equivalent to log inverse-frequency) and $\text{pmi}(\cdot;\cdot)$ is **pointwise mutual information** between two values under a joint distribution:

$$\text{pmi}(x;y) = h(x) + h(y) - h(x,y). \tag{4.5}$$

Essentially, each word has an inherent cost determined by its log inverse probability, mitigated to the extent that it is predictable from context ($\text{pmi}(w_i; w_{1:i-1})$).

Now define the **interaction information** between a sequence of m values $\{a\}$ drawn from a sequence of m random variables $\{\alpha\}$ (McGill, 1955; Bell, 2003) as:[3]

$$i(a_1;...;a_m) = \sum_{n=1}^{m} \sum_{I \in \binom{1:m}{n}} (-1)^{m-n-1} h(a_{I_1},...,a_{I_n}),$$

where the notation $\binom{1:m}{n}$ means all cardinality-n subsets of the set of integers 1 through m. The equation amounts to alternately adding and subtracting the joint surprisals of all subsets of values. For $m = 2$, expanding the equation reveals that mutual information is a

[3]Higher-order information terms are typically defined using a different sign convention and referred to as **coinformation** or **multivariate mutual information** (Bell, 2003). For even orders, interaction information is equal to coinformation. For odd orders, interaction information is equal to negative coinformation. We adopt our particular sign convention to make the generalization of information locality easier to express.

special case of interaction information.

Supposing that the noisy representation of context V is the result of running the veridical context $w_{1:i-1}$ through progressive erasure noise, we can see V as a sequence of values $v_{1:i-1}$, where each v_i is equal to either w_i or the erasure symbol E. Rewriting $\text{pmi}(w_i; V)$ as $\text{pmi}(w_i; v_{1:i-1})$, we can decompose it into interaction informations as follows:

$$\text{pmi}(w_i; v_{1:i-1}) = \sum_{n=1}^{i-1} \sum_{I \in \binom{1:i-1}{n}} i(w_i; v_{I_1}; ...; v_{I_n}), \tag{4.6}$$

The equation expresses a sum of interaction informations between the current word w_i and all subsets of the context values.[4]

[4]To see that Equation 4.6 is true, first note that we can express joint surprisal in terms of interaction information:

$$h(a_1, ..., a_m) = -\sum_{n=1}^{m} \sum_{I \in \binom{1:m}{n}} i(a_{I_1}; ...; a_{I_n}).$$

Now consider the pmi of a value a_i with a sequence $a_{1:i-1}$. Using the decomposition of joint surprisal to expand the definition of pmi in Equation 4.5, we get:

$$\begin{aligned}
\text{pmi}(a_i; a_{1:i-1}) &= h(a_i) + h(a_{1:i-1}) - h(a_i, a_{1:i-1}) \\
&= h(a_i) + h(a_{1:i-1}) - h(a_{1:i}) \\
&= h(a_i) - \sum_{n=1}^{i-1} \sum_{I \in \binom{1:i-1}{n}} i(a_{I_1}; ...; a_{I_n}) \\
&\quad + \sum_{n=1}^{i} \sum_{I \in \binom{1:i}{n}} i(a_{I_1}; ...; a_{I_n}).
\end{aligned}$$

In the final expression, all the terms that do not contain a_i cancel out, leaving:

$$\begin{aligned}
\text{pmi}(a_i; a_{1:i-1}) &= h(a_i) + \sum_{n=0}^{i-1} \sum_{I \in \binom{1:i-1}{n}} i(a_i; a_{I_1}; ...; a_{I_n}) \\
&= h(a_i) + \sum_{n=1}^{i-1} \sum_{I \in \binom{1:i-1}{n}} i(a_i; a_{I_1}; ...; a_{I_n}) - h(a_i) \\
&= \sum_{n=1}^{i-1} \sum_{I \in \binom{1:i-1}{n}} i(a_i; a_{I_1}; ...; a_{I_n}),
\end{aligned}$$

which gives Equation 4.6 when applied to w_i and $v_{1:i-1}$.

Now combining Equations 4.4 and 4.6, we get:

$$C(w_i|w_{1:i-1}) = h(w_i) - \mathop{\mathbb{E}}_{v|w_{1:i-1}} \left[\sum_{n=1}^{i-1} \sum_{I \in \binom{1:i-1}{n}} i(w_i; v_{I_1}; ...; v_{I_n}) \right]$$

$$= h(w_i) - \sum_{n=1}^{i-1} \sum_{I \in \binom{1:i-1}{n}} \sum_{v} p_N(v|w_{i:i-1}) i(w_i; v_{I_1}; ...; v_{I_n}).$$

Now if any element of an interaction information term is E, then that whole interaction information term is equal to 0. This happens because the probability that an element is erased is independent of the identity of other elements in the sequence, and thus E has no interaction information with any subset of those elements. That is, $i(w_i; v_{I_1}; ...; v_{I_n}) = 0$ unless $v_{I_j} = w_{I_j}$ for all j. This allows us to write:

$$C(w_i|w_{1:i-1}) = h(w_i) - \sum_{n=1}^{i-1} \sum_{I \in \binom{1:i-1}{n}} i(w_i; w_{I_1}; ...; w_{I_n}) \sum_{m \in \{0,1\}^{i-1}} p_N(m) m_I$$

where the variable m ranges over bit-masks of length $i-1$, and m_I is equal to 1 when all indices I in m are equal to 1, and 0 otherwise. Now $\sum_{m \in \{0,1\}^{i-1}} p_N(m) m_I$ is the total probability that all of a set of indices I survives erasure. Thus, informally:

$$C(w_i|w_{1:i-1}) = h(w_i) - \sum_{n=1}^{i-1} \sum_{I \in \binom{1:i-1}{n}} p_N(I \text{ survives}) i(w_i; w_{I_1}; ...; w_{I_n}). \tag{4.7}$$

That is, the cost of a word is its inherent cost minus its interaction informations with context, which are weighted by the probability that all elements of those interactions survive erasure.

Under progressive erasure noise, the probability that a subset of variables is erased in-

creases the farther left those variables are in the context. Therefore, Equation 4.7 expresses information locality: context elements which are predictive of w_i will only get to mitigate the cost of processing w_i if they are close to it. The surprisal-mitigating effect of a context element on a word w_i decreases as that element gets farther from w_i.

4.4.2 Noisy-context surprisal and dependency locality

Memory-based models of sentence processing account for apparent **dependency locality effects**, which is processing cost apparently arising from two words linked in a syntactic dependency appearing far from one another (Gibson, 1998). Dependency length has been proposed as a rough measure of comprehension and production difficulty, and studied as a predictor of reaction times (Grodner and Gibson, 2005; Demberg and Keller, 2008; Mitchell et al., 2010; Shain et al., 2016), and also as a theory of production preferences and linguistic typology, under the assumption that people prefer to produce sentences with short dependencies (dependency length minimization) (Hawkins, 1994; Gildea and Temperley, 2010; Futrell et al., 2015b; Rajkumar et al., 2016).

Dependency locality follows from information locality if words linked in a syntactic dependency have particularly high mutual information. To see this, consider only the lowest-order interaction information terms in Equation 4.7, truncating the summation over n at 1. We can write

$$C(w_i|w_{1:i-1}) = h(w_i) - \sum_{j=1}^{i-1} f(i-j)\text{pmi}(w_i; w_j) + R,$$

where R collects all the interaction information terms of order greater than 2, and $f(d)$ is the monotonically decreasing survival probability of a d-back word, described in Section 4.4.1. The effects of R are bounded because higher-order mutual information terms are more penalized by erasure noise than lower-order terms, simply because large sets of context items are more likely to experience at least one erasure.

If the effects of R are negligible, then the cost of a whole utterance w as a function of

word order is determined only by pairwise information locality:

$$C(w) \approx \sum_{i=1}^{|w|} h(w_i) - \sum_{i=2}^{|w|} \sum_{j=1}^{i-1} f(i-j)\text{pmi}(w_i; w_j).$$

If words linked in a dependency have higher mutual information than words that are not, then the processing cost as a function of word order is a monotonically increasing function of dependency length. Under this assumption, for which we provide evidence below, dependency locality effects can be seen as a special case of information locality effects. As a theory of production preferences or typology, processing cost as a monotonically increasing function of dependency length suffices to derive some of the major predictions of dependency length minimization (Ferrer i Cancho, 2015).

4.4.3 Mutual information and syntactic dependency

We have shown that noisy-context surprisal derives information locality, and argued that dependency locality can be seen as a special case of information locality. However, deriving dependency locality requires a crucial assumption that words linked in a dependency have higher mutual information than those words that are not. Here I provide empirical evidence that this is true. For a more theoretical perspective, see Section 5.2.

To test this assumption, we calculated mutual information between wordforms in various dependency relations in the Google Syntactic n-gram corpus (Goldberg and Orwant, 2013). We compared the mutual information of content words in a direct dependency relationship to content words in grandparent–grandchild and sister–sister dependency relationships. Mutual information was estimated using maximum likelihood estimation from frequencies, treating the corpus as samples from a distribution over (head, dependent) pairs. In order to exclude nonlinguistic forms, we only included wordforms if they were among the top 10000 most frequent wordforms in the corpus. The direct head–dependent frequencies were calculated from the same corpus as the grandparent-grandchild frequencies, so that all mutual information estimates are affected by the same frequency cutoff. The results are shown in Table 4.2: direct head–dependent pairs indeed have the highest mutual

Relation	MI (bits)
Head–dependent	1.79
Grandparent–dependent	1.34
Sister–sister	1.19

Table 4.2: Mutual information over wordforms in different dependency relations in the Syntactic n-gram corpus. The pairwise comparison of head–dependent and grandparent–dependent MI is significant at $p < 0.005$ by Monte Carlo permutation tests over n-grams with 500 samples. The comparison of head–dependent and sister–sister MI is not significant.

information.

To test the crosslinguistic validity of this generalization about syntactic dependency and mutual information, we calculated mutual information between the distributions over POS tags for dependency pairs of 43 languages in the Universal Dependencies corpus (Nivre et al., 2016). For this calculation, we used mutual information over POS tags rather than wordforms to avoid data sparsity issues. The results are shown in Figure 4-3. Again, we find that mutual information is highest for direct head–dependency pairs, and falls off for more distant relations. These results show that two words in a syntactic dependency relationship are more predictive of each other than two words in some other kinds of relationship.

We also compared the mutual information of word pairs in and out of dependency relationships while controlling for distance. This test has a dual purpose. First, it allows us to control for distance when claiming that words in dependency relationships have high mutual information. Second, it allows us to test a simple prediction of information locality as applied to language production: that words with high mutual information should be close together. For pairs of words (w_i, w_{i+k}), we calculated the pmi values among POS tags of the words. Figure 4-4 shows the average pmi of all words at each distance compared with the average pmi of the subset of words in a direct dependency relationship at that distance. In all languages, we find that words in a dependency relationship have higher pmi than the baseline, especially at close distances. Furthermore, we find that words at close distances tend to have higher pmi, regardless of whether they are in a dependency relationship.

Figure 4-3: Mutual information over POS tags for dependency relations in the Universal Dependencies 1.4 corpus, for languages with over 500 sentences. All pairwise MI comparisons are significant at $p < 0.005$ by Monte Carlo permutation tests over dependency observations with 500 samples.

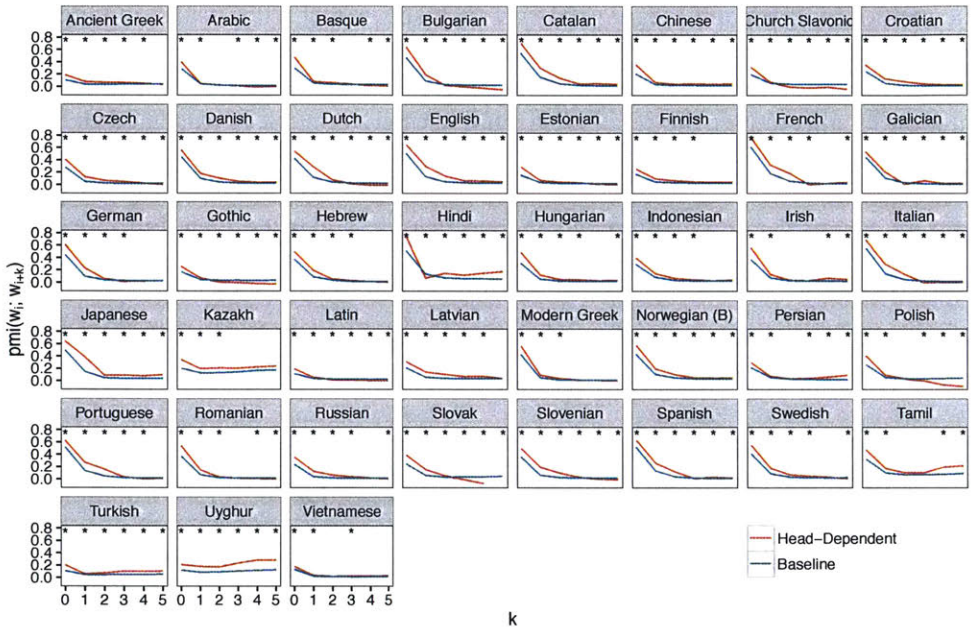


Figure 4-4: Average pointwise mutual information over POS tags for word pairs with k words intervening, for all words (baseline) and for words in a direct dependency relationship. Asterisks mark distances where the difference between the baseline and words in a dependency relationship is significant at $p < 0.005$ by Monte Carlo permutation tests over word pair observations with 500 samples.

4.4.4 Discussion

Information locality can be seen as a decay in the effectiveness of contextual cues for predicting words. Precisely such a decay in cue effectiveness was found to be effective for predicting entropy distributions across sentences in Qian and Jaeger (2012), although that work did not distinguish between an inherent, noise-based decay in cue effectiveness or optimized placement of cues.

The result of Gildea and Jaeger (2015), which shows that word orders in languages are optimized to minimize trigram surprisal of words, can be taken to show maximization of information locality under the noise distribution where context is truncated deterministically at length 2. Whereas Gildea and Jaeger (2015) treat dependency length minimization and trigram surprisal minimization as separate factors, under the view in this paper these two phenomena emerge as two aspects of information locality. In general, the mutual information of linguistic elements has been found to decrease with distance (Li, 1989; Lin and Tegmark, 2016), although this claim has only been tested for letters, not for larger linguistic units such as morphemes. The fact that linguistic units that are close typically have high mutual information could result from optimization of word order for information locality.

The idea that syntactically dependent words have high mutual information is also ubiquitously implicit in probabilistic models of language and in practical NLP models (e.g., Collins, 2003). For example, it is implied by head-outward generative models (Lafferty et al., 1992; Eisner, 1996, 1997; Klein and Manning, 2004), the first successful models for grammar induction. Mutual information has been used directly for unsupervised discovery of syntactic dependencies (Yuret, 1998) and evaluation of dependency parses (de Paiva Alves, 1996), as well as commonly for collocation detection (Church and Hanks, 1990). In addition to providing evidence for a crucial assumption in the derivation of information locality, our results also give evidence backing up the theoretical validity of such models and methods.

The derivation of information locality given here assumed progressive erasure noise for concreteness, but we believe it should be possible to derive this generalization for a large family of noise distributions.

4.5 Further applications

The idea of information locality as a factor influencing languages has many possible applications. The basic prediction is that linguistic elements that predict each other should be close. I believe this can predict and explain many interesting word order phenomena.

Adjective order One particularly intriguing universal of word ordering across languages is the relative closeness of adjectives to nouns (Hetzron, 1978; Dixon, 1982; Sproat and Shih, 1991). When multiple adjective modify a noun in English, they come in a fairly rigid order determined by semantic class, for example *big blue box* is a much more common and natural-seeming order than *blue big box*. In languages where adjectives follow nouns, the same ordering constraints appear in reverse.

