A Computational Cognitive Model of Syntactic Priming

David Reitter, Frank Keller, and Johanna D. Moore

School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, UK
Phone: +44-131-650-4407, Fax: +44-131-650-6626
Email: reitter@cmu.edu

Abstract

The psycholinguistic literature has identified two syntactic adaptation effects in language production: rapidly decaying short-term priming and long-lasting adaptation. To explain both effects, we present an ACT-R model of syntactic priming based on a wide-coverage, lexicalized syntactic theory that explains priming as facilitation of lexical access. In this model, two well-established ACT-R mechanisms, base-level learning and spreading activation, account for long-term adaptation and short-term priming, respectively. Our model simulates incremental language production and in a series of modeling studies we show that it accounts for (a) the inverse frequency interaction; (b) the absence of a decay in long-term priming; and (c) the cumulativity of long-term adaptation. The model also explains the lexical boost effect and the fact that it only applies to short-term priming. We also present corpus data that verifies a prediction of the model, i.e., that the lexical boost affects all lexical material, rather than just heads.

Keywords: syntactic priming, adaptation, cognitive architectures, ACT-R, categorial grammar, incrementality.

1. Introduction

The task of *language production* is often analyzed in terms of a processing chain which includes conceptualization, formulation, and articulation (Levelt, 1989). The conceptualization system selects concepts to express, and the formulation system decides how to express them. Formulation involves determining the lexical, syntactic, and semantic representations of the utterance. Syntax is vital for language production, as it determines the form of an utterance, which in turn is in a systematic relationship with the meaning of the utterance. In the present article, we focus on

syntactic priming in language production; syntactic priming refers to the fact that the syntactic form of an utterance varies with recent linguistic experience.

When producing an utterance, the language processor is faced with choices that affect the realization of the sentence to be produced. Typical choices include: Should the clause be formulated as passive or active? Should the verb phrase be realized as *give the man a book* (double object) or as *give a book to the man* (prepositional object)? Should the word order be *dropped off the children* or *dropped the children off*? These syntactic alternatives are very similar in terms of their meaning, but differ in their surface structure.

The factors influencing the choice between syntactic alternatives can be tracked experimentally (Bock, 1986). For instance, speakers that have a choice between producing the double object and the prepositional object construction (e.g., in a picture naming task) are more likely to choose the construction that they (or their interlocutors) have produced previously. The same holds for the use of passives. The general conclusion from such experiments is that syntactic choices are sensitive to *syntactic priming*: any decision in favor of a particular structure renders following decisions for the same or a related structure more likely. This article proposes a model of syntactic priming in human language production, which explains the syntactic priming effect and many of its known interactions as the result of a combination of well-validated principles of learning and memory retrieval. We are only concerned with syntactic (structural) priming in language production; lexical (or semantic) priming, priming in language comprehension, and priming at other levels of linguistic representations are outside the focus of this article.

Despite the large number of studies investigating and utilizing syntactic priming, the origin of such priming effects, as well as their temporal properties, are subject to debate. Some studies found that the priming effect disappeared after just a clause or a sentence (Levelt & Kelter, 1982; Branigan, Pickering, & Cleland, 1999; Wheeldon & Smith, 2003)—we will call this type of effect *short-term priming*. Other authors find priming effects that persist much longer (Hartsuiker & Kolk, 1998; Bock & Griffin, 2000; Branigan, Pickering, Stewart, & McLean, 2000)—we will call this effect *long-term priming*. Apart from their differing temporal properties, short-term and long-term priming effects also exhibit qualitative differences (to be discussed in detail in Section 2.1 below). This empirical duality raises the question about the cognitive substrate that underlies short- and long-term priming. Is there really only one priming effect, or are we dealing with two effects with distinct cognitive bases?

The cognitive model presented in this article contributes to answering this question. We focus on *syntactic choice* as it manifests itself in priming and present a model that is implemented using the principles and mechanisms of the ACT-R cognitive architecture (Anderson et al., 2004). ACT-R has the aim of explaining cognition through the interaction of a set of general cognitive components. This makes it possible to implement simulations that generate behavior in a transparent and modular way and make predictions across a wide range of domains, down to the level of experimentally observed reaction times. Our model of language production follows this approach, and relies in particular on a set of basic learning principles of ACT-R. We show how these principles, which are independently motivated by a wide range of experimental results, can be used to account for key syntactic priming effects reported in the psycholinguistic literature.

In this article, we show how ACT-R can be used to implement a simplified, yet plausible sentence production model. Our model generates grammatical English sentences given a semantic description, and it does so in line with empirical results on priming in human language production. The model is not intended to cover all or even many aspects of syntax; rather, we focus on those

syntactic constructions which have been used in priming experiments. However, the syntactic basis of the model is a linguistic theory that covers a wide range of syntactic phenomena, viz., Combinatory Categorial Grammar (Steedman, 1999). The model presented in this article is a revised and extended version of the model initially proposed by Reitter (2008).

The remainder of this article is structured as follows. Section 2 gives background on syntactic priming, surveys models of priming, and introduces Combinatory Categorial Grammar and ACT-R. Section 3 presents our model of syntactic priming, explaining the components of ACT-R along the way. We motivate our evaluation methods in Section 4 and present four simulations that explain known priming effects and a corpus experiment that tests a prediction arising from our model. In Section 5, we compare our approach to existing models of syntactic priming, as well as summarizing our main contributions.

2. Background: Priming and language production

If syntactic choices are repeated more than we would expect by chance (i.e., based on the frequencies of the relevant structures alone), then this effect is referred to as syntactic priming. It has been shown that speakers are not only sensitive to priming from their own speech, they also accept priming from their interlocutors in a dialogue (Branigan, Pickering, & Cleland, 2000). This is what Pickering and Garrod (2004) call the *alignment* of linguistic representations in dialogue.

Over the past two decades, a broad range of syntactic priming effects have been demonstrated experimentally, and a number of computational models been developed to account for them. We will review these in the present section, as well as introducing background on Combinatory Categorial Grammar and ACT-R that is pertinent to our own model of priming.

2.1. Evidence of syntactic priming

Syntactic priming has been demonstrated for a range of syntactic constructions in language production. A much-cited study by Bock (1986) showed priming effects that were clearly syntactic in nature. In her experiments, participants were asked to repeat prime sentences, and then to describe semantically unrelated pictures, which served as targets. Primes consisted of sentences with ditransitive verbs, whose dative argument could either be realized as a prepositional object (PO) or as a double object (DO) construction, for instance, *a rock climber sold some cocaine to an undercover agent*, vs. *a rock climber sold an undercover agent some cocaine*. The results show that participants were more likely to use a DO target after a DO prime, and a PO target after a PO prime.

In general, experimental studies on syntactic priming have used a small number of well-known alternations whose variants are assumed to be synonymous. Examples include:

- double vs. prepositional objects, as described above (Bock, 1986; Branigan, Pickering, & Cleland, 2000);
- active vs. passive voice, e.g., *the prince told an anecdote* (active) vs. *an anecdote was told by the prince* (passive) (Weiner & Labov, 1983; Bock, 1986);
- noun phrases with modifiers, e.g., *the red sheep* (adjectival) vs. *the sheep that's red* (relative) (Cleland & Pickering, 2003);
- optional *that* complementizers and relativizers in English, e.g., *I thought [that] you were gone* (V. Ferreira, 2003);
- high vs. low relative clause attachment in German, e.g., *Gabi bestaunte das Titelbild$_{neutr}$ der Illustrierten$_{fem}$, das$_{neutr}$/die$_{fem}$ …* (*Gabi admired the cover$_{neutr}$ of the magazine$_{fem}$,*

which$_{neutr}$/which$_{fem}$ . . .) (Scheepers, 2003).

Syntactic priming effects have mostly been demonstrated in carefully controlled psycholinguistic experiments, but priming phenomena are also attested in naturally occurring text or speech. In an early corpus study, Estival (1985) finds priming effects for actives and passives. Gries (2005b) uses a corpus to show not only syntactic priming effects, but also that verbs differ in their sensitivity to priming. Szmrecsanyi (2005) presents a study demonstrating long-term priming for a range of syntactic alternations in a dialogue corpus. Linguistic decisions in specific syntactic constructions have been explained by priming, such as parallelism in coordinate constructions (Dubey, Keller, & Sturt, 2008), and DO/PO and active/passive alternations (Snider, 2008).

Rather than focusing on particular syntactic alternations, Reitter, Moore, and Keller (2006b) use corpora to show that priming can be explained as an effect of the repeated use of phrase structure rules. In this setting, constructions such as passive voice translate to particular sets of phrase structure rules. Reitter, Moore, and Keller (2006b) use regression models to confirm that the probability of any rule occurring is strongly elevated immediately after a previous occurrence of the same rule. Such results show that priming is a general syntactic phenomenon, rather than being limited to specific syntactic alternations.

The size of the priming effects found in these studies depends on a range of factors such as the syntactic alternation or grammar rule used, the experimental task, and the modality (speech or text). In addition, a range of factors have been identified in the literature as interacting with priming:

- cumulativity: the presence of multiple primes enhances priming;
- inverse frequency interaction: less frequent constructions prime more;
- lexical boost: more priming occurs if prime and target share lexical material;
- decay: priming can decay if material intervenes between prime and target.

An adequate model of syntactic priming in production has to be able to capture all of these properties of priming; we will discuss each of them in turn.

The *cumulativity of priming* has been demonstrated by Jaeger and Snider (2008), who report a corpus study which shows that the strength of the priming effect increases with the number of primes that precede it in the corpus. Investigating *that* omission, Jaeger and Snider's (2008) data indicate that the likelihood of producing a *that* complementizer or relativizer increases with the number of *that* complementizers or relativizers used previously. The authors also present a similar result for the production of passive constructions. Cumulativity has not been investigated directly in psycholinguistic experiments, but Kaschak, Loney, and Borregine (2006) report a study that tests the effect of multiple exposure to prime constructions. They show that DO/PO priming is reduced if exposure is imbalanced between DO and PO instances; this can be seen as evidence for cumulativity: balanced (i.e., frequent) exposure leads to more priming compared to imbalanced (i.e., less frequent) exposure. Also, Hartsuiker and Westenberg (2000) provided indirect evidence for cumulativity in word order priming: the two word orders they compared differed in their pre-experimental baselines frequencies; this difference was diminished after the experiment, suggesting that long term cumulative priming had occurred.

Priming has also been found to show an *inverse frequency interaction:* less frequent syntactic decisions prime more than more frequent ones. This was first noted experimentally by Scheepers (2003) for relative clause attachment priming, and has recently been confirmed by Snider and Jaeger (2009) for the DO/PO alternation. Corpus studies are consistent with the experimental findings. Jaeger and Snider (2008) show that less frequent constructions trigger larger priming effects for *that* omission and active/passive constructions. The inverse frequency effect has also been reported for

priming of arbitrary syntactic rules by Reitter (2008).

The experimental record indicates that syntactic priming is affected by lexical repetition. If the prime and the target share open-class words, a stronger syntactic priming effect is found (compared to a condition where there is no lexical repetition between prime and target). This *lexical boost effect* has been demonstrated in many experiments, in which the head word was repeated between primes and targets in one condition. For instance, Pickering and Branigan (1998) demonstrate that syntactic priming effects are stronger when the same verb is used in the prime and the target, using prepositional object and double object constructions in written sentence completion. Gries (2005b) finds the boost in a corpus-based study, Cleland and Pickering (2003) find it for noun phrases (repeating the head noun) and Schoonbaert, Hartsuiker, and Pickering (2007) for second-language speakers of English. It could be that this lexical boost effect is simply an epiphenomenon resulting from the syntactic preferences of verbs. Different verbs allow different subcategorization frames, which in turn can differ in frequency. If a verb is repeated, so is the subcategorization preference of that verb, hence the resulting stronger priming effects could simply be the additive effect of such lexical-syntactic preferences. However, there is evidence against this hypothesis from studies that demonstrate lexical boost effects for constructions that do not involve verbs (Cleland & Pickering, 2003; Szmrecsanyi, 2005). More generally, recent experimental and corpus results suggest that the lexical boost effect is not restricted to head repetition (Raffray & Scheepers, 2009; Snider, 2008, 2009); we will return to this issue in Section 4.5.

The lexical boost is short-lived: The strength of priming is unaffected by head verb repetition when there is intervening linguistic material. Hartsuiker, Bernolet, Schoonbaert, Speybroeck, and Vanderelst (2008) elicited prime-target pairs at varying lags, manipulating whether verbs in the prime and target sentences were repeated. They found a lexical boost only in sentences that were adjacent, but not when two or six sentences intervened. In a series of studies, Kaschak and colleagues examined long-term priming effects and found no lexical boost, i.e., no enhanced syntactic repetition if the verb was repeated (Kaschak et al., 2006; Kaschak, 2007; Kaschak & Borregine, 2008).

A number of experimental studies have investigated *decay in syntactic priming,* but the results do not readily provide a coherent picture. Some studies suggest that the syntactic bias introduced by priming decays quickly. In Levelt and Kelter's (1982) early study on priming in spontaneous, spoken language production, the effect disappeared after one clause. In later studies involving written sentence production, syntactic priming also ceased to be detectable when just one sentence intervened between prime and target (Branigan et al., 1999; Wheeldon & Smith, 2003). Reitter (2008) found strong decay effects for syntactic priming in spoken language corpora, which occurred in the first seconds after a syntactic decision. Other studies contrast strongly with this. Hartsuiker and Kolk (1998) found no decay of priming in spoken language production when a one-second temporal lag was inserted between prime and target. In a spoken picture description task, Bock and Griffin (2000) and Bock, Dell, Chang, and Onishi (2007) demonstrated a form of syntactic priming that persists with two and even ten intervening sentences. These results were corroborated by Branigan, Pickering, Stewart, and McLean (2000), who found that priming in spoken production persists, whether or not there is a temporal lag or intervening linguistic material that delays the elicitation of the target.