Information locality predicts that adjectives that have high pmi with nouns should be close to those nouns. Thus if adjective orders can be explained in terms of mutual information, then adjective ordering constraints emerge as one instance of the very general constraint of information locality. This idea suggests it would be fruitful to test pmi as a predictor of relative orderings of adjective and nouns in corpora and ratings studies. It may also be possible to connect pmi at a theoretical level with concepts that have been claimed to predict adjective ordering, such as subjectivity (Scontras et al., 2017) and inherentness (Ziff, 1960).

Arguments and adjuncts Adjuncts are typically placed farther from their heads than arguments. They also seem to be less subject to dependency locality effects in processing: Shain et al. (2016) find that a dependency locality theory which does not factor in distance to adjuncts does better at predicting reading times than a theory that includes distances to adjuncts. If adjuncts have lower pmi with their heads than arguments, then information locality would explain the fact that they are often placed farther from their heads than arguments. In support of this idea, high mutual information has been taken as a signal for unsupervised discovery of argument relationships in NLP work (Church and Hanks, 1990; Aldezabal et al., 2002; Abend and Rappoport, 2010).

Word order change If information locality is a pressure affecting languages, then we should see in historical corpora that a set of words that have high pmi at one date appear closer together at a later date. It would be possible to test this straightforwardly, e.g. in the Google Books corpus. More generally, information locality provides a mathematical theory of word order change in grammaticalization (Hopper and Traugott, 2003). It predicts that when linguistic elements start to covary (indicating that they have a new meaning), they should become closer to each other.

Such a process appears to have happened, for example, the formation of the English *going to* future tense. As *going to* came to be a future tense marker, the words stopped admitting an adverb between them. In modern English, *going quickly to see you* means that a literal going event is taking place; *going to*. We can see the words *going* and *to* as becoming more and more tightly bound to each other by information locality as they covaried more and more statistically.

4.6 Conclusion

We have introduced a computational-level model of incremental sentence processing difficulty based on the principle that comprehenders have uncertainty about the previous input and act rationally on that uncertainty. Noisy-context surprisal accounts for key effects predicted by expectation-based and memory-based models, in addition to providing the first computational-level explanation of language-specific structural forgetting, which involves subtle interactions between memory and probabilistic expectations. Noisy-context surprisal also leads to a general principle of information locality offering a new interpretation of syntactic locality effects, and leading to broader and potentially different predictions than purely memory-based models.

Here we have used qualitative arguments and have used different specific noise distributions to make different points. Our aim has been to argue for the theoretical viability of noisy-context surprisal, without committing the theory to a particular noise distribution. We believe our predictions will be derivable under very general classes of noise distributions, and we plan to pursue these more general derivations in future work.

A more psychologically accurate model will likely use a more nuanced noise distribution than the simple decay functions in this paper, which do not capture the subtleties of human memory. Many studies of the effects of memory on linguistic processing have emphasized the importance of **similarity-based interference**, where memory retrieval appears to be inaccurate in that it confuses multiple similar context items (Gordon et al., 2001, 2006). These effects could be modelled in our framework via a noise model that swaps the positions of words based on their similarity. Also, simple decay functions to not capture memory retrieval effects of the kind described in Anderson and Schooler (1991), where different items in a sequence have different propensities to be forgotten, in accordance with rational allocation of resources for retrieval.

Seen as a noise distribution, this memory model implies that the erasure probability of a word is a function of the word's identity, and not only the word's position in the sequence as in Section 4.4.1. Including such noise distributions in the noisy-context surprisal model could provide a rich set of predictions to test the model more extensively.

Chapter 5

Conclusion

This work has argued that we can explain syntactic patterns in languages in terms of communicative efficiency under processing constraints. The basic prediction is that, assuming that human language processing is incremental and memory-constrained, we expect locality constraints on words that depend on each other for meaning (dependency locality), or words that predict each other (information locality).

This approach motivated a quantitative study of syntax using information theoretic tools in Chapter 2. In Chapter 3, I showed large-scale crosslinguistic evidence for dependency locality, and in Chapter 4, I derived the more general theory of information locality from a model of incremental processing with noisy context representations.

To conclude, I will expand on some theoretical ideas left hanging from the content chapters, and also offer further speculations on applications of these locality ideas.

5.1 Typological predictions from surprisal alone?

In general, in this thesis I am interested in processing theories that make predictions about word order, and I have shown in Chapter 3 that the predictions of dependency locality theory are very successful in this regard. Given that the two known major sources of processing difficulty in comprehension are surprisal and dependency locality, it makes sense to ask whether minimizing surprisal makes useful predictions about word order. In this section, I will argue that surprisal makes no interesting predictions about word order.[1] However, minimizing noisy-context surprisal, while fixing the amount of information about meaning to be transmitted, does result in new predictions.

Unlike the idea of minimizing dependency length, the idea of minimizing surprisal in order to obtain processing efficiency runs into basic philosophical problems. With dependency length, it is possible to have languages with higher or lower dependency lengths conveying the same amount of information. With surprisal, because bits of surprisal are an upper bound on bits of meaning conveyed, all reductions of the total surprisal of a sentence imply that less information might be conveyed. In the limit, naïve surprisal minimization leads to a trivial language that only contains one sentence, which has probability 1 and

[1] For similar argumentation, see Section 2.8.3 of Levy (2005).

surprisal 0. Clearly such a language runs afoul of another desideratum for language: in addition to being easy to process it must be able to convey information. Holding the amount of meaningful information expressed constant, no surprisal minimization is possible except the reduction of free variation (see Section 1.4.3).

Furthermore, I will show below that in an important sense, all possible word order systems convey the same amount of information.

The expected processing cost associated with a language L is generally the expected cost of its sentences:

$$C_{\text{lang}}(L) = \sum_{s \in L} p_L(s) C_{\text{sent}}(s).$$

In surprisal theory, the processing cost of a word in a sentence is the information content of the word given preceding context:

$$C_{\text{word}}(w_i | w_{1:i-1}) = -\log p_L(w_i | w_{1:i-1}).$$

Since surprisal is linear, the processing cost of a sentence is the sum of costs of the words, which ends up being equal to the surprisal of the sentence as a whole:

$$\begin{aligned} C_{\text{sent}}(s) &= \sum_{i=1}^{|s|} -\log p_L(w_i | w_{1:i-1}) \\ &= -\log \prod_{i=1}^{|s|} p_L(w_i | w_{1:i-1}) \\ &= -\log p_L(s). \end{aligned}$$

So the cost of a language under surprisal is just the entropy of the language:

$$\begin{aligned} C_{\text{lang}}(L) &= -\sum_{s \in L} p_L(s) \log p_L(s) \\ &= H(L). \end{aligned}$$

It follows that the expected processing cost of the language in surprisal theory is not affected by *any* word order transformation which preserves the overall information content

of a language. If a language has deterministic unambiguous word order rules conditional on meaning, then any set of deterministic unambiguous word order rules for this language will have the same total information content. Thus for fixed meanings, all languages which are one-to-one mappings of meaning to form will have the same expected surprisal cost. In general, for a given level of information content a and uncertainty b about word order given meaning, all transformations which preserve a and b have the same expected surprisal cost. Thus surprisal minimization tells us nothing about the utility of word order for expressing meanings, beyond that extraneous variation should be minimized.

It is possible that minimization of variance in surprisal, or minimization of some non-linear function of surprisal, would still result in interesting word order predictions. Such predictions have been pursued in Maurits et al. (2010).

It is worth comparing this result with work such as Gildea and Jaeger (2015), which has argued that word orders minimize surprisal cost. In that work, surprisal cost is defined using an n-gram model, which does not give probabilities for words conditional on the full prefix before those words. The limited conditioning information for each word means that n-gram surprisal values do not correspond to sentence probabilities. I would argue that n-gram surprisal is actually a form of noisy-context surprisal, where the context is truncated at $n-1$ words. From this perspective, we see that minimizing n-gram surprisal corresponds to a form of information locality, in that putting minimizing n-gram surprisal means putting words that predict other words within a window of $n-1$ of each other.

5.2 Dependencies and mutual information

Connecting dependency locality theory to information locality theory required the assumption, empirically verified in Section 4.4.3, that dependency pairs have the highest mutual information compared to other pairs of words. I will call this the **HDMI hypothesis**: that heads and dependents have high mutual information. Here I discuss some theoretical issues raised by the HDMI hypothesis, and sketch some general reasons why we might expect it to be true.

5.2.1 HDMI from head-outward generative models

First I will argue that the HDMI hypothesis follows from a language being well-described by head-outward generative dependency models. I will argue that under a head-outward model, the true parse of a corpus has higher head-dependent mutual information than any other parse of the corpus.

Suppose we have a corpus of sentences L and a parse t, which is an arrangement of the words in sentences into dependency tree structures. The probability of the corpus under a head-outward model is (Eisner, 1996, 1997):

$$\begin{aligned} p(L|M,t) &= \prod_{s \in L} \prod_{w \in s} p_M(w|t(s,w)) \\ &= \prod_{(h,d) \in L_t} p_M(d|h), \end{aligned} \tag{5.1}$$

where M is a conditional distribution giving the probability of a dependent wordform given a head wordform, $t : S \times W \rightarrow W$ is a mapping from a word to its head word within a sentence, and L_t is a reduction of the corpus L into a series of head–dependent pairs according to t.

Now let's think about the mutual information between heads and dependents in the corpus L with parse t. Let $I(H_t; D)$ represent the mutual information between dependents D and heads H_t as identified by the parse t. Under the head-outward model, each value of D was generated directly conditional on the corresponding value in H_t. The **data processing inequality** (Cover and Thomas, 2006) holds that for any joint distribution (A, B) where B is generated directly from A, and for all functions f, $I(A; B) \leq I(f(A); B)$. For any parse t' of L, we can see it as defining a mutual information $I(H_{t'}; D) = I(f(H_t); D)$, where f is a function that replaces a head in parse t with its corresponding head in parse t'. Thus by the data processing inequality, $I(H_{t'}; D) \leq I(H_t; D)$.

This argument shows that under head-outward models, heads and dependents represent the word pairs (forming a tree over words in sentences) that maximize mutual information. Thus the HDMI property follows generally from these models.

Relation to PCFGs

The HDMI hypothesis is deeply connected to head-outward generative models as descriptions of linguistic distributions. If we have good reason to suspect language does follow such a distribution, then we should expect the HDMI hypothesis to be true. Here I address the question of how head-outward generative models relate to other common probability models for language data.

Head-outward generative models are a subset of probabilistic grammars where rewrite probabilities are multiplied to get the probability of a derivation. The most prominent kind of probabilistic grammar in this sense is the ubiquitous probabilistic context-free grammar (PCFG) (Suppes, 1970; Sankoff, 1971). In head-outward generative dependency models, each dependency arc in a sentence is conditionally independent of the rest of the sentence given its head. More generally, a sentence is considered to be built out of many parts consisting of dependency arcs and words, and the probability of a sentence is the product of the probabilities of those parts. In this way, head-outward generative dependency models are similar to probabilistic context-free grammars, and in fact can be reduced to them (Johnson, 2007).

5.2.2 Why should sentence probabilities factor nicely?

Here I discuss the relationship between language at is described with a grammatical formalism and language as a probability distribution over sentences that we might observe in a corpus. It is commonly assumed that the distribution over strings in a context-free language is given by a PCFG, but I wish to problematize that assumption (see also Kornai (2011) for further problematization).

We can see a dependency parse (or a CFG parse) as encoding a sequence of rules that are applied to derive a sentence. The set of rules provides a description of the set of strings in a language. But when we think of a language as a tool for expressing meanings, it is not immediately obvious that the *probabilistic* distribution over strings in such a language would be well described by a model where the probabilities of the rules are simply multiplied together, as in head-outward dependency models and PCFGs.

A dependency grammar or a context-free grammar only describes the *support* of the probability distribution over sentences. In principle there could be all kinds of covariance among rules, which would mean that rule applications are not independent in probability. Thus the probability of a sentence would not be the simple product of the probabilities of the rules that generate it.

To see this, consider a context-free language L containing all the sentences of English which admit a context-free description, and suppose that the derivations of these sentences exist in a one-to-one mapping with meanings. Then the probability of encountering a sentence with derivation d expressing meaning m is the probability that a speaker wants to express m. The probability distribution over m is unconstrained in this construction.

As an example of how unconstrained m can lead to non-PCFG distributions over context free stringsets, consider the language L spoken by speakers in a desert, where the distribution over m in their environment is such that they say *I want water* with very high probability and all other sentences with nominal probability. The sentence *I want water* can have the same derivation as in a description of English, with multiple rules producing the subject NP, the matrix VP, and so on, as in the grammar in the first column of Table 5.1. Yet all the probability mass is on one sentence, and not on the other sentences that share rules with it. I am essentially describing a spike-and-slab distribution over context-free derivations, with the spike on the derivation of *I want water* and the slab on all other derivations. This example is meant to show just how much the probability distribution over context-free sentences can vary as a function of m.

To flesh this example out more, suppose we have fixed the context-free description of the language L (the rule column in Table 5.1), such that the derivation of any sentence is fixed. For example, the sentence *I want water*, which by construction has abnormally high probability, has a derivation with the application of multiple rules, just as in a description of English. If we take this fixed grammar and assign probabilities to the rules (choosing the probability column in Table 5.1, but keeping the rule column fixed), it becomes immediately obvious that it is not possible to reproduce the distribution over sentences where *I want water* has high probability and everything else has uniformly low probability, because the probability of the derivation of *I want water* is the product of the probability of

	Rule	Probability
1	S $\rightarrow$ NP VP	p_1
2	NP $\rightarrow$ PRP	p_2
3	NP $\rightarrow$ DT NN	p_3
4	NP $\rightarrow$ NN	p_4
5	VP $\rightarrow$ VB	p_5
6	VP $\rightarrow$ VB NP	p_6
7	PRP $\rightarrow$ I	p_7
8	VB $\rightarrow$ want	p_8
9	NN $\rightarrow$ water	p_9
...	...	...

Table 5.1: Non-exhaustive (P)CFG for example language L.

its rules $p_1p_2p_7p_6p_8p_4p_9$, and if these probabilities are high then they will also give higher probability to sentences which are partially similar to *I want water* such as *I want food.* It is thus very unclear that the fact that a language is well-described by a particular context-free grammar implies that its sentences follow a PCFG distribution with those rules, or even that they factorize in any way with respect to the grammar rules.

In the same way that a CFG description of a language does not imply that it should follow a PCFG distribution, it does not follow immediately that a dependency grammar description of a language implies that it should follow a head-outward generative distribution. In general, the question is why rule applications appear to be probabilistically independent in the language as a distribution.

5.2.3 Coding Factorization Conjecture

I do not have an answer for why sentence probabilities appear to be well-modelled as products of rule probabilities. But I would like to advance a conjecture which I call the **Coding Factorization Conjecture**, which is that if an efficient code for a meaning distribution M works by successively applying rewrite rules, then the rules should correspond to dimensions of meaning such that the rule application probabilities are maximally independent of each other in the language as a distribution. That is, the rules should represent a **factorization** of M: a set of independence assumptions about M, or equivalently a representation of M as a set of maximally independent random variables, as might be encoded

in a Bayes net. If this conjecture is true, then the link between syntactic dependency and mutual information—and between context-free languages and PCFG p-languages—could be established through communicative efficiency.

5.3 Context-independence and context-dependence

One overall impression that comes out of this work is that languages should avoid context-dependence, or at least that when context-dependence exists, it should be among utterance elements that are close. By **context-dependence** I mean that any form of dependency exists between a word and the context it appears in: context-dependence is when either the probability or the interpretation of a linguistic element depends on the context it appears in. The result demonstrated in Section 1.5.5 shows that context-dependence of interpretation results in low utility languages under processing constraints, and Chapter 4 shows that statistical dependence of words on their contexts results in increased processing cost. The basic intuition is just that context-dependence is harmful when speakers and listeners can't remember context. Here I discuss implications of that idea.

5.3.1 Why is language context-dependent?

The question arises of why natural languages do seem to have so much context-dependence, although they still have far less than what is theoretically possible or what is desirable in certain digital codes (see Section 1.3.5).

Here I will sketch a proposals for why context-dependence exists in natural language, based on nonstationarity of the meaning distribution M.