Hartsuiker et al. (2008) were able to resolve this apparent contradiction: they found that the lexical boost effect decays quickly, i.e., an increase in priming with lexical repetition is only observable if there is no lag between the prime and the target. The priming effect as such, however, is long-lived, and persists across intervening trials, independent of modality (written or spoken).

The studies in the literature that reported rapidly decaying priming effects used lexical repetition, while the studies that reported no decay did not use lexical repetition, consistent with Hartsuiker et al.'s (2008) findings. Hartsuiker et al. (2008) propose that therefore two mechanisms are required to explain syntactic priming: short-term priming is lexically driven and relies on an activation-based mechanism; long-term priming is independent of lexical material and uses an implicit learning mechanism. The idea that priming is due to implicit learning has also been proposed by Bock and Griffin (2000) and Bock et al. (2007), and underlies Chang, Dell, and Bock's (2006) model of syntactic priming. The model we propose in this article follows Hartsuiker et al.'s (2008) idea and postulates two mechanisms, based on spreading activation and implicit learning, respectively.

There are also a number of qualitative differences between short-term (rapidly decaying) and long-term (long lasting) priming that can be observed in corpus data. *(a)* While long-term priming correlates with task success in task-oriented dialogue, short-term priming does not. This was shown in a study (Reitter & Moore, 2007) on syntactically annotated speech from dyads that were engaged in a task requiring communication and cooperation. *(b)* Another difference is that short-term priming seems to affect only constituents, but not sequences of syntactic categories that cross constituent boundaries. *(c)* Long-term priming, however, has been found to be insensitive to constituent structure, i.e., it shows sequence priming (Reitter & Keller, 2007). *(d)* Finally, short-term priming is stronger in task-oriented dialogue than in spontaneous conversation (Reitter, Moore, & Keller, 2006b). Taken together, the temporal and qualitative differences between short- and long-term priming suggest that these two forms of priming have distinct cognitive bases. We will argue that these two effects should be explained by separate mechanisms in ACT-R, an issue we will return to in Section 2.4 below.

2.2. *Models of syntactic priming*

In the following, we provide an overview of the most important models that have been developed to capture structural priming. To enable meaningful comparison with our own model, we will concentrate on models that are implemented as computational simulations (rather than being theoretical accounts or statistical analyses). We only deal with models that are specifically designed to capture syntactic priming; general language production models will be discussed in Section 5.1, to the extent that they are relevant for modeling syntactic priming.

Chang et al. (2006) present a connectionist model that accounts for certain aspects of syntactic priming. Their Dual Path Model is primarily concerned with language acquisition, but it also offers an explanation for syntactic priming using the same mechanism. The model incorporates two processing pathways: a meaning system, which encodes event semantic representations of words, and a sequencing system, which determines the order in which words are produced. The model is implemented as a Simple Recurrent Network and trained using error-driven learning. It successfully explains a number of language acquisition and priming phenomena, including that priming can be insensitive to decay, that comprehension priming is similar to production priming, that priming is sensitive to the meaning of the primed structure in some cases, but is not influenced by function morphemes. However, the model is unable to account for lexical boost effects (Chang et al., 2006, p. 263). It is also unable to capture the interaction between lexical repetition and decay reported by Hartsuiker et al. (2008), and more generally, the Dual Path Model has no way of explaining the qualitative differences between short-term priming and long-term priming (as the model is designed

explicitly to unify acquisition and priming).[1] Another key limitation of the Dual Path Model is that it can only simulate comprehension-to-production priming, but not production-to-production priming. It requires an external input in order to generate an error signal for its error-driven learning mechanism. Furthermore, the Dual Path Model is trained on artificial language data, from which it learns transitions of abstractions of words, similar to part-of-speech categories. It therefore does not directly represent hierarchical relations between words and phrases. It is therefore unclear whether it can explain the sensitivity of priming to constituent structure (Reitter & Keller, 2007).

Another connectionist model of syntactic priming has recently been presented by Malhotra (2009). This model is conceptually similar to the Dual Path Model, but achieves a broader coverage of the empirical domain, and also dispenses with certain key assumptions, viz., that priming is a consequence of error-driven learning and that priming and language acquisition share the same underlying mechanism. Malhotra takes a dynamic systems approach: the model consists of a network of nodes which are connected by inhibitory and excitatory connections; as a function of the input activation, the network can settle on a number of stable states. Its behavior can then be analyzed using differential equations, and Malhotra demonstrates that the model can capture standard priming effects, including the lexical boost, the inverse frequency interaction, and cumulative priming. Malhotra's model shares with the Dual Path Model a principled limitation in that it does not explicitly represent syntactic structure; rather, it selects the syntactic structure of an utterance from a look-up table, which means that, at least in the current implementation, it is only able to deal with a limited number of pre-determined syntactic alternations. However, in contrast to the Dual Path Model, Malhotra's model incorporates an explicit account of short-term and long-term memory, and thus is able to distinguish short-term and long-term priming, and capture the fact that only the former shows a lexical boost effect. Malhotra (2009) reports experiments on comprehension-to-production priming only, and while he argues that his model is able to explain production-to-production priming in principle, designing simulations to show this does not seem to be straightforward (Malhotra, 2009, p. 296).

Snider (2008) presents an exemplar-based model of syntactic priming. His model draws on existing spreading activation accounts of lexical priming (Krott, Schreuder, & Baayen, 2002; Kapatsinski, 2006); the innovation is that he replaces the lexical representations in these models with syntactic representations derived from Data-oriented Parsing (DOP; Bod, 1992). DOP decomposes a syntax tree into all possible subtrees, which then form the exemplars over which spreading activations is computed in Snider's (2008) model. The resulting model makes three key predictions: less frequent exemplars prime more (the inverse frequency effect), exemplars that are more similar prime more (the lexical boost effect is a special case of this), and exemplars with more neighbors prime less (this is called the neighborhood density effect and is a standard finding for lexical priming). Snider (2008) reports a series of corpus studies involving the PO/DO and active/passive alternations that show that these predictions are borne out. While Snider's (2008) model provides an elegant account for frequency-based effects in priming (and should also be able to capture cumulativity in priming, though this is not explicitly modeled by Snider, 2008), it is unclear how the model would deal with decay in priming, and account for the difference between long-term and short-term priming (see Section 2.1), which Hartsuiker et al. (2008) hypothesizes have separate underlying mechanisms (Snider's model only has the spreading activation mechanism at its disposal). Furthermore, Snider's (2008) model is not currently implemented, making it difficult to assess its generality. In particular,

---

[1]However, under a more restricted view, the Dual Path Model could be regarded as an accurate explanation of syntax acquisition and long-term priming effects (leaving short-term priming to be explained independently).

the assumption that all subtrees of all syntactic structures ever produced by a speaker are stored is hard to reconcile with the need for efficient memory storage and retrieval. Furthermore, in contrast to Chang et al. (2006) and Malhotra (2009), Snider does not propose a learning mechanism for his model; it is unclear how association strengths are acquired. (Krott et al., 2002, propose a way of deriving association strengths for their lexical model, but it is non-trivial to generalize to DOP-style syntactic representations.)

2.3. A syntactic basis for incremental processing

An important choice in devising a model of language production is whether structure building follows a single analysis of a partial utterance or tracks multiple syntactic structures in parallel. Similarly, we need to determine if syntactic decisions for the whole utterance are made before producing the first word, or if syntactic construction proceeds incrementally. A language generator could work top-down, driven only by semantics. In that case, the last word of a sentence, or the last phrase of a long utterance, could be generated first, and it would have to be stored before it is uttered. An incremental generator, on the other hand, selects and adjoins every word to the current syntactic representation as it is produced, and very little buffering is necessary.

Various studies have examined the degree of incrementality in comprehension and production (see F. Ferreira and Swets, 2002, for a summary). To evaluate whether speaking begins before phonological planning is complete for the whole utterance, experimenters have manipulated the phonological complexity of words at the beginning and end of utterances. Wheeldon and Lahiri (1997) tested incrementality in production by manipulating the availability of information needed early and late when generating a sentence. Their participants were given a noun phrase and a question and had to answer as quickly as possible in a full sentence. Wheeldon and Lahiri found that participants began their sentences earlier when the first word was phonologically less complex. In further experiments, participants were asked to plan their sentences carefully. Then, sentence production latencies depended on the complexity of the entire utterance—not just on the first word. Wheeldon and Lahiri conclude that speakers start speaking whenever possible.

On the syntactic level, incrementality can be tested by manipulating the set of choices that a speaker needs to consider before beginning to decide on a sentence-initial word. For instance, F. Ferreira and Swets (2002) could provoke thorough planning and (some) incremental behavior in their participants, depending on how pressured the participants were to speak quickly. As an explanation for this contrast, they propose that incrementality should not be seen as "architectural". Instead, speakers strike a balance between speaking quickly and planning accurately; this balance can be adjusted to suit the circumstances. In this article, we will therefore assume that the human language production mechanism exhibits *flexible incrementality*, i.e., that the degree to which sentences are generated word-by-word is not hard-wired in the processor but can be adjusted depending on factors such as the task at hand and the resources available.

Operationalizing incrementality in production is a challenge for phrase-structure based models, as these accounts require planning processes that are computationally expensive and non-incremental. The grammar formalism therefore plays an important role in developing a model that generates natural language incrementally, such that speaking can start before the utterance production is complete. This motivates our choice of Combinatory Categorial Grammar (CCG; Steedman, 2000) for the model presented here. CCG is a lexicalized grammar formalism that has been developed based on cognitive and computational considerations and supports a large degree of syntactic incrementality.

An argument in favor of CCG is the fact that CCG allows both incremental and non-incremental realization; it assumes that several constituent structures can be valid representations of a single surface structure and its semantics. These structures may or may not be maintained by the processor in parallel. In CCG, partial analyses can be produced for the words on the left edge of an utterance with minimal representational overhead: a single category is sufficient to describe the combinatorial properties of an incrementally generated phrase.

A second argument in favor of CCG is that syntactic priming affecting complex syntactic structures (such as priming of subcategorization frames) can be explained in terms of priming of CCG categories, which are more complex than regular syntactic categories and incorporate subcategorization information (among other information). For example, Reitter, Hockenmaier, and Keller (2006) use regression models to show priming effects for CCG categories in corpora of transcribed and syntactically annotated dialogue. The study specifically demonstrates that lexical and non-lexical categories of the same CCG type prime each other, which is predicted under the CCG view of syntax, but not under a standard phrase structure view.

Syntactic types in CCG are more expressive and more numerous than standard parts of speech: there are around 500 highly frequent CCG types, as compared to the standard 50 or so Penn Treebank part-of-speech categories. CCG's compact, yet expressive encoding of syntactic categories allows it to interact well with key assumptions of ACT-R, such as the absence of a distinct notion of working memory (see Section 2.4 for details). Transient storage of information takes place in the interfaces (*buffers*) between different architectural components. CCG is compatible with this architectural property, since only a small amount of information about the current syntactic parse needs to be stored during sentence production. As long as the combinatorial process proceeds incrementally, the production algorithm is able to operate with a minimum of temporary storage to track the partial utterance produced so far.

In CCG, words are associated with lexical categories which specify their subcategorization behavior, e.g., $((S\backslash NP)/NP)/NP$ is the lexical category for ditransitive verbs in English such as *give* or *send*. These verbs expect two NPs (the objects) to their right, and one NP (the subject) to their left. Generally, complex categories X/Y or $X\backslash Y$ are constructs that lead to a constituent with category X if combined with a constituent of category Y to their right ($/Y$) or to their left ($\backslash Y$). Only a small number of combinatory processes are licensed in CCG, which can be described via rule schemata such as Forward Application:

(1)

Forward Application: $X/Y \ \ Y \quad \Rightarrow_> \ \ X$

Forward Application is the most basic operation (and used by all variants of categorial grammar). In the following example, multiple instances of forward application are used to derive the sentence category S, beginning with the category $(S\backslash NP)/NP$ (transitive verb):

$$
\begin{array}{ccc}
\textbf{I} & \textbf{saw} & \textbf{the man} \\
\hline
NP & (S\backslash NP)/NP & NP \\
\end{array}
$$

$$
\cfrac{\cfrac{NP \quad (S\backslash NP)/NP \quad NP}{S\backslash NP}{}^{>}}{S}{}^{<}
$$

CCG assumes a small number of other rule schemata, justified in detail by (Steedman, 2000):

(2)

Backward Application:	Y	X\Y	$\Rightarrow_>$ X
Forward Composition:	X/Y	Y/Z	$\Rightarrow_\mathbf{B}$ X/Z
Backward Composition:	Y\Z	X\Y	$\Rightarrow_\mathbf{B}$ X\Z
Backw. Crossed Composition:	Y/Z	X\Y	$\Rightarrow_\mathbf{B}$ X/Z
Forward Type-raising:	X		$\Rightarrow_\mathbf{T}$ T/(T\X)
Coordination:	X conj X		$\Rightarrow_\mathbf{\Phi}$ X

For our proposed model, we thus assume a categorial syntactic framework. Lemmas and syntactic categories are represented as declarative knowledge and we assume syntactic categories that encode information about the subcategorization frame of a given phrase as well as linearization information. The retrieval of lemmas is biased by prior use, which results in priming effects. The access and the syntactic combination of lexical and phrasal material is controlled by a small set of rules which form the syntactic core and encode universal, language-independent principles. These principles are implemented as instances of ACT-R production rules (of IF-THEN form). In the ACT-R architecture, the precondition (IF) portions of production rules are tested in parallel before one rule is selected for execution.

It is important to note that the use of CCG does not constitute a claim that other grammar formalisms, could not account for priming effects in connection with an ACT-R memory model. This applies in particular to lexicalized grammar formalisms such as Lexicalized Tree-adjoining Grammar (Joshi, Levy, & Takahashi, 1975) or Head-driven Phrase Structure Grammar (Pollard & Sag, 1994).