If we assume M is nonstationary, then context-dependence arises because M has shifted away from a state that allowed a certain context-independent code to exist. Suppose $\mathcal{L}$ is a context-independent code for M, meaning that the probability and interpretation of each word is independent of its context. Such a code represents a factorization of M, because the independent components of L correspond to independent components of some parameterization of M. But if M changes in such a way that those independence assumptions are no longer valid, then the components of the resulting L will no longer be independent

in the language as a distribution. If the process by which $\mathcal{L}$ adapts to M is slow, then no language may achieve full context-independence.

This idea leads to a prediction that we might expect more long-range context-dependence in languages spoken in environments that are changing quickly. It is possible that morphological complexity corresponds to a lack of long-range context-dependence, because it makes words more self-contained in their distribution and interpretation; in that case, the observed higher morphological complexity of languages spoken by small societes (Lupyan and Dale, 2010) might result from a more stationary meaning distribution in those societies. Some parts of language structures change more quickly than others (Dediu and Cysouw, 2013); this may be the result of these structures corresponding to more or less stationary parts of the meaning distribution. See also Baronchelli et al. (2013) for similar ideas.

5.3.2 Duality of patterning

A distinctive property of natural language is **duality of patterning**, the fact that phonemes which bear no consistent relationship to meaning[2] combine into morphemes, which carry more or less transparent meanings (Hockett, 1959). That is, within a morpheme, the interpretation of a phoneme is maximally context-dependent; the interpretation of a whole morpheme itself, on the other hand, is highly context-independent. Previous work in evolutionary linguistics (Nowak and Krakauer, 1999; Tria et al., 2012; Spike et al., 2016) has shown that duality of patterning can emerge in part as a mechanism for robust communication in noise, because it is easier to discriminate between collections of a small number of symbols than to discrimate between many unrelated symbols. What information locality adds is that it provides a new reason for morphemes to consist of *contiguous* sequences of phonemes, and the reason that larger and larger linguistic units show less and less internal contiguity.

[2]See the literature on *phonaesthemes* for exceptions (Bergen, 2004).

5.3.3 Endocentricity

A distinctive property of natural language syntax, and a fact which I have relied on extensively in this work, is that the combinatorial and compositional properties of utterances can be well-described in terms of head–dependent relationships among words. This is the same as saying that the behavior of a phrase is largely determined by a distinguished element in it, its head. The fact that syntax operates over heads (for the most part) is the principle of **endocentricity** (see Section 3.1 for more details).

Information locality can provide a partial explanation for endocentricity. Endocentricity basically means that the context-dependence of a word is mostly concentrated on exactly one other word. That is, the probability and interpretation of a word depends primarily on exactly one other word, and the rest of the words are irrelevant. When this is true in a language, let us call it a 1-endocentric language. Let's compare language where every word is dependent on exactly one other word to a language where every word is dependent on exactly 2 other words: call this a 2-endocentric language. Now if incremental memory constraints affect comprehension and production, then I claim the 1-endocentric language is better than the 2-endocentric language. This follows given erasure noise affecting context representations. On average, any set of 2 words is more likely to suffer erasure of at least one element than any set of 1 words. In a 2-endocentric language, a context-dependent word has a greater risk of incorrect interpretation because one of the crucial context words was forgotten.

Taken to its natural conclusion, this argument actually shows that languages should be 0-endocentric: i.e. every part of the utterance should be context-independent. The arguments in Section 5.3.1 might explain why this ideal has not been reached, and it might explain why some languages and constructions go beyond 1-endocentricity.[3]

[3]Another important property of endocentricity is that linguistic heads and dependents typically form a tree structure. In part, this is a definitional matter: those sequences of words whose head–dependent relations form trees are called sentences.

5.4 Incremental sequence samplers

In this section I will discuss an alternative derivation of the idea of information locality from incremental planning algorithms for generating sequences.

In the framework I have argued for, the fundamental mechanism by which processing factors influence languages is that language comprehension and production are done relative to *approximations* of a language reflecting information processing constraints (Section 1.5). I make this more concrete by claiming that when speakers and listeners are processing language incrementally, they are doing so using *approximate samplers* for the language (Section 5.4.1). Then I discuss possible approximate sampling algorithms for distributions over sequences, which are constrained by incrementality (Sections 5.4.2 and 5.4.3). I show conditions under which these incremental sampling algorithms favor sequence distributions that have information locality (Section 5.4.4). Finally I discuss connections with recent advances in deep reinforcement learning (Section 5.4.5).

5.4.1 Samplers

The difference between the language as an agent knows it and the language as an agent applies it is the same as the distinction between a distribution and a sampler. A **sampler** for a distribution is an algorithm which generates samples from the distribution. Some distributions that are easy to characterize may be hard to generate samples from, and some distributions that are easy to sample from may be hard to calculate analytically, due to intractable normalizing constants (MacKay, 2003). Crucially, a sampler may be *approximate*; in fact, approximate samplers are often necessary in order to use certain distributions in practical applications (Neal, 1993). Approximate samplers often contain representations of the distribution they are attempting to sample from, even if they cannot produce samples from it precisely.

In the context of this thesis, I am proposing to model language as it is used in a community of agents. The agents may know a language perfectly, but they must implement samplers to generate and encode actual samples from the language. These samplers may contain errors or operate under heavy resource constraints, hence introducing unavoidable

approximation error.

The task of a speaker is to take a known language $\mathcal{L}$—a distribution over utterances given a meaning—and generate samples from it. If she does so using an approximate sampler, then that sampler actually generates from some other distribution, which is $\mathcal{L}_s$. Similarly, the task of a listener is to take a known language $\mathcal{L}$ and infer the meanings that were intended under it. Assuming that the listener does this inference using Bayes rule, the various expectations that must be computed may be done by Monte Carlo approximation, drawing samples using an approximate sampler that actually generates from the approximating distribution $\mathcal{L}_l$.

5.4.2 Incremental samplers for sequences

A language as a code $\mathcal{L}$ generates sequences of symbols. We think human language processing is highly incremental, so in a sampling framework, we should think about samplers that generate sequences by taking a context representation c and figuring out what word w to generate next. I call this kind of sampler an **incremental sequence sampler**.

In this section, I discuss consequences of thinking of human language performance in terms of incremental sequence samplers. The discussion results in a new perspective on information locality, and suggests some practical algorithms for natural language processing.

Let us consider incremental sequence samplers for distributions over W^*. An incremental sequence sampler consists of two subsidiary samplers d and e. d implements a probabilistic **decoding function** $d : C \to W$ which produces an element of W conditional on a context $c \in C$. e implements a probabilistic **encoding function** $e : W \times C \to C$ which incorporates a word w into a context representation $c \in C$, producing a new context representation $c' \in C$. The decoding function chooses what symbol to produce next, and the encoding function chooses how to represent what has been produced so far. Given these two parameters, the sampler produces a sequence $\mathbf{w} \in W^*$ by taking a context c, generating a word $w = d(c)$, and then generating a new context $c' = e(w, c)$. This process repeats recursively until the end-of-sequence symbol is generated. This algorithm is summarized in Algorithm 2. Thus the sampler produces $\mathbf{w}$ with probability:

Algorithm 2 Algorithm to generate a sequence given an incremental sequence sampler parameterized by probabilistic functions e and d. ϵ is the empty sequence.

function GENERATEFROM(c)
  $w \leftarrow d(c)$
  if $w = \#$ **then**
    return ϵ
  else
    $c' \leftarrow e(w, c)$
    return $w \cdot$ GENERATEFROM(c')
  end if
Result is GENERATEFROM(#).

$$p_d^e(\mathbf{w}) = p_d^e(\mathbf{w}|\#), \text{where}$$

$$p_d^e(\mathbf{w}|c) = p_d(w_1|c) \times \begin{cases} 1 & \text{if } w_1 = \# \\ \mathop{\mathbb{E}}_{c' \sim e(w_1, c)} \left[p_d^e(w_{2:|\mathbf{w}|}|c') \right] & \text{otherwise.} \end{cases}$$

Now consider a sampler meant to produce samples from a probability distribution over sequences $\mathbf{W}$. Such a sampler should choose e and d to minimize KL divergence from the sampler distribution to the target distribution:

$$\operatorname*{argmin}_{e,d} D_{\mathrm{KL}}(p_d^e \rightarrow p_{\mathbf{W}}), \tag{5.2}$$

where p_d^e is the probability distribution over sequences generated by the sampler. A sampler that minimizes Equation 5.2 is an autoencoder for $\mathbf{W}$.

I have described a sampling algorithm that is an autoencoder for sequences. But the speaker and listener are really interested in finding the best representation of linguistic sequences for the purpose of *encoding and decoding meaning*. It is not immediately obvious that the best strategy for this goal is to find the representation that is best for autoencoding these sequences. However, I argue here that learning a representation that can autoencode sequences means that we maximize an upper bound on how much information about meaning is present in the representation of the sequences. We saw previously that for the listener's utility (Section 1.5), learning to autoencode sequences has the effect of pushing up a lower bound on utility (Equation 1.22). In general, if an autoencoder's context representation c contains represents k bits of information about $\mathbf{w}$, then it can encode at most

k bits of information contained in w about any additional variable M. Therefore, when we want to make a representation of meaning in linguistic sequences, it is justifiable to train a representation to autoencode those linguistic sequences, because this representation maximizes an upper bound on informativity about meaning.

5.4.3 Planning for incremental sequence samplers

In this section, I will show how to describe an incremental sequence sampler as a planning algorithm. In the next section, I will use the concepts built up here to show two conditions under which incremental sequence samplers favor sequence distributions that have information locality.

An incremental sequence sampler is parameterized by a probabilistic decoding function $d : C \to W$ and an encoding function $e : W \times C \to C$. If we want the function d to produce a valid continuation of a sequence given an encoded context c, then d must solve a planning problem (Lashley, 1951; MacDonald, 2013). Similarly, if we want e to produce a context representation that enables the correct completion of a sequence, and if e is resource-constrained, then e must solve a planning problem in order to know what context information is worth keeping (the "memory for what is to come", Rosenbaum et al. 2007).

Here I characterize d and e as planning algorithms that maximize a Q-function (Sutton and Barto, 1998).

Suppose that when the encoder e is faced with a context c and a word w, it can deploy a number of different **encoding actions** $a \in A, a : W \times C \to C$. For example, the set of encoding actions may consist of an action $a_{\text{keep}}(w, c) = c \cdot w$ which concatenates w onto c, or $a_{\text{drop}}(w, c) = c$, which forgets the word w. For optimal planning, the encoder should choose the next action according to the policy:

$$\pi_e(w, c) = \operatorname*{argmax}_{a \in A} Q_e(a, w, c, c)$$

$$Q_e(a, w, c, \hat{c}) = \mathop{\mathbb{E}}_{w'|c,w} \left[U(w', a(w, \hat{c})) + \gamma \max_{a' \in A} Q_e(a', w, c \cdot w, a(w, \hat{c})) \right], \tag{5.3}$$

where $0 \leq \gamma \leq 1$ is a future-discount parameter, $U(w, \hat{c})$ is the utility of a word in a context, c represents the context as it currently is, and $\hat{c}$ represents the planner's projection of how the context will look in the future after taking certain actions.

If we use the utility function $U(w, \hat{c}) = -h(w|\hat{c})$, then the policy π_e is an autoencoder. To see this, expand Equation 5.3 as follows:

$$
\begin{aligned}
Q_e(a, w, c, \hat{c}) &= \mathop{\mathbb{E}}_{w'|c,w}\left[U(w', a(w, \hat{c})) + \gamma \max_{a' \in A} Q_e(a', w', c \cdot w, a(w, \hat{c}))\right] \qquad (5.3)\\
&= \mathop{\mathbb{E}}_{w'|c,w}\left[-h(w'|a(w, \hat{c}))\right] + \gamma \mathop{\mathbb{E}}_{w'|c,w}\left[\max_{a' \in A} Q_e(a', w, c \cdot w, a(w, \hat{c}))\right]\\
&= -H(W'|a(w, \hat{c}) \rightarrow W'|c, w) + \gamma \mathop{\mathbb{E}}_{w'|c,w}\left[\max_{a' \in A} Q_e(a', w, c \cdot w, a(w, \hat{c}))\right]\\
&= -H(W'|a(w, \hat{c}) \rightarrow W'|c, w)\\
&+ \gamma \mathop{\mathbb{E}}_{w'|c,w}\left[\max_{a' \in A} -H(W''|a'(w', a(w, \hat{c})) \rightarrow W''|c, w, w')\right] + ...
\end{aligned}
$$

Thus maximizing Q_e corresponds to minimizing cross-entropy with expected future words, and the resulting sequence has minimal cross entropy from the sequence it is meant to approximate.

A decoder that aims to approximate $\mathbf{w}$ should follow the following policy to generate words:

$$
\begin{aligned}
\pi_d(c) &= \operatorname*{argmax}_{w \in W} Q_d(w, c)\\
Q_d(w, c) &= \mathop{\mathbb{E}}_{c'|c,w}\left[U(w, c') + \gamma \max_{w' \in W} Q_d(w', c')\right],
\end{aligned}
$$

that is, it chooses the next word such that the resulting context representation will maximize some utility. This utility might reflect the speaker's intended meaning, for example.

The Q function for e included an expectation over $w'|c, w$ (that is, over the next word given the current word and context), and the Q function for d included an expectation over $c'|c, w$ (that is, the next context given the current word and context). How are these expectations calculated? One option is that the encoder e could use a decoder d to sample $w'|c, w$, and d could use an encoder e to sample $c'|c, w$. This recursion implies that produc-

ing language requires interleaving plans for simulated producers and comprehenders, and is reminiscent of Rational Speech Acts models (Frank and Goodman, 2012). Such interleaving of plans is dealt with in a Q-function framework in Kleiman-Weiner et al. (2016). I leave it for future work to determine the details of this.

5.4.4 Information locality from planning

I will show that distributions over sequences that have information locality are easier to autoencode according to the encoding function e as defined above. Incremental sequence samplers favor sequence distributions characterized by information locality under two different conditions. The first condition is when there is future-discounting in the planning algorithm (i.e., $\gamma < 1$). The second condition is when there is storage cost associated with keeping information in memory. These considerations provide an alternative motivation for information locality as a constraint on languages.

Supposing the set of encoding actions is $A = \{a_{\text{keep}}, a_{\text{drop}}\}$, let us consider the expected utility for a_{drop} on the next word w'. It is:

$$\begin{aligned}
\mathop{\mathbb{E}}_{w'|c,w}\left[-h(w'|a_{\text{drop}}(w,\hat{c}))\right] &= -H(W'|c \to W'|c,w) \\
&= -H(W'|c,w) - D_{\text{KL}}(W'|c \to W'|c,w) \\
&= -H(W'|c,w) - \mathop{\mathbb{E}}_{w'\sim W'|c,w}\left[\text{pmi}(w;w'|c)\right].
\end{aligned}$$

Expanding the recursive function Q_e, and dropping the entropy terms which are irrelevant for the optimization problem, we see the utility of dropping w is upper bounded by:

$$-\mathop{\mathbb{E}}_{w'\sim W'|c,w}\left[\text{pmi}(w;w'|c) - \gamma \mathop{\mathbb{E}}_{w''\sim W''|c,w,w'}\left[\text{pmi}(w;w''|c) - ...\right]\right],$$

where the upper bound comes from assuming we select a_{keep} for all future actions. That is, the importance of keeping w in memory is its expected pmi with future words, decreasing in importance according to the future discount. If there is some additional cost to keeping w in memory, and keeping w in memory will only pay off far in the future, then this system based on future-discounted reward might incorrectly drop w. On the other hand, if w will

W_1	W_2	W_3
a	c	a
a	d	a
b	c	b
b	d	b

Table 5.2: Language L used to demonstrate that information locality arises from planning with memory storage costs. The words W_1 and W_2 have long-range dependence; the word W_2 is noise.

pay off soon—if the words that it has high mutual information with are close—then the system is less likely to make a mistake.

A similar result can be derived with $\gamma = 1$ simply by assuming storage cost for w that continues for the whole time w is kept represented in the context representation c. If it is necessary to store w for a long time and this long-term storage is costly, then the planner might decide to drop it because the payoff down the line is not large enough. When sequences have information locality, this is less of a problem.

I will use an example to demonstrate the derivation of information locality from planning with memory storage cost. I will show that when there is storage cost, a planning autoencoder might choose not to remember critical contexts, which results in it not modelling the probability distribution over strings correctly. Consider a probabilistic language L defined in Table 5.2, where each string has uniform probability. The language consists of fixed-length strings of three words, where the first and third words are correlated, and the intervening second word is noise. Suppose that the set of possible encoding actions is $A = \{a_{\text{keep}}, a_{\text{drop}}\}$, and that we are currently viewing the first word W_1 and deciding whether to keep it or drop it as part of the context representation. Given word w_1 with a representation of previous context c, and setting $\gamma = 1$ so there is no future discount, the Q-function we want to maximize is:

$$Q(a_1, w_1, c) = \mathop{\mathbb{E}}_{w_2|w_1,c}\left[U(w_2, a_1(w_1, c)) + \max_{a_2} Q(a_2, w_2, c \cdot w_1, a_1(w_1, c))\right].$$

Let us assume a utility function $U(w, c) = -h(w|c) - C(c)$, where $C(c)$ is the storage cost for a context representation. For example, the storage cost function could be the length in

words of a context represented as a string of words. Then the Q-function comes out to:

$$Q(a_1, w_1, c) = \mathop{\mathbb{E}}_{w_2|w_1,c}\left[-h(w_2|a_1(w_1,c)) - C(a_1(w_1,c)) + \max_{a_2} Q(a_2, w_2, c\cdot w_1, a_1(w_1,c))\right]$$

$$\begin{aligned} &= -\mathop{\mathbb{E}}_{w_2|w_1,c}[h(w_2|a_1(w_1,c))] - C(a_1(w_1,c)) \\ &\quad + \mathop{\mathbb{E}}_{w_2|w_1,c}\left[\max_{a_2} \mathop{\mathbb{E}}_{w_3|c,w_1,w_2}[-h(w_3|a_2(w_2,a_1(w_1,c)))] - C(a_2(w_2,a_1(w_1,c)))\right]. \end{aligned} \tag{5.4}$$

Now there is no utility in remembering W_2, because it is noise uncorrelated with any other word, but there may be cost associated with it. So we can assume that the maximization inside Equation 5.4 always selects a_{drop}. Thus $a_2(w_2, a_1(w_1,c)) = a_1(w_1,c)$, so we can write:

$$\begin{aligned} Q(a_1, w_1, c) = &-\mathop{\mathbb{E}}_{w_2|w_1,c}[h(w_2|a_1(w_1,c))] - C(a_1(w_1,c)) \\ &+ \mathop{\mathbb{E}}_{w_3|w_1,c}[-h(w_3|a_1(w_1,c))] - C(a_1(w_1,c)) \end{aligned}$$

$$\begin{aligned} &= -H(W_2|a_1(w_1,c) \to W_2|w_1,c) - C(a_1(w_1,c)) \\ &\quad - H(W_3|a_1(w_1,c) \to W_3|w_1) - C(a_1(w_1,c)) \end{aligned}$$

$$= -H(W_2|a_1(w_1,c) \to W_2|w_1,c) - H(W_3|a_1(w_1,c) \to W_3|w_1) - 2C(a_1(w_1,c)).$$

Now let's consider the relative advantage of a_{keep} over a_{drop}. We will also assume the initial context c is the empty string ϵ, representing the fact that there is no relevant context to consider before the first word. Thus c is not informative about any word. We consider the difference in Q values for the two actions:

$$\begin{aligned} Q(a_{\text{keep}}, w_1) - Q(a_{\text{drop}}, w_1) = &-H(W_2|w_1 \to W_2|w_1) - 2C(c\cdot w_1) - H(W_3|w_1 \to W_3|w_1) \\ &+ H(W_2 \to W_2|w_1) + 2C(c) + H(W_3 \to W_3|w_1) \end{aligned}$$

$$= -H(W_2|w_1) - 2C(c \cdot w_1) - H(W_3|w_1) + H(W_2 \to W_2|w_1) + 2C(c) + H(W_3 \to W_3|w_1)$$
$$= -H(W_2) - 2C(c \cdot w_1) - H(W_3|w_1) + H(W_2) + 2C(c) + H(W_3 \to W_3|w_1)$$
$$= -2C(c \cdot w_1) - H(W_3|w_1) + 2C(c) + H(W_3|w_1) + D_{\text{KL}}(W_3|\varnothing \to W_3|w_1)$$
$$= -2C(c \cdot w_1) + 2C(c) + D_{\text{KL}}(W_3 \to W_3|w_1)$$
$$= \underbrace{\mathop{\mathbb{E}}_{w_3|w_1}[\text{pmi}(w_1; w_3)]}_{\text{in favor of } a_{\text{keep}}} - \underbrace{2(C(c \cdot w_1) - C(c))}_{\text{in favor of } a_{\text{drop}}}.$$

If the cost differential for keeping w_1 in memory ($2(C(c \cdot w_1) - C(c))$) exceeds the reward $\mathbb{E}_{w_3|w_1}[\text{pmi}(w_1; w_3)]$, then the optimal planner will drop w_1 from memory, and will thus not accurately model the probability distribution over strings. Then given the context `ac`, it will predict `a` and `b` with equal probability, because it did not store a representation of the first word `a` in memory. Thus it will generate sequences that differ from the target distribution over sequences.

If we generalize the example, we can see that it shows how sequences with information locality are more easily encoded by these planning autoencoders. In the language of Table 5.2, one word of noise intervened between the critical words W_1 and W_3. Now consider if d words of uncorrelated noise intervene between W_1 and W_{d+1} which is a copy of W_1. Then the difference in utilities between storing W_1 (a_{keep}) and dropping W_1 (a_{drop}) is:

$$Q(a_{\text{keep}}, w_1) - Q(a_{\text{drop}}, w_1) = \mathop{\mathbb{E}}_{w_{d+1}|w_1}[\text{pmi}(w_1; w_{d+1})] - (d+1)(C(c \cdot w_1) - C(c)).$$

Thus as the distance between the relevant words W_1 and W_{d+1} grows, the optimal planning autoencoder is more likely to erroneously drop W_1 from its context representation, and thus not model the probability distribution over sequences accurately.

The example above shows how probabilistic languages with information locality can be more accurately represented by planning autoencoders. It shows how information locality in human languages can be considered to follow from planning constraints, inasmuch as we think the planning involved in human language processing (both comprehension and production) is done incrementally with constrained memory.

5.4.5 Connection to deep reinforcement learning

In recent years, deep neural networks have gotten very promising results when used in the framework of Q-functions (Mnih et al., 2016; Silver et al., 2016). In these setups, a neural network is trained to approximate the Q function based on simulated outcomes, and a planner uses these approximate values to select the best action. In fact, something very close to the planning framework I sketched above has been proposed in the neural network literature for modeling sequential data. Yu et al. (2017) is one example; they use concatenation for the encoding function, and the decoding function is approximated using deep Q-learning to implement an objective very similar to the autoencoder described above. Guo (2015) is similar. Inasmuch as information locality, apparently a major feature of natural language syntax, falls out of such a framework, the results here suggest that this planning framework may be very well suited for modelling natural language sequences, because the framework has a realistic inductive bias.

Advances in deep learning have also come from taking a game theoretic approach in order to let multiple neural networks train each other. For example, Goodfellow et al. (2014) introduce Generative Adversarial Networks, a setup for unsupervised learning where one neural network (the generator) tries to mimic data from some distribution, and another neural network (the discriminator) tries to discriminate real samples from samples from the generator. The interleaving-plans aspect of the autoencoder for sequences described here suggests that another kind of game theoretic approach might be useful for language. We might imagine d and e from Section 5.4.3 as two neural networks that cooperate in order to successfully encode sequences.

Recent work has shown that neural network agents that use reinforcement learning can cooperate to develop a language with some natural language like properties (Mordatch and Abbeel, 2017). If these networks use the kinds of planning algorithms described above to produce and understand sequences, then I suggest they might develop syntax that looks very similar to that in natural languages.

5.5 Conclusion

I have argued that we can see natural languages as optimizing a utility function that is affected by information processing constraints. When these information processing constraints are constraints on memory in incremental processing, then we get out that natural languages should follow locality constraints: information locality and, as a special case, dependency locality.

Much empirical and theoretical work remains to be done to verify the ideas in this thesis. Nevertheless, the intuitions underlying them are straightforward and I believe they could have considerable explanatory potential.

This work should not be seen in conflict with the fields it is adjacent to. For example, I do not wish to present the utility function from Section 1.5 as an alternative to common models of language evolution; rather it is intended as a rough high-level description of the objective that these models are implicitly maximizing. Similarly, the noisy-context surprisal account of structural forgetting should not be taken as an alternative to neural network-based accounts (Frank et al., 2016), but rather as a highly generalized description of what neural networks are doing (prediction based on lossily compressed context). My hope is that the framework described here is a useful tool for reasoning about communication under information processing constraints, and that information locality can provide an external explanation for syntactic phenomena in these terms.

I also hope the ideas and empirical measures developed here give traction on understanding natural language in applied settings. Along these lines, Gulordava and Merlo (2016) have applied the measures of dependency length from Chapter 3 and word order freedom from Chapter 2 in order to study the behavior of dependency parsers. Also, the derivation of information locality effects from incremental comprehension (Section 1.5.5) and planning (Section 5.4.4) suggests that highly incremental models might have inductive biases which are useful for natural language tasks.

Appendices

Appendix A

Dependency Length under Different Dependency Annotation Schemes

The results in Chapter 3 show that dependency length as measured in dependency corpora is shorter than various random baselines. The question arises of whether these results might be dependent on the particular dependency annotation scheme used Most of the corpora used in that section originally come in a format where content words are heads of function words, for example in a prepositional phrase such as *in the house*, the word *in* is considered a dependent of the word *house*. This annotation style is called **content-head** dependencies. The results from Chapter 3 were calculated by automatically transforming the corpora to **function-head** format, where the word *in* is the head of *house* in the phrase *in the house*. For more details, see Section 3.2.1. The code for performing the translation from content-head to function-head dependencies is available online at `http://github.com/Futrell/cliqs`. For detailed discussion of the effects of content-head vs. function-head dependencies for crosslinguistic comparability, see Section 2.3.5.

In this appendix, I present the results from Section 3.2 comparing content-head and function-head dependencies. Here I show results based on the Universal Dependencies 2.0 corpora, which were not available when the work in Chapter 3 was originally conducted. This expands the number of languages to 50.

Figures A-1 through A-4 show results using the original content-head dependencies for the random projective baseline, the random fixed-order baseline, the random head-consistent baseline, and the random head-fixed baseline, respectively (see Section 3.2 for definitions). Figures A-5 through A-8 show results using automatically derived function-head dependencies. The choice of annotation style does not affect the overall pattern of results.

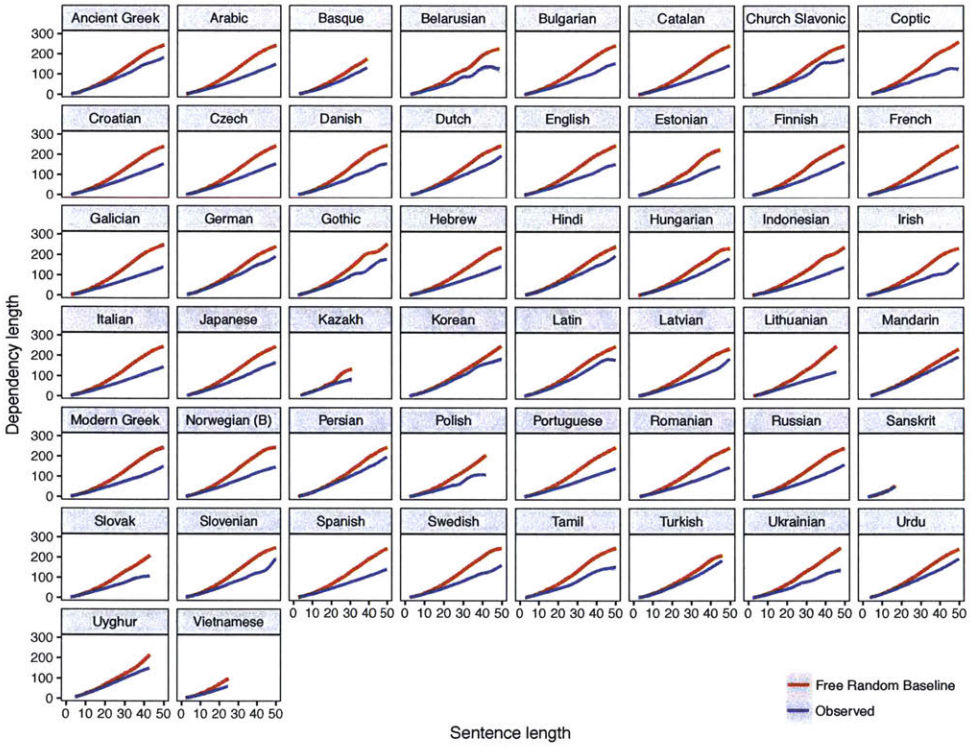


Figure A-1: Random vs. observed dependency lengths compared to the **free projective** baseline, with **content-head** dependencies.

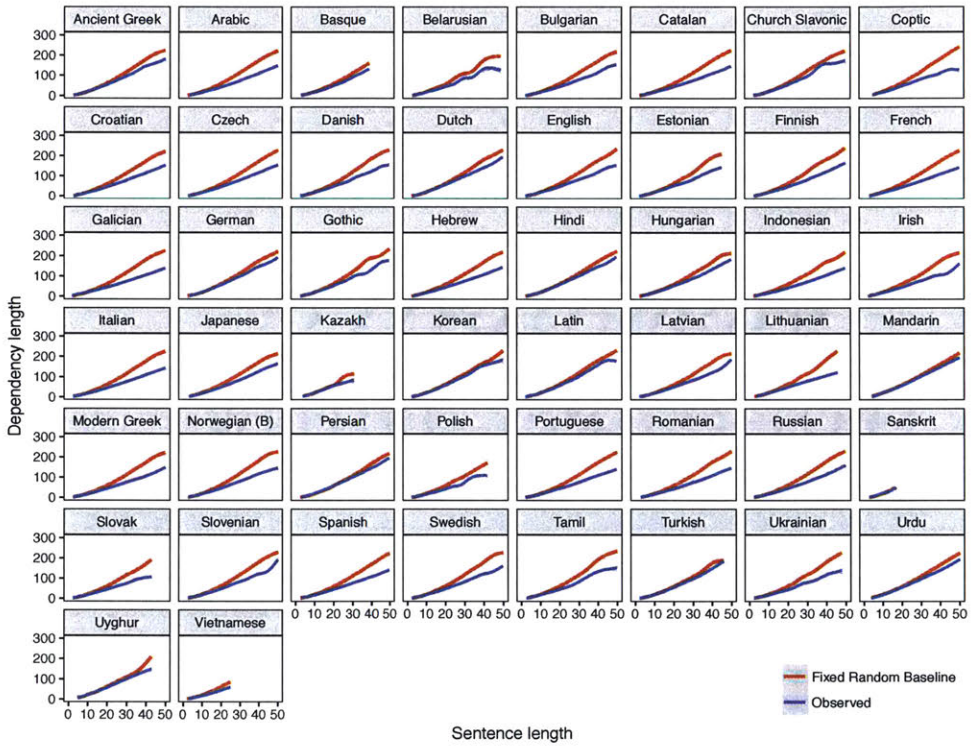


Figure A-2: Random vs. observed dependency lengths compared to the **fixed projective** baseline, with **content-head** dependencies.