In common with other lexicalized grammar theories, CCG assumes that much syntactic information associated with single words is kept with each word in the lexicon. Subcategorization information is tied to words in the lexicon rather than inferred from syntactic configuration in sentences. Experimental results support this view. In a study by Melinger and Dobel (2005), participants were primed to use either DO or PO realizations (in German and Dutch) in a picture description task. Priming effects were obtained even though the primes consisted of just a single word: a semantically unrelated ditransitive verb which subcategorized either for the DO or the PO structure. This is compatible only with a lexically-based view of priming, where no exposure to actual syntactic structure is necessary to prime a subcategorization frame.

2.4. Modeling priming in ACT-R

ACT-R (Anderson et al., 2004) is a general cognitive architecture whose constraints are intended to be cognitively realistic and motivated by empirical results. It has been widely used to model experimental data qualitatively and quantitatively. Like other cognitive architectures, ACT-R specifies how information is encoded in memory, retrieved, and processed.

ACT-R defines three core elements (see Figure 1). *Buffers* hold temporary information about goals and the system's state. *Procedural memory* consists of IF-THEN *production rules* that generate requests, which then trigger memory retrieval. The result of a retrieval can then be tested by other production rules, conditional on information in buffers. *Declarative memory* is organized in *chunks*, which are lists of attribute-value pairs that bundle information. Chunks compete for activation, and this is where lexical and syntactic decision-making takes place in our model. A chunk's activation consists of two main components: base-level activation and spreading activation, which we will discuss in turn.
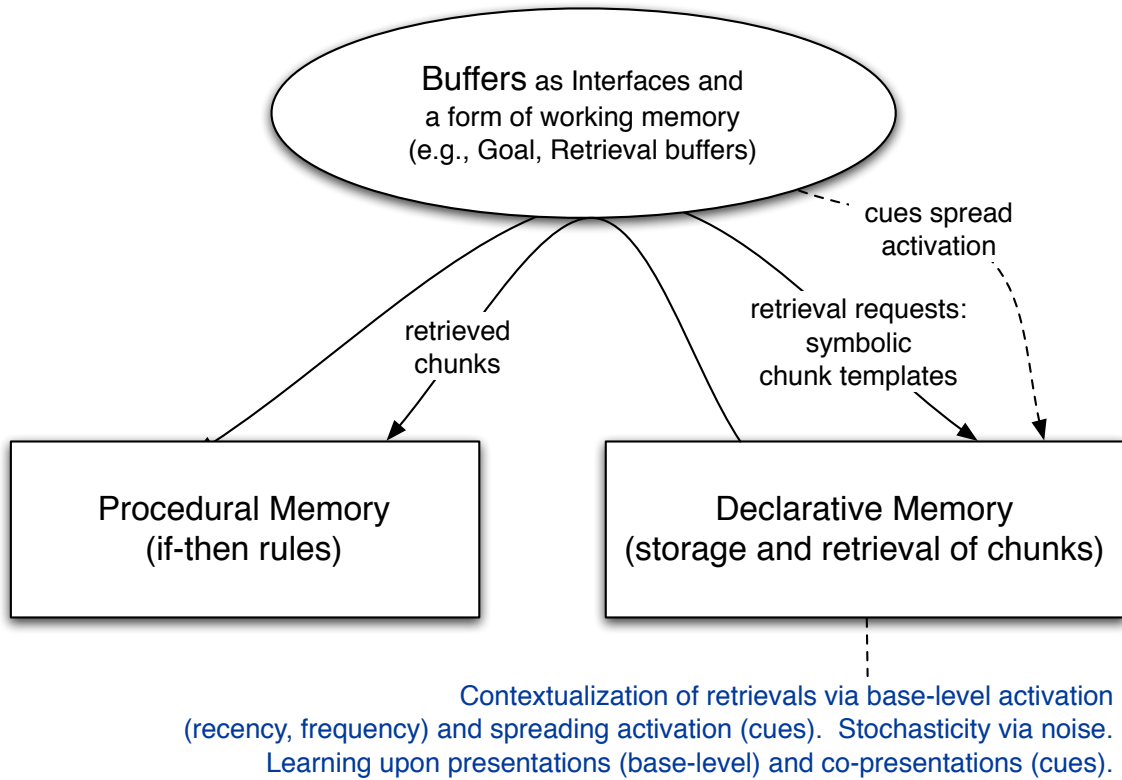
Figure 1. The portion of the ACT-R architecture relevant to the model: Procedural and Declarative Memory interact in ACT-R via Buffers.

Base-level activation is learned over a long period of time and increases with each use of the chunk, which can be thought of as retrieval. The more recent a retrieval, the stronger is its impact; base-level activation decays over time. In the context of priming, base-level activation is the central mechanism used to model the preferential access of memorized material. ACT-R's base-level learning function includes an activation decay that is similar to what is observed in short-term priming. Base-level activation is the first term in a sum describing the overall activation of a chunk *i*:

$$A_i = \log \sum_{j=1}^{n} t_j^{-d} + \sum_{j=1}^{m} S_{ji} + \varepsilon \tag{1}$$

where *n* identifies the number of presentations of the chunk and t_j the time since the *j*-th presentation; *d* is a decay parameter (typically set to 0.5 in ACT-R models). The two remaining terms of the sum describe spreading activation in cue-based memory-retrieval (see below), where *m* is the number of cues. The noise term ε is randomly sampled from a logistic distribution for every retrieval. As an example of the effect of base-level learning, consider Figure 2: here, the activation of a chunk is shown over time, with 14 presentations of the chunk at randomly chosen times (the details of this graph are irrelevant here, and will be discussed as part of Simulation 1 below).

Spreading activation makes it possible to retrieve a chunk given one or more cues. Any chunk

Figure 2. The activation level of the 'ditrans-to' syntax chunk during a series of presentations (retrieval cycles) of this chunk. The activation levels result from ACT-R's base-level learning function, which predicts a decay over time. The dashed lines indicate the activation levels 250 seconds after the first and the last presentation, respectively.

that is present in a buffer may serve as a cue to other chunks held in memory if the model assumes an association S_{ji} between the two chunks j and i. In ACT-R, priming effects are commonly explained by spreading activation. Consider semantic priming between words as an example: *dog* is retrieved from memory more quickly when *cat* has been retrieved recently and is available in a (semantic) buffer. Because the two words are related, activation spreads from this buffer to the chunk *dog* in memory. In the context of language production, spreading activation predicts a facilitatory effect of related linguistic material. For instance, in a head-initial language, the head would facilitate the recognition or production of its complements.

To summarize, ACT-R offers two basic mechanisms that can potentially explain priming: base-level learning and spreading activation. If we model *priming as learning*, then priming emerges through changes in the retrieval probability of syntactic categories stored in memory. If we model *priming as spreading activation*, then priming is caused by activation emanating from lexical forms retained in buffers. In the following, we will argue that an empirically valid model of priming needs to combine both mechanisms. This argument relies on the assumption that there are two kinds of repetition bias, which have distinct cognitive bases: short-term priming and long-term priming (see Section 2.1). Aside from this duality, our argument also draws heavily on experimental evidence regarding the interaction of priming with frequency.