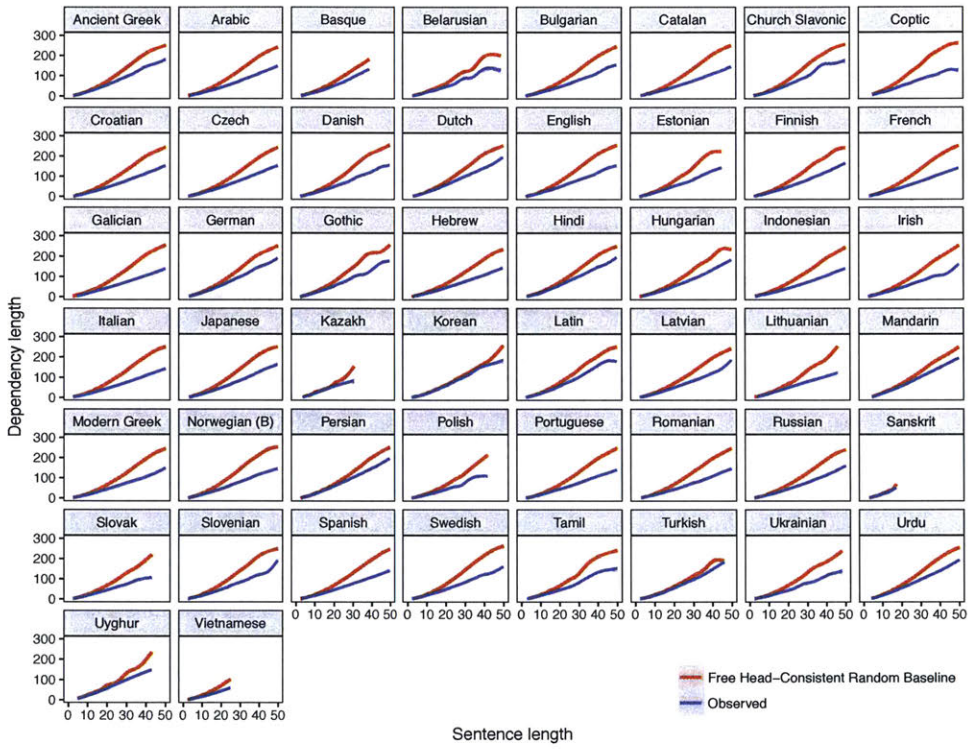


Figure A-3: Random vs. observed dependency lengths compared to the **free head-consistent projective** baseline, with **content-head** dependencies.

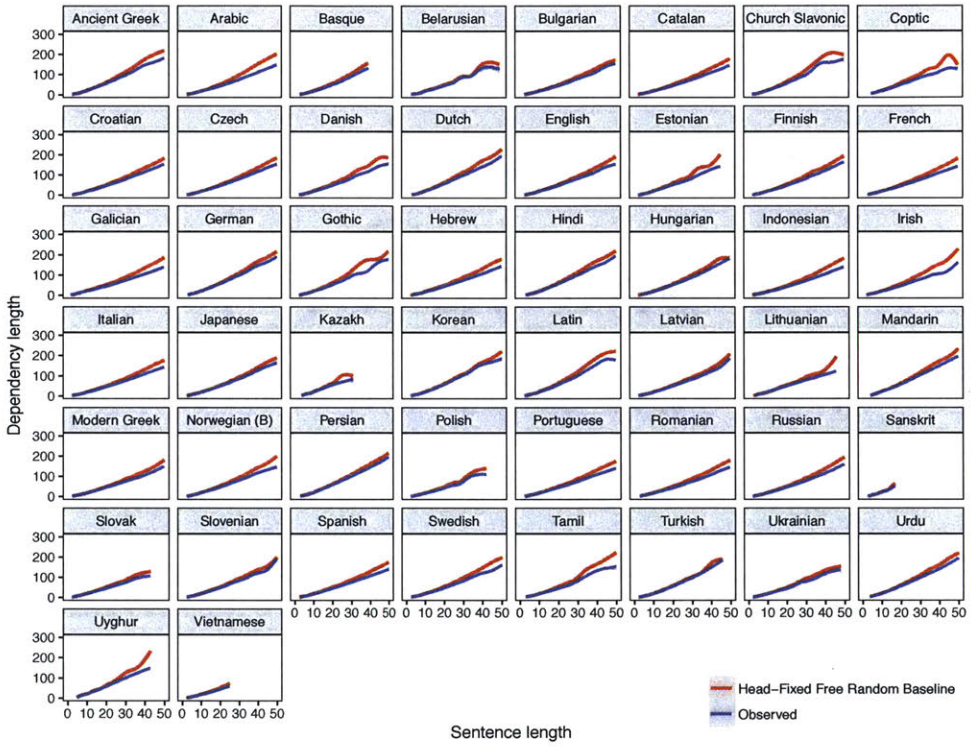


Figure A-4: Random vs. observed dependency lengths compared to the **free head-fixed projective** baseline, with **content-head** dependencies.

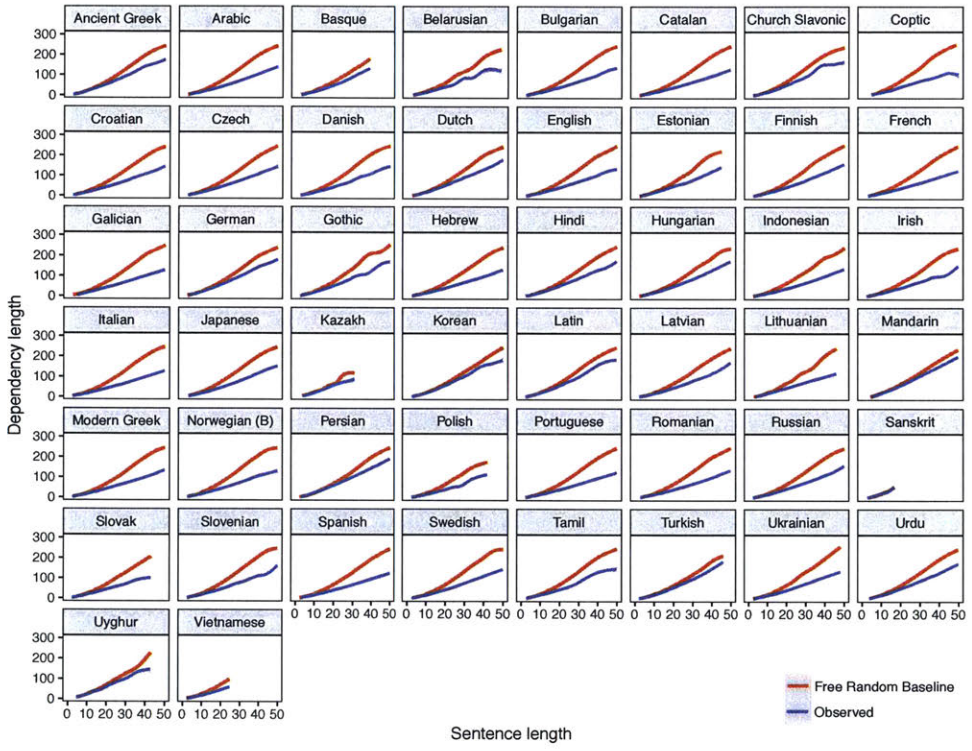


Figure A-5: Random vs. observed dependency lengths compared to the **free projective** baseline, with **function-head** dependencies.

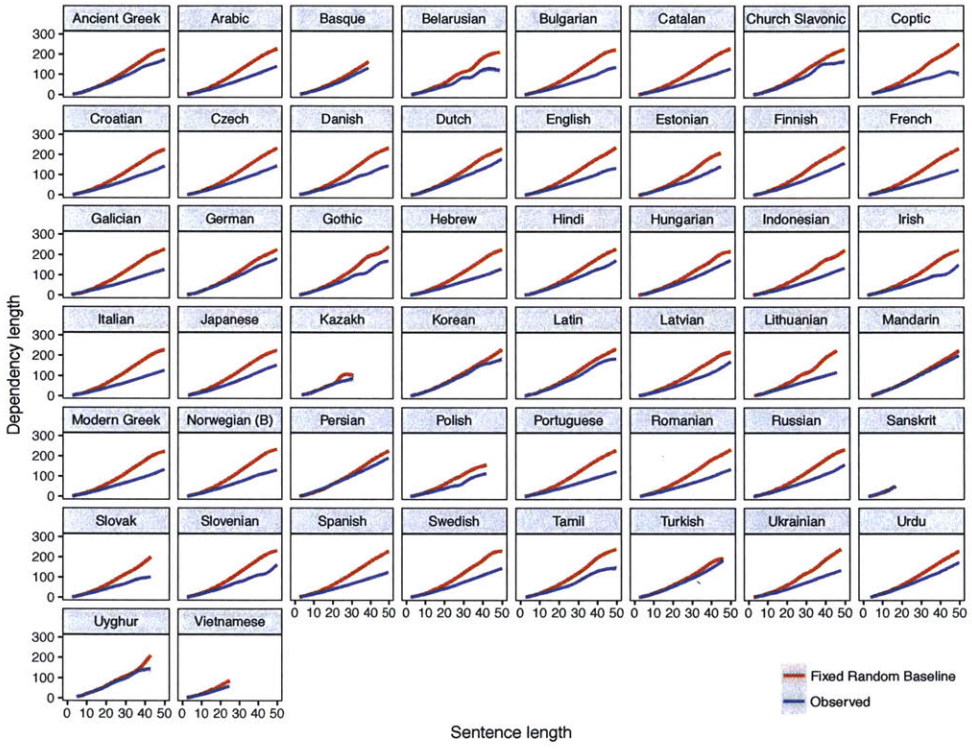


Figure A-6: Random vs. observed dependency lengths compared to the **fixed projective** baseline, with **function-head** dependencies.

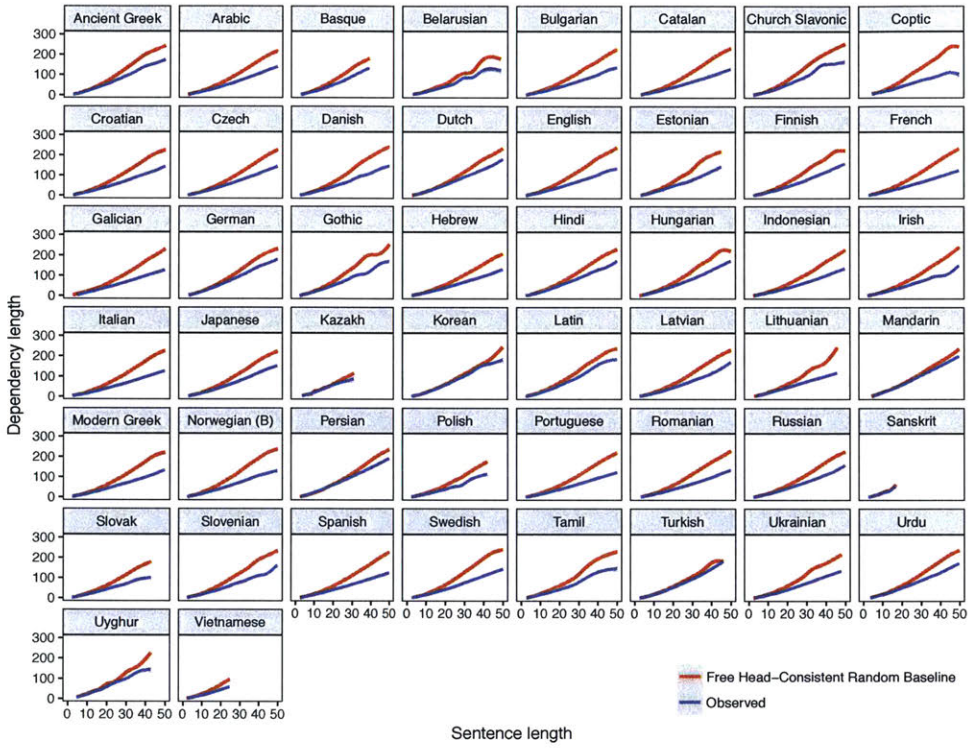


Figure A-7: Random vs. observed dependency lengths compared to the **free head-consistent projective** baseline, with **function-head** dependencies.

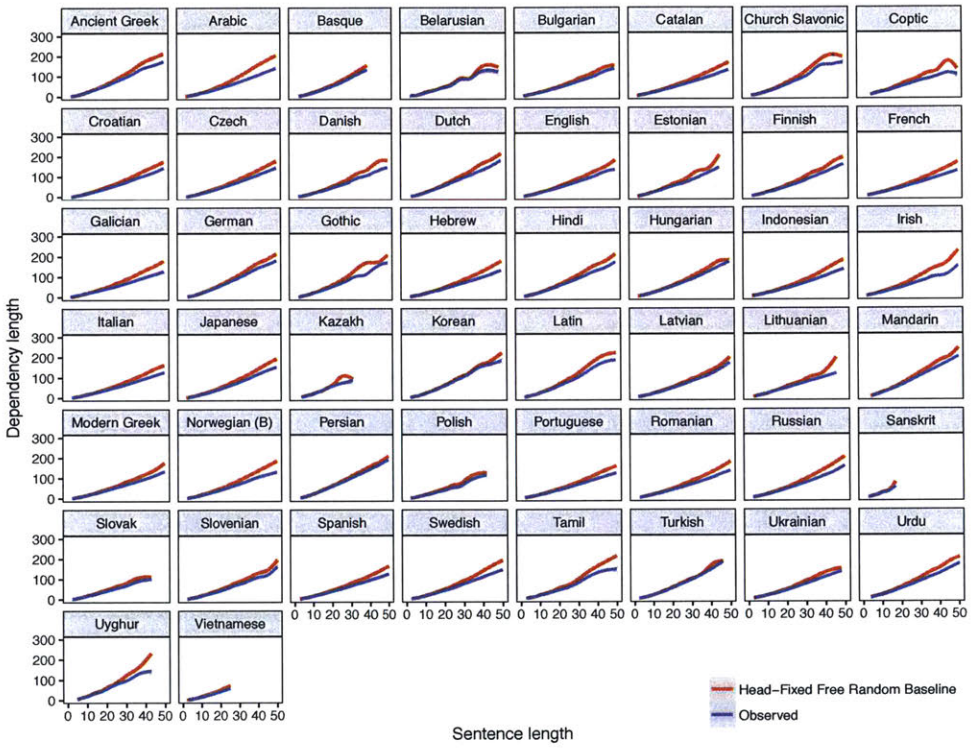


Figure A-8: Random vs. observed dependency lengths compared to the **free head-fixed projective** baseline, with **function-head** dependencies.

Bibliography

Abend, O. and Rappoport, A. (2010). Fully unsupervised core-adjunct argument classification. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 226–236. Association for Computational Linguistics.

Abney, S. P. (1987). *The English noun phrase in its sentential aspect*. PhD thesis, Massachusetts Institute of Technology.

Abney, S. P. and Johnson, M. (1991). Memory requirements and local ambiguities of parsing strategies. *Journal of Psycholinguistic Research*, 20(3):233–250.

Abramov, O. and Mehler, A. (2011). Automatic language classification by means of syntactic dependency networks. *Journal of Quantitative Linguistics*, 18(4):291–336.

Adger, D. (2003). *Core Syntax: A Minimalist Approach*. Oxford University Press, Oxford.

Agić, Ž., Aranzabe, M. J., Atutxa, A., Bosco, C., Choi, J., de Marneffe, M.-C., Dozat, T., Farkas, R., Foster, J., Ginter, F., Goenaga, I., Gojenola, K., Goldberg, Y., Hajič, J., Johannsen, A. T., Kanerva, J., Kuokkala, J., Laippala, V., Lenci, A., Lindén, K., Ljubešić, N., Lynn, T., Manning, C., Martínez, H. A., McDonald, R., Missilä, A., Montemagni, S., Nivre, J., Nurmi, H., Osenova, P., Petrov, S., Piitulainen, J., Plank, B., Prokopidis, P., Pyysalo, S., Seeker, W., Seraji, M., Silveira, N., Simi, M., Simov, K., Smith, A., Tsarfaty, R., Vincze, V., and Zeman, D. (2015). Universal dependencies 1.1. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

Aissen, J. (2003). Differential object marking: Iconicity vs. economy. *Natural Language & Linguistic Theory*, 21(3):435–483.

Aldezabal, I., Aranzabe, M., Gojenola, K., Sarasola, K., and Atutxa, A. (2002). Learning argument/adjunct distinction for Basque. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, volume 9, pages 42–50. Association for Computational Linguistics.

Anderson, J. R. and Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2(6):396.

Archer, E., Park, I. M., and Pillow, J. W. (2013). Bayesian and quasi-Bayesian estimators for mutual information from discrete data. *Entropy*, 15(5):1738–1755.

Archer, E., Park, I. M., and Pillow, J. W. (2014). Bayesian entropy estimation for countable discrete distributions. *Journal of Machine Learning Research*, 15:2833–2868.

Austin, P. and Bresnan, J. (1996). Non-configurationality in Australian aboriginal languages. *Natural Language and Linguistic Theory*, 14:215–268.

Baronchelli, A., Chater, N., Christiansen, M. H., and Pastor-Satorras, R. (2013). Evolution in a changing environment. *PLOS ONE*, 8(1):1–8.

Barzilay, R. and McKeown, K. R. (2005). Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Bates, D., Mächler, M., Bolker, B., and Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Bates, E. and MacWhinney, B. (1989). Functionalism and the competition model. In MacWhinney, B. and Bates, E., editors, *The Crosslinguistic Study of Sentence Processing*, pages 3–76. Cambridge University Press.

Behaghel, O. (1932). *Deutsche Syntax: Eine geschichtliche Darstellung. Band IV: Wortstellung*. Carl Winter, Heidelberg, Germany.

Bell, A. J. (2003). The co-information lattice. In *Proceedings of the Fifth International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 921–926.

Bell, D. A. (1953). The internal information of English words. In Jackson, W., editor, *Communication Theory*, pages 383–391. Academic Press, New York.

Belz, A., White, M., Espinosa, D., Kow, E., Hogan, D., and Stent, A. (2011). The first surface realisation shared task: Overview and evaluation results. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 217–226. Association for Computational Linguistics.

Bentz, C. and Alikaniotis, D. (2016). The word entropy of natural languages. *arXiv*.

Bentz, C., Alikaniotis, D., Samardžić, T., and Buttery, P. (2017). Variation in word frequency distributions: Definitions, measures and implications for a corpus-based language typology. *Journal of Quantitative Linguistics*, 24(2-3):128–162.

Bentz, C., Ruzsics, T., CorpusLab, U., Space, F., Koplenig, A., and Samardzic, T. (2016). A comparison between morphological complexity measures: typological data vs. language corpora. *CL4LC 2016*, page 142.

Bergen, B. K. (2004). The psychological reality of phonaesthemes. *Language*, 80(2).

Birner, B. and Ward, G. (2006). Information and structure. In Aarts, B. and McMahon, A., editors, *The Handbook of English Linguistics*, pages 291–317. Blackwell, Oxford.

Blahut, R. E. (1983). *Theory and Practice of Error-Control Codes*. Addison-Wesley, Reading, Massachusetts.

Bloomfield, L. (1933). *Language*. Henry Holt, New York.

Bock, J. K. (1982). Toward a cognitive psychology of syntax: Information processing contributions to sentence formulation. *Psychological Review*, 89:1–47.

Bock, J. K. and Warren, R. K. (1985). Conceptual accessibility and syntactic structure in sentence formulation. *Cognition*, 21:47–67.

Bohnet, B., Björkelund, A., Kuhn, J., Seeker, W., and Zarrieß, S. (2012). Generating non-projective word order in statistical linearization. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 928–939. Association for Computational Linguistics.

Bohnet, B., Mille, S., Favre, B., and Wanner, L. (2011). < StuMaBa >: From deep representation to surface. In *Proceedings of the 13th European workshop on natural language generation*, pages 232–235. Association for Computational Linguistics.

Braune, F., Bauer, D., and Knight, K. (2014). Mapping between English strings and reentrant semantic graphs. In *International Conference on Language Resources and Evaluation (LREC)*.

Bresnan, J. (1982). *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA.

Bresnan, J. (1997). The emergence of the unmarked pronoun: Chichewa pronominals in Optimality Theory. In *Proceedings of the 23rd Annual Meeting of the Berkeley Linguistics Society*, University of California, Berkeley. Berkeley Linguistics Society.

Bresnan, J., Kaplan, R., Peters, S., and Zaenen, A. (1982). Cross-serial dependencies in Dutch. *Linguistic Inquiry*, 13:613–635.

Brillouin, L. (1953). The negentropy principle of information. *Journal of Applied Physics*, 24:1152–1163.

Brillouin, L. (1956). *Science and Information Theory*. Academic Press.

Burton, N. G. and Licklider, J. C. R. (1955). Long-range constraints in the statistical structure of printed English. *American Journal of Psychology*, 68(4):650–653.

Bybee, J. (2010). *Language, Usage and Cognition*. Cambridge University Press, Cambridge.

Bybee, J. L. and Slobin, D. I. (1982). Why small children cannot change language on their own: Evidence from the English past tense. In Alqvist, A., editor, *Papers from the Fifth International Conference on Historical Linguistics*, pages 29–37. John Benjamins, Amsterdam.

Chang, F. (2009). Learning to order words: A connectionist model of heavy NP shift and accessibility effects in Japanese and English. *Journal of Memory and Language*, 61:374–397.

Chang, P.-C. and Toutanova, K. (2007). A discriminative syntactic word order model for machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, page 9.

Che, W., Li, Z., and Liu, T. (2012). *Chinese Dependency Treebank 1.0 LDC2012T05*. Linguistic Data Consortium, Philadelpha.

Chen-Main, J. and Joshi, A. K. (2010). Unavoidable ill-nestedness in natural language and the adequacy of tree local-MCTAG induced dependency structures. In Bangalore, S., Frank, R., and Romero, M., editors, *Proceedings of the 10th International Conference on Tree Adjoining Grammars and Related Formalisms (TAG+10)*, pages 53–60, Yale University.

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

Chomsky, N. (1988). *Language and Problems of Knowledge: The Managua Lectures*. MIT Press, Cambridge, MA.

Christiansen, M. H. and Chater, N. (2016). The Now-or-Never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, pages 1–19.

Chung, F. R. K. (1984). On optimal linear arrangements of trees. *Computers & Mathematics with Applications*, 10(1):43–60.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Clark, E. V. (1987). The principle of contrast: A constraint on language acquisition. In MacWhinney, B., editor, *Mechanisms of Language Acquisition*, pages 1–33. Lawrence Erlbaum Assoc., Hillsdale, NJ.

Clark, J. E., Yallop, C., and Fletcher, J. (2007). *Introduction to Phonetics and Phonology*. Blackwell, Oxford.

Collins, M. (2003). Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Comrie, B. (1981). *Language Universals and Linguistic Typology*. University of Chicago Press, Chicago, 1st edition.

Corbett, G. G. (2006). *Agreement*. Cambridge University Press.

Corbett, G. G., Fraser, N. M., and McGlashan, S., editors (1993). *Heads in Grammatical Theory*. Cambridge University Press, Cambridge.

Cornish, H., Dale, R., Kirby, S., and Christiansen, M. H. (2017). Sequence memory constraints give rise to language-like structure through iterated learning. *PLOS ONE*, 12(1):1–18.

Cover, T. and Thomas, J. (2006). *Elements of information theory*. John Wiley & Sons, Hoboken, NJ.

Croft, W. A. (2001). Functional approaches to grammar. In Smelser, N. J. and Baltes, P. B., editors, *International Encyclopedia of the Social and Behavioral Sciences*, pages 6323–6330. Elsevier Sciences, Oxford.

Croft, W. A. (2003). *Typology and universals*. Cambridge Textbooks in Linguistics. Cambridge University Press, Cambridge, 2nd edition.

Crooks, G. E. (2016). On measures of entropy and information. Tech. Note 009 v0.5. Available from http://threeplusone.com/info.

Cruise, B. (2014). Journey into information theory. Khan Academy video series. Online at `https://www.khanacademy.org/computing/computer-science/informationtheory`.

Culbertson, J., Smolensky, P., and Legendre, G. (2012). Learning biases predict a word order universal. *Cognition*, 122:306–329.

Dambacher, M., Kliegl, R., Hofmann, M., and Jacobs, A. M. (2006). Frequency and predictability effects on event-related potentials during reading. *Brain research*, 1084(1):89–103.

de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal stanford dependencies: A cross-linguistic typology. In Chair), N. C. C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavík, Iceland. European Language Resources Association (ELRA).

de Paiva Alves, E. (1996). The selection of the most probable dependency structure in Japanese using mutual information. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 372–374.

DeDeo, S., Hawkins, R. X. D., Klingenstein, S., and Hitchcock, T. (2013). Bootstrap methods for the empirical study of decision-making and information flows in social systems. *Entropy*, 15(6):2246–2276.

Dediu, D. and Cysouw, M. (2013). Some structural aspects of language are more stable than others: A comparison of seven methods. *PLOS ONE*, 8(1):1–20.

Demberg, V. and Keller, F. (2008). Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

Demberg, V. and Keller, F. (2009). A computational model of prediction in human parsing: Unifying locality and surprisal effects. In *Proceedings of the 31st Annual Meeting of the Cognitive Science Society*, Amsterdam, The Netherlands. Cognitive Science Society.

Demberg, V., Keller, F., and Koller, A. (2013). Incremental, predictive parsing with psycholinguistically motivated tree-adjoining grammar. *Computational Linguistics*, 39(4):1025–1066.

Dixon, R. (1982). *Where have all the adjectives gone? And other essays in semantics and syntax*. Mouton, Berlin, Germany.

Dryer, M. S. (1992). The Greenbergian word order correlations. *Language*, 68(1):81–138.

Dryer, M. S. (2002). Case distinctions, rich verb agreement, and word order type (Comments on Hawkins' paper). *Theoretical Linguistics*, 28(2):151–158.

Dryer, M. S. (2006). On Cinque on Greenberg's Universal 20.

Dryer, M. S. (2011). The branching direction theory of word order correlations revisited. In Scalise, S., Magni, E., and Bisetto, A., editors, *Universals of Language Today*. Springer, Berlin.

Dye, M., Milin, P., Futrell, R., and Ramscar, M. (in press). A functional theory of gender paradigms. In Kiefer, F., Blevins, J. P., and Bartos, H., editors, *Morphological Paradigms and Functions*. Brill, Leiden.

Eisner, J. M. (1996). Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 340–345.

Eisner, J. M. (1997). An empirical comparison of probability models for dependency grammar. Technical report, IRCS Report 96–11, University of Pennsylvania.

Eisner, J. M. and Smith, N. A. (2005). Parsing with soft and hard constraints on dependency length. In *Proceedings of the Ninth International Workshop on Parsing Technology*, pages 30–41. Association for Computational Linguistics.

Ellis, C. A. (1969). *Probabilistic Languages and Automata*. PhD thesis, University of Illinois, Urbana.

Fedzechkina, M., Jaeger, T. F., and Newport, E. L. (2012). Language learners restructure their input to facilitate efficient communication. *Proceedings of the National Academy of Sciences*, 109(44):17897–17902.

Ferreira, V. S. and Yoshita, H. (2003). Given-new ordering effects on the production of scrambled sentences in japanese. *Journal of psycholinguistic research*, 32(6):669–692.

Ferrer i Cancho, R. (2005). Zipf's law from a communicative phase transition. *The European Physical Journal B-Condensed Matter and Complex Systems*, 47(3):449–457.

Ferrer i Cancho, R. (2006). Why do syntactic links not cross? *Europhysics Letters*, 76(6):1228.

Ferrer i Cancho, R. (2015). The placement of the head that minimizes online memory: A complex systems approach. *Language Dynamics and Change*, 5(1):114–137.

Ferrer i Cancho, R. (2016). The meaning-frequency law in Zipfian optimization models of communication. *Glottometrics*, 35:28–37.

Ferrer i Cancho, R. (2017). The optimality of attaching unlinked labels to unlinked meanings. *Glottometrics*, 36:1–16.

Ferrer i Cancho, R. and Díaz-Guilera, A. (2007). The global minima of the communicative energy of natural communication systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(06):P06009.

Ferrer i Cancho, R. and Liu, H. (2014). The risks of mixing dependency lengths from sequences of different length. *Glottotheory*, 5(2):143–155.

Ferrer i Cancho, R. and Solé, R. (2003). Least effort and the origins of scaling in human language. *Proceedings of the National Academy of Sciences*, 100(3):788.

Filippova, K. and Strube, M. (2009). Tree linearization in English: Improving language model based approaches. In *Proceedings of NAACL-HLT (Short Papers)*, pages 225–228.

Frank, M. and Goodman, N. (2012). Quantifying pragmatic inference in language games. *Science*, 336.

Frank, S. L., Trompenaars, T., Lewis, R. L., and Vasishth, S. (2016). Cross-linguistic differences in processing double-embedded relative clauses: Working-memory constraints or language statistics? *Cognitive Science*, 40:554–578.

Frisson, S., Rayner, K., and Pickering, M. J. (2005). Effects of contextual predictability and transitional probability on eye movements during reading. *Learning, Memory*, 31(5):862–877.

Fromkin, V. (1971). The non-anomalous nature of anomalous utterances. *Language*, 47:27–52.

Fromkin, V. (1980). *Errors in linguistic performance: Slips of the tongue, ear, pen, and hand*. Academic Press, New York.

Futrell, R. (2010). German grammatical gender as a nominal protection device. Senior Thesis, Stanford University.

Futrell, R. and Gibson, E. (2015). Experiments with generative models for dependency tree linearization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1978–1983, Lisbon, Portugal. Association for Computational Linguistics.

Futrell, R., Hickey, T., Lee, A., Lim, E., Luchkina, E., and Gibson, E. (2015a). Cross-linguistic gestures reflect typological universals: A subject-initial, verb-final bias in speakers of diverse languages. *Cognition*, 136:215–221.

Futrell, R. and Levy, R. (2017). Noisy-context surprisal as a human sentence processing cost model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 688–698, Valencia, Spain.

Futrell, R., Mahowald, K., and Gibson, E. (2015b). Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341.

Futrell, R., Mahowald, K., and Gibson, E. (2015c). Quantifying word order freedom in dependency corpora. In *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 91–100, Uppsala, Sweden.

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

Garrett, M. F. (1975). The analysis of sentence production. In Bower, G. H., editor, *The psychology of learning and motivation*, volume 9. Academic Press, New York.

Garrett, M. F. (1980). Levels of processing in sentence production. In Butterworth, B., editor, *Language Production, Volume 1: Speech and talk*. Academic Press, London.

Gelling, D., Cohn, T., Blunsom, P., and Graça, J. (2012). The Pascal challenge on grammar induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 64–80.

Gelman, A. and Hill, J. (2007). *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, Cambridge, UK.

Gennary, S. P. and MacDonald, M. C. (2008). Semantic indeterminacy in object relative clauses. 58(4):161–187.

Gibson, E. (1991). *A computational theory of human linguistic processing: Memory limitations and processing breakdown*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Gibson, E. (1998). Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Gibson, E. (2000). The dependency locality theory: A distance-based theory of linguistic complexity. In Marantz, A., Miyashita, Y., and O'Neil, W., editors, *Image, language, brain: Papers from the first mind articulation project symposium*, pages 95–126.

Gibson, E., Piantadosi, S. T., Brink, K., Bergen, L., Lim, E., and Saxe, R. (2013). A noisy-channel account of crosslinguistic word-order variation. *Psychological science*, 24(7):1079–1088.

Gibson, E. and Thomas, J. (1999). Memory limitations and structural forgetting: The perception of complex ungrammatical sentences as grammatical. *Language and Cognitive Processes*, 14(3):225–248.

Gildea, D. and Jaeger, T. F. (2015). Human languages order information efficiently. *arXiv*, abs/1510.02823.

Gildea, D. and Temperley, D. (2007). Optimizing grammars for minimum dependency length. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 184–191, Prague, Czech Republic.