Lewis and Vasishth (2005) present an ACT-R model of language comprehension, in which partial analyses of a sentence are stored in and retrieved from declarative memory as it is being analyzed. Comprehension difficulties are explained through the decay of accessibility of stored information, as opposed to a general cost associated with temporary storage. Their model is of interest here given that comprehension and production systems probably share lexical and syntactic knowledge. Lewis & Vasishth's model differs from our model in that syntactic information is encoded as production rules: "[Their] model ... assumes that much grammatical knowledge is encoded procedurally in a large set of quite specific production rules that embody the skill of parsing. [Their] model thus posits a structural distinction between the representation of lexical knowledge and the representation of abstract grammatical knowledge" (Lewis & Vasishth, 2005, p. 384). This view has much conceptual appeal. However, in Lewis & Vasishth's model, production rules encode grammatical knowledge and the comprehension algorithm at the same time. It remains to be shown how syntactic knowledge in such a model can transfer from comprehension to production.

ACT-R defines a form of sub-symbolic learning that applies to production rules. The assumption is that a rule's *utility* is computed by counting successful rule applications, resulting in a preference score that makes it possible to select the best rule among competing ones. However, ACT-R's current framework does not include a logarithmic decay for rule preferences, contrary to what we see in priming data. Retrieval from declarative memory therefore offers a more plausible account of priming results than the use of procedural memory as in Lewis & Vasishth's model. Furthermore, lexical boost effects require links from lexical to syntactic knowledge. These links are symbolic in their model and cannot readily explain the probabilistic nature of priming and the lexical boost.[2]

Another piece of evidence regarding the mechanism underlying priming comes from Bock et

---

[2]However, a recent ACT-R-based model of syntactic parallelism by Dubey et al. (2008) addresses this point to some degree. Dubey et al. (2008) follow Lewis and Vasishth (2005) in assuming that syntactic knowledge is encoded as production rules, but they show that a utility-based learning mechanism for such rules, combined with a logarithmic decay can account for parallelism effects in sentence comprehension, which Dubey et al. (2008) argue are a form of syntactic priming.

al. (2007) who show that primes that are comprehended result in as much priming as primes that are produced. This points to the fact that declarative rather than procedural memory underpins priming: declarative chunks can be assumed to be the same for comprehension and production, whereas procedural rules would presumably differ for the two modalities (as the parsing and generation algorithms differ), making Bock et al.'s (2007) result unexpected.

2.4.1. Priming as base-level learning

In a learning-based account of priming, we use the ACT-R base-level learning mechanism to acquire a relative preference for CCG syntactic categories. This means that the base-level activation of a syntactic category or its associative links increase with exposure (i.e., with the number of retrievals of that category), and decay with time. CCG categories can express syntactic configurations, and therefore the priming of categories accounts for the priming of syntactic structures (an example of a relevant CCG category is a ditransitive verb expecting a prepositional object). Single words, whose lexicon entries comprise their syntactic categories, exert priming in our model.

In Section 4.1, we will describe a simulation using ACT-R's learning function that accounts for the lexical effect found by Melinger and Dobel (2005), who found priming with isolated lexical items as primes, and also captures the inverse frequency interaction, i.e., the fact that low-frequency syntactic decisions prime more than high-frequency ones. However, three other properties of syntactic priming remain unexplained by an account that relies only on priming as learning:

• Quantitatively, short-term priming and long-term priming seem to differ in their rate of *decay* (short-term priming decays more quickly). Qualitatively, the two effects differ in their interaction with task success, with dialogue genre, and with syntactic properties of what is repeated (see Section 2.1). This means that it is not possible to explain short-term priming and long-term priming as a single, general learning effect. Under such a unified view of priming, we would expect any long-term priming to be preceded by short-term priming. If activation of an item in memory is increased in the long run, then it must also be increased shortly after presentation. Due to the strong decay of short term-priming, interactions with factors such as task success should be stronger rather than weaker for short-term priming.

• A learning-based account of syntactic priming could not explain lexical boost effects. It assumes that lexical and syntactic information is stored and retrieved jointly, hence repetition makes both types of information directly accessible, and we would expect no syntactic priming at all if there is no lexical repetition.

• The lexical boost appears to be restricted to short-term priming. Under a learning-based account that unifies short- and long-term priming, we would expect that both types of priming are equally sensitive to lexical repetition.

2.4.2. Priming as spreading activation

As we saw in the previous section, an explanation of priming that relies solely on base-level learning is not sufficient, as it fails to explain some of the key empirical differences between short- and long-term priming. We must therefore appeal to ACT-R's spreading activation mechanism as well.[3] Priming is then a result of activation that spreads from working memory (i.e., buffers) to longer-term memory, making retrieval both more likely and faster. This assumes that lexical forms

---

[3]It is conceivable to use spreading activation alone to model priming, as suggested by Snider (2008); for a discussion of the shortcomings of this approach see Section 2.2.

used during production are held in buffers for a short while after they have been processed, often beyond the immediate utterance at hand. Holding the lexical forms in buffers is sensible, given that consecutive utterances tend to use overlapping referents if the discourse is coherent (Grosz & Sidner, 1986).

As discussed earlier, priming interacts with syntactic frequency: rare constructions show stronger priming (inverse frequency interaction). For both long-term priming and short-term priming, we can explain the interaction through diminished learning for high-frequency chunks: ACT-R's base-level learning function (see Equation (1)) leads to lower priming when the chunk is common, and associative learning explains additional short-term priming effects in a similar way. Modeling priming as spreading activation, we assume that lexical forms persist in a buffer in order to process their semantic contribution, e.g., for the duration of a sentential unit, until they are replaced by other lexical forms. Similarly, semantic information can persist, even beyond the utterance. By virtue of being in a buffer, lexical forms and semantic information can then spread activation from the buffer to associated chunks in memory, such as to syntactic categories. In effect, while the lexical and semantic material is in the buffer, it is acting as a cue to retrieve a syntactic category (or indeed another lexical form) in the next processing step. The more frequent the syntactic category is, the greater is its prior probability, which leads to a low empirical ratio (see Equation (2), Section A.3 in the Appendix) and a smaller adjustment of the weights between lexical and syntactic chunks, resulting in smaller spreading activation from the lexical cue (prime) during retrieval of the target. In short, highly frequent syntax chunks will see diminished gains from the association with lexical items.

In the next section, we will describe a computational model of syntactic priming in ACT-R that synthesizes the two explanations: priming as learning and priming as spreading activation. In our model, both mechanisms contribute to the overall priming effect observed, but only spreading activation causes lexical boost effects. Our model therefore can be seen as an instantiation of the view that "a basic syntactic repetition effect may reflect the operation of a longer-lived, implicit learning mechanism, whereas in the shorter term, the binding of specific contents (lexical, semantic, or thematic) and positions in specific structures triggers the repetition of structure" (V. Ferreira and Bock, 2006, p. 1025, in a comprehensive review of the experimental literature on priming).