Gildea, D. and Temperley, D. (2010). Do grammars minimize dependency length? *Cognitive Science*, 34(2):286–310.

Givón, T. (1991). Markedness in grammar: distributional, communicative and cognitive correlates of syntactic structure. *Stud Lang*, 15:335–370.

Givón, T. (1992). On interpreting text-distributional correlations. Some methodological issues. In Payne, D. L., editor, *Pragmatics of word order flexibility*, page 305–320. John Benjamins Publishing Co, Amsterdam and Philadelphia.

Goldberg, Y. and Orwant, J. (2013). A dataset of syntactic-ngrams over time from a very large corpus of English books. In *Second Joint Conference on Lexical and Computational Semantics*, volume 1, pages 241–247.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. *arXiv*, abs/1406.2661.

Goodman, N. D. and Stuhlmüller, A. (2013). Knowledge and implicature: Modeling language understanding as social cognition. *Topics in Cognitive Science*, 5(1):173–184.

Gordon, P., Hendrick, R., and Johnson, M. (2001). Memory interference during language processing. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 27(6):1411–1423.

Gordon, P. C., Hendrick, R., Johnson, M., and Lee, Y. (2006). Similarity-based interference during language comprehension: Evidence from eye tracking during reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(6):1304–1321.

Grandvalet, Y. and Bengio, Y. (2005). Semi-supervised learning by entropy minimization. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 529–536. MIT Press.

Gray, R. M. (1990). *Entropy and Information Theory*. Springer-Verlag, New York.

Greenberg, J. H. (1963). Some universals of grammar with particular reference to the order of meaningful elements. In Greenberg, J. H., editor, *Universals of Language*, pages 73–113. MIT Press, Cambridge, MA.

Greenberg, J. H. (1966). *Language universals, with special reference to feature hierarchies.* Mouton, The Hague, The Netherlands.

Grice, H. (1975). Logic and conversation. In Cole, P. and Morgan, J. L., editors, *Syntax and Semantics, Vol. 3, Speech Acts*, pages 41–58. Academic Press, New York.

Griffiths, T. L. and Kalish, M. L. (2007). Language evolution by iterated learning with Bayesian agents. *Cognitive Science*, 31:441–480.

Grodner, D. and Gibson, E. (2005). Consequences of the serial nature of linguistic input for sentential complexity. *Cognitive Science*, 29(2):261–290.

Gulordava, K. and Merlo, P. (2015a). Diachronic trends in word order freedom and dependency length in dependency-annotated corpora of Latin and Ancient Greek. *Proceedings of the Third International Conference on Dependency Linguistics (Depling 2015)*, pages 121–130.

Gulordava, K. and Merlo, P. (2015b). Structural and lexical factors in adjective placement in complex noun phrases across romance languages. In *CoNLL*, pages 247–257.

Gulordava, K. and Merlo, P. (2016). Multi-lingual dependency parsing evaluation: a large-scale analysis of word order properties using artificial data. *Transactions of the Association for Computational Linguistics*, 4:343–356.

Guo, H. (2015). Generating text with deep reinforcement learning. *arXiv*, abs/1510.09202.

Hale, J. (2001). A probabilistic Earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics and Language Technologies*, pages 1–8.

Hale, K. (1983). Warlpiri and the grammar of non-configurational languages. *Natural Language and Linguistic Theory*, 1:5–47.

Halliday, M. A. K. (1967). Notes on transitivity and theme in English. *Journal of Linguistics*, 3:37–81.

Harper, L. H. (1964). Optimal assignments of numbers to vertices. *Journal of the Society for Industrial Applied Mathematics*, 12:131–135.

Haspelmath, M. (2008). Parametric versus functional explanations of syntactic universals. In Biberauer, T., editor, *The limits of syntactic variation*, pages 75–107. Benjamins.

Haspelmath, M. (2010). Framework-free grammatical theory. In Heine, B. and Narrog, H., editors, *The Oxford Handbook of Linguistic Analysis*, pages 341–365. Oxford University Press, 2nd edition.

Haspelmath, M. (2011). The indeterminacy of word segmentation and the nature of morphology and syntax. *Folia Linguistica*, 45(1):31–80.

Hawkins, J. A. (1990). A parsing theory of word order universals. *Linguistic Inquiry*, 21(2):223–261.

Hawkins, J. A. (1994). *A performance theory of order and constituency.* Cambridge University Press, Cambridge.

Hawkins, J. A. (1998). Some issues in a performance theory of word order. In Siewierska, A., editor, *Constituent Order in the Languages of Europe*, pages 729–81. Mouton de Gruyter, Berlin.

Hawkins, J. A. (2004). *Efficiency and complexity in grammars.* Oxford University Press, Oxford.

Hawkins, J. A. (2014). *Cross-linguistic Variation and Efficiency.* Oxford University Press, Oxford.

Hays, D. G. (1964). Dependency theory: A formalism and some observations. *Language*, 40:511–525.

He, W., Wang, H., Guo, Y., and Liu, T. (2009). Dependency based Chinese sentence realization. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 809–816. Association for Computational Linguistics.

Hetzron, R. (1978). On the relative order of adjectives. In Seller, H., editor, *Language Universals*. Narr, Tübingen, Germany.

Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4(1):11–26.

Hockett, C. F. (1959). Animal 'languages' and human language. *Human Biology*, 31(1):32–39.

Hockett, C. F. (1960). The origin of language. *Scientific American*, 203(3):88–96.

Hopcroft, J. E. and Ullman, J. D. (1979). *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley.

Hopper, P. and Traugott, E. C. (2003). *Grammaticalization.* Cambridge University Press, Cambridge, UK, 2nd edition.

Hudson, R. A. (1990a). *English Word Grammar.* Blackwell.

Hudson, R. A. (1990b). *Word Grammar.* Blackwell.

Husain, S., Vasishth, S., and Srinivasan, N. (2015). *Journal of Eye Movement Research*, 8(2):1–12.

Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 53:188–196.

Iordanskii, M. A. (1974). Minimal numbering of the vertices of trees. *Soviet Mathematics, Doklady*, 15:1311–1315.

Jackendoff, R. (2002). *Foundations of Language: Brain, Meaning, Grammar, Evolution.* Oxford University Press.

Jaeger, T. F. (2006). *Redundancy and syntactic reduction in spontaneous speech.* PhD thesis, Stanford University.

Jaeger, T. F. and Tily, H. J. (2011). On language 'utility': Processing complexity and communicative efficiency. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(3):323–335.

Jakulin, A. and Bratko, I. (2003). Quantifying and visualizing attribute interactions. *arXiv*, abs/cs/0308002.

Johnson, M. (2007). Transforming projective bilexical dependency grammars into efficiently-parsable cfgs with unfold-fold. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 168–175, Prague, Czech Republic.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F. B., Wattenberg, M., Corrado, G., Hughes, M., and Dean, J. (2016). Google's multilingual neural machine translation system: Enabling zero-shot translation. *arXiv*, abs/1611.04558.

Joshi, A. K. (1990). Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes*, 5:1–27.

Joshi, A. K., Shanker, K. V., and Weir, D. (1991). The convergence of mildly context-sensitive grammar formalisms. In Sells, P., Shieber, S., and Wasow, T., editors, *Foundational Issues in Natural Language Processing*, pages 31–81. MIT Press, Cambridge, MA.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *arXiv*, 1602.02410.

Juba, B., Kalai, A. T., Khanna, S., and Sudan, M. (2011). Compression without a common prior: An information-theoretic justification for ambiguity in language. In *2nd Symposium on Innovations in Computer Science.*

Kamide, Y., Altmann, G., and Haywood, S. (2003). The time-course of prediction in incremental sentence processing: Evidence from anticipatory eye movements. *Journal of Memory and Language*, 49(1):133–156.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401.

Kayne, R. S. (1994). *The Antisymmetry of Syntax.* MIT Press, Cambridge, MA.

Kemp, C. and Regier, T. (2012). Kinship categories across languages reflect general communicative principles. *Science*, 336(6084):1049–1054.

Khetarpal, N., Neveu, G., Majid, A., Michael, L., and Regier, T. (2013). Spatial terms across languages support near-optimal communication: Evidence from Peruvian Amazonia, and computational analyses. In Knauff, M., Pauen, M., Sebanz, N., and Wachsmuth, I., editors, *Proceedings of the 35th annual meeting of the Cognitive Science Society*, pages 764–769, Austin, TX. Cognitive Science Society.

King, A. and Wedel, A. (2017). Redundancy and the lexicon: the effect of word informativity on word shape. Talk presented at the 30th Annual CUNY Conference on Human Sentence Processing, Cambridge, MA. Online at `http://tedlab.mit.edu/cuny_abstracts/383_Final_Manuscript.pdf`.

Kiparsky, P. (1997). The rise of positional licensing. In von Kemenade, A. and Vincent, N., editors, *Parameters of morphosyntactic change*, pages 460–494. Cambridge University Press.

Kirby, S. (1999). *Function, selection, and innateness: The emergence of language universals*. Oxford University Press, Oxford.

Kirby, S. (2002). Learning, bottlenecks and the evolution of recursive syntax. In Briscoe, E., editor, *Linguistic evolution through language acquisition: Formal and computational models*, pages 173–203. Cambridge University Press, Cambridge.

Kirby, S., Cornish, H., and Smith, K. (2008). Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences*, 105(31):10681–10686.

Kirby, S., Griffiths, T., and Smith, K. (2014). Iterated learning and the evolution of language. *Current Opinion in Neurobiology*, 28C:108–114.

Kirby, S., Tamariz, M., Cornish, H., and Smith, K. (2015). Compression and communication in the cultural evolution of linguistic structure. *Cognition*, 141:87–102.

Kleiman-Weiner, M., Ho, M. K., Austerweil, J. L., Littman, M. L., and Tenenbaum, J. B. (2016). Coordinate to cooperate or compete: Abstract goals and joint intentions in social interaction. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*.

Klein, D. and Manning, C. D. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, page 478.

Kliegl, R., Grabner, E., Rolfs, M., and Engbert, R. (2004). Length, frequency, and predictability effects of words on eye movements in reading. *European Journal of Cognitive Psychology*, 16:262–284.

Kornai, A. (2011). Probabilistic grammars and languages. *Journal of Logic, Language and Information*, 20(3):317–328.

Kuhlmann, M. (2013). Mildly non-projective dependency grammar. *Computational Linguistics*, 39(2):355–387.

Kuhlmann, M. and Nivre, J. (2006). Mildly non-projective dependency structures. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 507–514, Sydney, Australia. Association for Computational Linguistics.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, pages 79–86.

Kutas, M. and Hillyard, S. A. (1984). Brain potentials during reading reflect word expectancy and semantic association. *Nature*, 307:161–163.

Lafferty, J. D., Sleator, D., and Temperley, D. (1992). Grammatical trigrams: A probabilistic model of link grammar. In *Proceedings of the AAAI Conference on Probabilistic Approaches to Natural Language*.

Langacker, R. (1987). *Foundations of Cognitive Grammar*, volume 1. Stanford University Press, Stanford.

Langacker, R. (1991). *Foundations of Cognitive Grammar*, volume 2. Stanford University Press, Stanford.

Lashley, K. S. (1951). The problem of serial order in behavior. In Jeffress, L. A., editor, *Cerebral mechanisms in behavior*, pages 112–136. Wiley, Oxford.

Lehmann, W. P. (1973). A structural principle of language and its implications. *Language*, 49:47–66.

Levelt, W. J. M. (1982). Linearization in describing spatial networks. In Peters, S. and Saarinen, E., editors, *Processes, Beliefs, and Questions*, pages 199–220. Reidel, Dordrecht.

Levy, R. (2005). *Probabilistic Models of Word Order and Syntactic Discontinuity*. PhD thesis, Stanford University, Stanford, CA.

Levy, R. (2008a). Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Levy, R. (2008b). A noisy-channel model of rational human sentence comprehension under uncertain input. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 234–243.

Levy, R. (2011). Integrating surprisal and uncertain-input models in online sentence comprehension: formal techniques and empirical results. In *ACL*, pages 1055–1065.

Lewis, R. L. and Vasishth, S. (2005). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, 29(3):375–419.

Li, W. (1989). Mutual information functions of natural language texts. Technical report, Santa Fe Institute Working Paper #1989-10-008.

Lin, H. W. and Tegmark, M. (2016). Critical behavior from deep dynamics: A hidden dimension in natural language. *arXiv*, abs/1606.06737.

Liu, H. (2008). Dependency distance as a metric of language comprehension difficulty. *Journal of Cognitive Science*, 9(2):159–191.

Liu, H. (2010). Dependency direction as a means of word-order typology: A method based on dependency treebanks. *Lingua*, 120(6):1567–1578.

Liu, H. and Li, W. (2010). Language clusters based on linguistic complex networks. *Chinese Science Bulletin*, 55(30):3458–3465.

Liu, Y., Zhang, Y., Che, W., and Qin, B. (2015). Transition-based syntactic linearization. In *Proceedings of NAACL*. Association for Computational Linguistics.

Lu, Q., Xu, C., and Liu, H. (2016). Can chunking reduce syntactic complexity of natural languages? *Complexity*, 21(S2):33–41.

Luce, R. D. (1986). *Response times*. Oxford University Press, New York.

Luce, R. D. (2003). Whatever happened to information theory in psychology? *Review of General Psychology*, 7:183–188.

Lupyan, G. and Dale, R. (2010). Language structure is partly determined by social structure. *PLOS ONE*, 5(1):e8559.

MacDonald, M. C. (1999). Distributional information in language comprehension, production, and acquisition: three puzzles and a moral. In MacWhinney, B., editor, *The Emergence of Language*, pages 177–196. Lawrence Erlbaum Associates, Mahwah, NJ.

MacDonald, M. C. (2013). How language production shapes language form and comprehension. *Frontiers in Psychology*, 4:226.

MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge University Press, Cambridge, UK.

MacWilliams, F. J. and Sloane, N. J. A. (1981). *The Theory of Error Correcting Codes*. North-Holland, Amsterdam.

Mahowald, K., Fedorenko, E., Piantadosi, S. T., and Gibson, E. (2013). Info/information theory: speakers choose shorter words in predictive contexts. *Cognition*, 126:313–318.

Mandelbrot, B. (1953). An informational theory of the statistical structure of language. *Communication theory*, 84:486–502.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman & Company.

Marslen-Wilson, W. D. (1975). Sentence perception as an interactive parallel process. *Science*, 189:226–228.

Maurits, L., Navarro, D., and Perfors, A. (2010). Why are some word orders more common than others? A uniform information density account. In *Advances in Neural Information Processing Systems*, pages 1585–1593.

McCarthy, J. J. (2002). *A thematic guide to Optimality Theory*. Cambridge University Press, Cambridge.

McCloskey, J. (1993). Constraints on syntactic processes. In Jacobs, J., von Stechow, A., Sternefeld, W., and Vennemann, T., editors, *Syntax: An international handbook of contemporary research*, volume 1, pages 496–506. Mouton de Gruyter, Berlin.

McDonald, R., Nivre, J., Quirmbach-Brundage, Y., Goldberg, Y., Das, D., Ganchev, K., Hall, K., Petrov, S., Zhang, H., Täckström, O., Bedini, C., Bertomeu Castelló, N., and Lee, J. (2013). Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

McFadden, T. (2003). On morphological case and word-order freedom. In *Proceedings of the Berkeley Linguistics Society*.

McGill, W. J. (1955). Multivariate information transmission. *IEEE Transactions on Information Theory*, 4(4):93–111.

Mel'čuk, I. A. (1988). *Dependency syntax: Theory and practice*. SUNY Press.

Merkel, J. (1885). Die zeitlichen Verhältnisse der Willensthätigkeit. *Philosophische Studien*, 2:73–127.

Meyerhoff, M. (2006). *Introducing Sociolinguistics*. Routledge.

Miller, G. A. (1955). Note on the bias of information estimates. In *Information Theory in Psychology: Problems and Methods*, pages 95–100.