3. A model of syntactic priming in ACT-R

In this section, we describe our ACT-R model of language production, which is designed to account for syntactic priming. We then explain, step by step, how the model produces a natural language sentence. Our description draws on the exposition of the ACT-R architecture in Section 2.4 above, but we introduce additional ACT-R background where necessary. We adopt the typographic conventions in Table 1 for ACT-R components and representations.

3.1. Working memory and procedural knowledge

Working memory in ACT-R is only present implicitly in the form of buffers. Items from memory are passed between declarative memory, procedural control components and other architectural components in buffers. Buffers can only hold a list of attribute-value pairs, where each value identifies a chunk in working memory. For buffers, the values are atomic, which means that the content of a buffer is non-transparent: the internal representation of a chunk in working memory is not directly accessible. ACT-R defines a set of buffers that correspond to interfaces to components of the

Table 1: Typographical conventions adopted for the description of the ACT-R model.

Concept	Typography	Example
Name of a chunk	Single quotes	'offer-lexform'
Attribute of a chunk	Small capitals	AGENTSEM
Value of an attribute	Double quotes	"word"
Type of a chunk	Capitalized	Lexical Form
Name of a rule	Capitalized and italics	*Select-Clause-Head*
Name of a buffer	Capitalized	Goal or Retrieval

architecture. This includes the Goal buffer, which stores the state of the central control mechanism, and the Retrieval buffer which is used for retrieving chunks from declarative memory.

In our model the Goal buffer initially holds a semantic description of the utterance to be produced. This description consists of a flat predicate-argument structure. For a sentence such as *the policeman gave the speeding ticket to the driver*, the semantics consists of a predicate ('give'), an agent ('policeman'), a theme ('speeding-fine') and a goal ('driver'). The Goal buffer also tracks the state of the generation process. This way, rules can ensure that the model's actions are carried out in the correct sequence. Depending on the state of the system represented in the buffers, procedural rules may match and request information from declarative memory. This process is explained in detail in Section A.1 in the Appendix.

3.2. Declarative knowledge: the lexicon

The content of declarative memory in ACT-R consists of chunks, which are defined as bundles of attribute-value pairs. Values may be numbers, strings or references to (names of) other chunks. Unlike linguistic descriptions in the form of feature structures, ACT-R chunks cannot contain other chunks—they can only reference them. Thus, declarative chunks are flat. (This is the reason why the internal structure of a referenced chunk is not accessible.)

Section A.2 in the Appendix gives an overview of the types of chunks that we assume in our model. Priming crucially depends on syntax chunks and lexical form chunks. *Lexical form chunks* contain core linguistic information about a full-form lexicon entry. For example, the following chunk is one of several lexical forms matching the request for the semantics 'speeding-fine':

$$
\begin{bmatrix}
\text{'speeding-fine-lexform'} \\
\text{IS-A} \quad \text{Lexical Form} \\
\text{SEM} \quad \text{'speeding-fine'} \\
\text{LEX} \quad \text{"speeding ticket"}
\end{bmatrix}
$$

Here, 'speeding-fine-lexform' is the name of this chunk, IS-A, SEM and LEX are attribute names, Lexical Form, 'speeding-fine' and "speeding ticket" are values, of which "speeding ticket" is a string, and Lexical Form gives the type of the chunk; 'speeding-fine' is the name of another chunk, indicating the semantics that the chunk realizes. In this model, semantic chunks are atomic and do not carry further information.

The fact that memorized information is divided up into chunks is essential for devising a processing algorithm, demonstrating that language production is possible with proven cognitive

abilities and constraints. It is also important for the functionality and predictions of the model, as each request for a chunk provides an opportunity for the processor to select a chunk out of a set of possible matching ones. Each memory retrieval can potentially lead to delays, errors, and priming biases. Attributes are not the only characteristics distinguishing chunks from one another. A chunk may also relate to other chunks, spreading activation to them whenever it is referenced in a buffer.

The lexical form above is still missing any specification of the syntactic properties of *speeding ticket*. Consider another lexical chunk that needs to be retrieved to realize the sentence: the lexical chunk for *gave*.[4] Its possible syntactic properties are described in two separate *syntax chunks*. This is the syntax chunk for a ditransitive verb with an object in a prepositional phrase with *to*:

$$
\begin{bmatrix}
\text{'ditrans-to'} & \\
\text{I\small{S}-A} & \text{Syntax Chunk} \\
\text{L\small{EFT}} & \text{'trans-to'} \\
\text{C\small{OMBINATOR}} & \text{'/'} \\
\text{R\small{IGHT}} & \text{'NP'}
\end{bmatrix}
$$

A syntax chunk specifies a syntactic category (a syntactic type in CCG terminology); the CCG type encoded here is $((S\backslash NP)/PP_{to})/NP$. The 'trans-to' type describes a transitive verb $(S\backslash NP)/PP_{to}$, so the ditransitive type is defined as a "transitive (to) verb with an additional NP". The 'trans-to' type, in turn, refers to a prepositional phrase type with *to*.

The alternative syntactic realization of *gave*, with a double object construction is encoded in CCG type $((S\backslash NP)/NP)/NP$, which corresponds to the following syntax chunk:

$$
\begin{bmatrix}
\text{'ditrans'} & \\
\text{I\small{S}-A} & \text{Syntax Chunk} \\
\text{L\small{EFT}} & \text{'trans'} \\
\text{C\small{OMBINATOR}} & \text{'/'} \\
\text{R\small{IGHT}} & \text{'NP'}
\end{bmatrix}
$$

Basic experimental results on priming indicate that these syntax chunks are separate from one another, and from the lexical form they share: priming effects can be obtained even if the prime and the target contain different verbs.

Recall that a chunk's activation is determined by spreading activation and base-level activation. In the lexicon, spreading activation makes it possible to retrieve a syntactic variant given the lexical form. Often, lexical forms are connected to several syntax chunks, allowing the speaker to choose different variants. Some variants will be more common and others less common: this distribution (the *frame selection bias*) is reflected in the different strengths of the links from lexical forms to syntax chunks (see Figure 3). For instance, the lexical form *gave* spreads activation to two syntax chunks, namely to $((S\backslash NP)/PP_{to})/NP$ (a ditransitive verb with a prepositional phrase complement; Figure 3, top) and to $((S\backslash NP)/NP)/NP$ (a ditransitive verb with double object complements, Figure 3, bottom). Activation spread is not uniformly distributed across the different syntax chunks—verbs, for instance, will have more and less preferred subcategorization frames. The more