Mitchell, J., Lapata, M., Demberg, V., and Keller, F. (2010). Syntactic and semantic factors in processing difficulty: An integrated measure. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 196–206.

Mnih, V., Puigdomenech Badia, A., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asychronous methods for deep reinforcement learning. *arXiv*, abs/1602.01783.

Mordatch, I. and Abbeel, P. (2017). Emergence of grounded compositional language in multi-agent populations. *arXiv*, abs/1703.04908.

Neal, R. M. (1993). Probabilistic inference using Markov Chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto. Technical Report CRG-TR-93-1.

Newmeyer, F. J. (1998). *Language Form and Language Function*. MIT Press, Cambridge, MA.

Newmeyer, F. J. (2005). *Possible and Probable Language: A Generative Perspective on Linguistics*. Oxford University Press, Oxford.

Nichols, J. (1986). Head-marking and dependent-marking grammar. *Language*, 62(1):56–119.

Nivre, J. (2005). Dependency grammar and dependency parsing. Technical report, Växjö University.

Nivre, J. (2015). Towards a universal grammar for natural language processing. In *Computational Linguistics and Intelligent Text Processing*, pages 3–16. Springer.

Nivre, J., Agić, Ž., Ahrenberg, L., Aranzabe, M. J., Asahara, M., Atutxa, A., Ballesteros, M., Bauer, J., Bengoetxea, K., Berzak, Y., Bhat, R. A., Bick, E., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Cebiroğlu Eryiğit, G., Celano, G. G. A., Chalub, F., Çöltekin, Ç., Connor, M., Davidson, E., de Marneffe, M.-C., Diaz de Ilarraza, A., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eli, M., Erjavec, T., Farkas, R., Foster, J., Freitas, C., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Ginter, F., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., Gonzáles Saavedra, B., Grioni, M., Grūzītis, N., Guillaume, B., Hajič, J., Hà M, L., Haug, D., Hladká, B., Ion, R., Irimia, E., Johannsen, A., Jørgensen, F., Kaşıkara, H., Kanayama, H., Kanerva, J., Katz, B., Kenney, J., Kotsyba, N., Krek, S., Laippala, V., Lam, L., Lê Hng, P., Lenci, A., Ljubešić, N., Lyashevskaya, O., Lynn, T., Makazhanov, A., Manning, C., Mărănduc, C., Mareček, D., Martínez Alonso, H., Martins, A., Mašek, J., Matsumoto, Y., McDonald, R., Missilä, A., Mititelu, V., Miyao, Y., Montemagni, S., Mori, K. S., Mori, S., Moskalevskyi, B., Muischnek, K., Mustafina, N., Müürisep, K., Nguyn Th, L., Nguyn Th Minh, H., Nikolaev, V., Nurmi, H., Osenova, P., Östling, R., Øvrelid, L., Paiva, V., Pascual, E., Passarotti, M., Perez, C.-A., Petrov, S., Piitulainen, J., Plank, B., Popel, M., Pretkalniņa, L., Prokopidis, P., Puolakainen, T., Pyysalo, S., Rademaker, A., Ramasamy, L., Real, L., Rituma, L., Rosa, R., Saleh, S., Saulīte, B., Schuster, S., Seeker, W., Seraji, M., Shakurova, L., Shen, M., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Smith, A., Spadine, C., Suhr, A., Sulubacak, U., Szántó, Z., Tanaka, T., Tsarfaty, R., Tyers, F., Uematsu, S., Uria, L., van Noord, G., Varga, V., Vincze, V., Wallin, L., Wang, J. X., Washington, J. N., Wirén, M., Žabokrtský, Z., Zeldes, A., Zeman, D., and Zhu, H. (2016). Universal dependencies 1.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University in Prague.

Nivre, J., Bosco, C., Choi, J., de Marneffe, M.-C., Dozat, T., Farkas, R., Foster, J., Ginter, F., Goldberg, Y., Hajič, J., Kanerva, J., Laippala, V., Lenci, A., Lynn, T., Manning, C.,

McDonald, R., Missilä, A., Montemagni, S., Petrov, S., Pyysalo, S., Silveira, N., Simi, M., Smith, A., Tsarfaty, R., Vincze, V., and Zeman, D. (2015). *Universal Dependencies 1.0*. Universal Dependencies Consortium.

Noji, H. and Miyao, Y. (2014). Left-corner transitions on dependency parsing. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 2140–2150, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Noji, H., Miyao, Y., and Johnson, M. (2016). Using left-corner parsing to encode universal structural constraints in grammar induction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 33–43, Austin, TX.

Nowak, M. A. and Krakauer, D. C. (1999). The evolution of language. *Proceedings of the National Academy of Sciences*, 96:8028–8033.

Nowak, M. A., Krakauer, D. C., and Dress, A. (1999). An error limit for the evolution of language. *Proceedings of the Royal Society B: Biological Sciences*, 266:2131–2136.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Park, Y. A. and Levy, R. (2009). Minimal-length linearizations for mildly context-sensitive dependency trees. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 335–343, Boulder, Colorado. Association for Computational Linguistics.

Pate, J. K. (2017). Optimization of American English, Spanish, and Mandarin Chinese over time for efficient communication.

Pavão, R., Savietto, J. P., Sato, J. R., Xavier, G. F., and Helene, A. F. (2016). On sequence learning models: Open-loop control not strictly guided by Hick's law. *Scientific Reports*, 6(23018).

Pereira, F. (2000). Formal grammar and information theory: Together again? *Philosophical Transactions of the Royal Society*, 358:1239–1253.

Piantadosi, S. T., Tily, H., and Gibson, E. (2011). Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529.

Piantadosi, S. T., Tily, H., and Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition*, 122(3):280–291.

Plotkin, J. B. and Nowak, M. A. (2001). Major transitions in language evolution. *Entropy*, 3:227–246.

Pollard, C. and Sag, I. (1987). *Information-based syntax and semantics.* Center for the Study of Language and Information, Stanford, CA.

Pollard, C. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar.* Center for the Study of Language and Information, Stanford, CA.

Polyanskiy, Y. and Wu, Y. (2016). Lecture notes on information theory. Lecture notes from MIT course 6.441, Spring 2016. Online at `https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-441-information-theory-spring-2016/lecture-notes/MIT6_441S16_course_notes.pdf`.

Popel, M., Mareček, D., Štěpánek, J., Zeman, D., and Žabokrtský, Z. (2013). Coordination structures in dependency treebanks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 517–527, Sofia, Bulgaria.

Prüfer, H. (1918). Neuer Beweis eines Satzes über Permutationen. *Archiv der Mathematischen Physik*, 27:742 – 744.

Qian, T. and Jaeger, T. F. (2012). Cue effectiveness in communicatively efficient discourse production. *Cognitive Science*, 36:1312–1336.

Radford, A. (1997). *Syntactic Theory and the Structure of English.* Cambridge University Press, Cambridge, UK.

Rajkumar, R., van Schijndel, M., White, M., and Schuler, W. (2016). Investigating locality effects and surprisal in written English syntactic choice phenomena. *Cognition*, 155:204–232.

Regier, T., Carstensen, A., and Kemp, C. (2016). Languages support efficient communication about the environment: Words for snow revisited. *PLOS ONE*, 11(4):e0151138.

Regier, T., Kay, P., and Khetarpal, N. (2007). Color naming reflects optimal partitions of color space. *Proceedings of the National Academy of Sciences*, 104:1436–1441.

Regier, T., Kemp, C., and Kay, P. (2015). Word meanings across languages support efficient communication. In *The Handbook of Language Emergence*, pages 237–263. Wiley-Blackwell, Hoboken, NJ.

Resnik, P. (1992). Left-corner parsing and psychological plausibility. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 191–197. Association for Computational Linguistics.

Richie, R. (2016). Functionalism in the lexicon. *The Mental Lexicon*, 11(3):429.

Rijkhoff, J. (1990). Explaining word order in the noun phrase. *Linguistics*, 28(1):5–42.

Roland, D., Dick, F., and Elman, J. L. (2007). Frequency of basic English grammatical structures: A corpus analysis. *Journal of Memory and Language*, 57(3):348–379.

Ros, I., Santesteban, M., Fukumora, K., and Laka, I. (2015). Aiming at shorter dependencies: The role of agreement morphology. *Language, Cognition and Neuroscience*, 30(9):1156.

Rosa, R., Mašek, J., Mareček, D., Popel, M., Zeman, D., and Žabokrtský, Z. (2014). HamleDT 2.0: Thirty dependency treebanks Stanfordized. In Chair), N. C. C., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Rosenbaum, D. A., Cohen, R. G., Jax, S. A., Weiss, D. J., and van der Wel, R. (2007). The problem of serial order in behavior: Lashley's legacy. *Hum. Mov. Sci*, 26:525–554.

Salge, C., Ay, N., Polani, D., and Prokopenko, M. (2015). Zipf's law: Balancing signal usage cost and communication efficiency. *PLOS ONE*, 10(10):1–14.

Sankoff, D. (1971). Branching processes with terminal types: Applications to context-free grammars. *Journal of Applied Probability*, 8:233–240.

Sapir, E. (1921). *Language, an introduction to the study of speech*. Harcourt, Brace and Co., New York.

Saussure, F. d. (1916). *Course in general linguistics*. Open Court Publishing Company.

Schneider, D. W. and Anderson, J. R. (2011). A memory-based model of Hick's law. *Cognitive Psychology*, 62(3):193–222.

Scontras, G., Degen, J., and Goodman, N. D. (2017). Subjectivity predicts adjective ordering preferences. *Open Mind: Discoveries in Cognitive Science*, 1(1):53–65.

Shain, C., van Schijndel, M., Futrell, R., Gibson, E., and Schuler, W. (2016). Memory access during incremental sentence processing causes reading time latency. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 49–58, Osaka, Japan.

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:623–656.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

Sleator, D. and Temperley, D. (1991). Parsing English with a link grammar. Technical report, Carnegie Mellon University Computer Science technical report CMU-CS-91-196.

Slobin, D. I. (1973). Cognitive prerequisites for the development of grammar. In Slobin, D. I. and Ferguson, C. A., editors, *Studies of Child Language Development*. Holf, Rinehart & Winston, New York.

Smith, E. E. (1968). Choice reaction time: An analysis of the major theoretical positions. *Psychological Bulletin*, 69:77–110.

Smith, N. A. and Eisner, J. M. (2006). Annealing structural bias in multilingual weighted grammar induction. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 569–576. Association for Computational Linguistics.

Smith, N. J. and Levy, R. (2013). The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Spike, M., Smith, K., and Kirby, S. (2016). Minimal pressures leading to duality of patterning. In Roberts, S., Cuskley, C., McCrohon, L., Barceló-Coblijn, L., Fehér, O., and Verhoef, T., editors, *The Evolution of Language: Proceedings of the 11th International Conference (EVOLANGX11)*. Online at `http://evolang.org/neworleans/papers/129.html`.

Sproat, R. and Shih, C. (1991). The cross-linguistic distribution of adjective ordering restrictions. In Georgopoulos, C. and Ishihara, R., editors, *Interdisciplinary approaches to language: Essays in honor of S.-Y. Kuroda*, pages 565–593. Kluwer Academic, Dordrecht, Netherlands.

Suppes, P. (1970). Probabilistic grammars for natural languages. *Synthese*, 22:95–116.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Tanaka, M. N., Branigan, H. P., McLean, J. F., and Pickering, M. J. (2011). Conceptual influences on word order and voice in sentence production: Evidence from Japanese. *Journal of Memory and Language*, 65:318–330.

Teichner, W. H. and Krebs, M. J. (1974). Laws of visual choice reaction time. *Psychological Review*, 81:75–98.

Temperley, D. (2005). The dependency structure of coordinate phrases: A corpus approach. *Journal of Psycholinguistic Research*, 34(6):577–601.

Temperley, D. (2007). Minimization of dependency length in written English. *Cognition*, 105(2):300–333.

Temperley, D. (2008). Dependency-length minimization in natural and artificial languages. *Journal of Quantitative Linguistics*, 15(3):256–282.

Tesnière, L. (1959). *Eléments de syntaxe structurale*. Librairie C. Klincksieck.

Tily, H. J. (2010). *The role of processing complexity in word order variation and change*. PhD thesis, Stanford University.

Tin, H. K. (1962). On the amount of information. *Theory Prob. Appl.*, 7(4):439–444.

Tomasello, M. (2003). *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press, Cambridge, MA.

Travis, L. (1989). Parameters of phrase structure. In Baltin, M. R. and Kroch, A. S., editors, *Alternative Conceptions of Phrase Structure*, pages 263–279. University of Chicago Press, Chicago.

Tria, F., Galantucci, B., and Loreto, V. (2012). Naming a structured world: A cultural route to duality of patterning. *PLOS ONE*, 7(6):1–8.

van Lint, J. H. (1999). *Introduction to Coding Theory*, volume 86 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 2nd edition.

Vasishth, S., Chopin, N., Ryder, R., and Nicenboim, B. (2017). Modelling dependency completion in sentence comprehension as a bayesian hierarchical mixture process: A case study involving Chinese relative clauses. *arXiv*, abs/1702.00564.

Vasishth, S., Suckow, K., Lewis, R. L., and Kern, S. (2010). Short-term forgetting in sentence comprehension: Crosslinguistic evidence from verb-final structures. *Language and Cognitive Processes*, 25(4):533–567.

Vennemann, T. (1974). Theoretical word order studies: Results and problems. *Papiere zur Linguistik*, 7:5–25.

Wasow, T. (2002). *Postverbal Behavior*. CSLI Publications, Stanford, CA.

Weir, D. J. (1988). *Characterizing mildly context-sensitive grammar formalisms*. PhD thesis, University of Pennsylvania, Philadelphia, PA.

Welford, A. T. (1960). The measurement of sensory-motor performance: Survey and reappraisal of twelve years' progress. *Ergonomics*, 3:189–230.

White, M. and Rajkumar, R. (2012). Minimal dependency length in realization ranking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 244–255. Association for Computational Linguistics.

Wolf, F. and Gibson, E. (2005). Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2):249–287.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J.

(2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv*, abs/1609.08144.

Xu, C. and Liu, H. (2015). Can familiarity lessen the effect of locality? A case study of Mandarin Chinese subjects and the following adverbials. *Poznań Studies in Contemporary Linguistics*, 51(3):463–485.

Xu, Y. and Regier, T. (2014). Numeral systems across languages support efficient communication: From approximate numerosity to recursion. In Bello, P., Guarini, M., McShane, M., and Scassellati, . B., editors, *Proceedings of the 36th annual meeting of the Cognitive Science Society*, pages 1802–1807, Austin, TX. Cognitive Science Society.

Xu, Y., Regier, T., and Malt, B. C. (2016). Historical semantic chaining and efficient communication: The case of container names. *Cognitive Science*, 40:2081–2094.

Yamashita, H. and Chang, F. (2001). "Long before short" preference in the production of a head-final language. *Cognition*, 81(2):B45–B55.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). SeqGAN: Sequence generative adversarial nets with policy gradient. In *The Thirty-First AAAI Conference on Artificial Intelligence (AAAI 2017)*.

Yuret, D. (1998). *Discovery of linguistic relations using lexical attraction*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA.

Zeman, D., Dušek, O., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtský, Z., and Hajič, J. (2014). HamleDT: Harmonized multi-language dependency treebank. *Language Resources and Evaluation*, 48(4):601–637.

Zeman, D., Marecek, D., Popel, M., Ramasamy, L., Stepánek, J., Zabokrtskỳ, Z., and Hajic, J. (2012). HamleDT: To parse or not to parse? In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 2735–2741.

Zhang, Y. (2013). Partial-tree linearization: Generalized word ordering for text synthesis. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2232–2238. AAAI Press.

Ziff, P. (1960). *Semantic analysis*. Cornell University Press, Ithaca, NY.

Zipf, G. K. (1945). The meaning-frequency relationship of words. *Journal of General Psychology*, 33:251–266.

Zipf, G. K. (1949). *Human behavior and the principle of least effort*. Addison-Wesley Press, Oxford, UK.