---

[4]We assume sentences to be in past tense; lexical chunks encode full forms and abstract away from regular morphological processes.
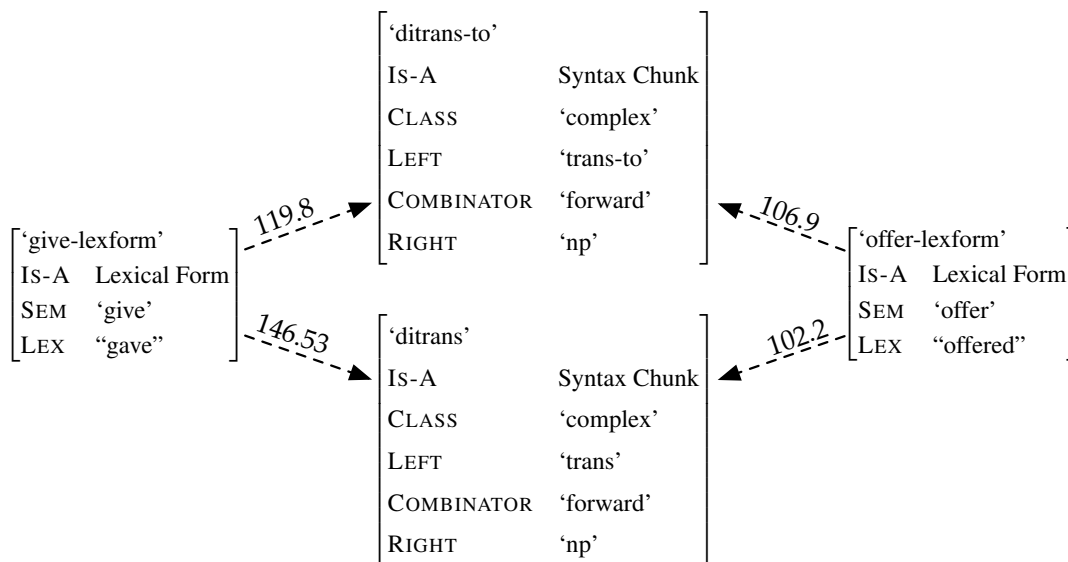
Figure 3. Lexical forms (*gave*, *offered*) can select multiple syntactic categories (ditransitive with one *to* complement, top; ditransitive with NP-NP structure, bottom), which can however differ in frame selection bias. The link strengths shown here were estimated from the Wall Street Journal (CCGBank) corpus (see Section A.5 in the Appendix for details).

common choice is more highly activated, and thus the source of more spreading activation. However, speakers may make other choices, either due to noise or because of priming, which adds to the overall activation of a syntax chunk.

Further chunks encode the order of thematic roles such as AGENT, THEME or GOAL, see Sections A.2 and A.4 in the Appendix.

3.3. Lexical access as the locus of priming

Independent of the specific syntactic representation, a language production process has to access the lexicon to decide about words and the syntactic forms used for words and phrases (in a lexicalized grammar). This is modeled as the retrieval of a pair of lexical and syntactic chunks from declarative memory. During the production process, these chunks are reinforced, so that their relative retrieval probability is elevated after their use. Thus, retrieval and base-level learning of chunks in declarative memory are the core aspects of the model that bring about syntactic priming (Simulations 1–3).

Base-level activation, as discussed in Section 2.4 is a learning mechanism that gives rise to both short-term and long-term priming in the model. Consider again Figure 2, showing the activation of a syntax chunk, which is presented several times over the course of 5,000 seconds (1.4 hours). This figure was generated using the full model, i.e., a full sentence was generated for each presentation of the chunk. The chunk the model activates is a syntactic form for a verb that subcategorizes a prepositional object complement with the preposition *to*, i.e., the form 'ditrans-to'. The more highly this chunk is activated, the more likely the model is to choose the PO variant over the

DO variant at the time.

A further crucial learning mechanism concerns the links between chunks that determine spreading activation during syntactic retrieval, which gives rise to lexical boost effects. The model learns that certain semantic material and certain syntactic choices may co-occur, so that a re-use of semantic material aids the cue-based retrieval of syntactic chunks (Simulation 4 and Experiment 1).

Lexical forms such as *give* or *offer* spread activation to syntactic variants. Because variants are retrieved after lexical forms, the lexical form and its semantic contribution governs the production process. In ACT-R, once a production rule has been selected, its effects can not be undone: there is no backtracking. This means that a choice is not influenced by the accessibility of choices that follow later: the choice of a lexical form does not depend on the preferred syntactic variant. Once a lexical form is chosen, however, the syntactic variant is subject to any bias that is introduced by priming (or other) effects.

3.4. Generation algorithm

In what follows, we describe the algorithm that generates sentences such as *the policeman gave the flower to the girl*. The algorithm assumes the semantic description of a single clause as input, i.e., earlier planning steps have already been carried out to the point at which syntactic realization can begin.

Initially, the Goal buffer holds the current semantics, consisting of a predicate and arguments associated with thematic roles (such as: AGENT, associated with 'policeman'). The names of argument chunks are stored in the slots AGENTSEM, THEMESEM, GOALSEM. During processing, the Goal buffer holds values in the following further slots:

- CONTEXT TYPE, a slot to describe the CCG type (a chunk name) of the currently generated phrase (initially set to a special value 'beginning-of-clause').

- NEW TYPE, a slot to store the CCG type of the portion of text currently generated, which is to be adjoined to the CONTEXT TYPE (initially empty).

The algorithm proceeds as follows:

1. Retrieve a lexical form of the *semantic head* for the semantics in the Goal buffer. This head is taken to be the verb if a clause is to be realized, or the noun for noun phrases.

Repeat:

2. Request and retrieve the next (most active) Thematic Role from memory. Stop if no further role can be retrieved: the sentence has been generated.
3. Identify the argument associated with the retrieved Thematic Role, and request and retrieve from memory a lexical form for the semantics of this argument.
4. Request and retrieve a syntax chunk from memory for the retrieved lexical form and store the LEFT, COMBINATOR, RIGHT values of that node in the Goal buffer as the NEW TYPE.
5. Adjoin: Combine the NEW TYPE with the CONTEXT TYPE according to one of the combinatorial rules.

This algorithm would be sufficient if generation could take place in a fully incremental fashion. However, the notion of *flexible incrementality* (see Section 2.3) requires the syntactic realization

algorithm to be able to plan ahead. We therefore introduce a step of (limited) recursion. The current state (represented by the semantics and by CONTEXT TYPE) needs to be stored. A sub-phrase is begun, starting with an empty CONTEXT TYPE, with new material forming a separate constituent until the current CONTEXT TYPE may be adjoined to the saved type on the stack. Here, we implement a limited version with stack size $n = 1$, replacing step 5 in the previous algorithm as follows:

> 5. Adjoin: Combine the NEW TYPE with the CONTEXT TYPE either according to one of the combinatorial rules or by retrieving a learned combination from memory,[5] updating CONTEXT TYPE with the resulting combination and clearing NEW TYPE. Depending on whether the combination succeeds:
>
> > 1. *If unsuccessful (not combinable) and* STACKED TYPE *is empty,* copy the CONTEXT TYPE into STACKED TYPE and move NEW TYPE into CONTEXT TYPE.
> > 2. *If successful and* STACKED TYPE *is filled,* attempt to adjoin the (new) CONTEXT TYPE to STACKED TYPE, updating CONTEXT TYPE with the resulting combination and clearing STACKED TYPE.

The algorithm can be extended directly to cover n-fold recursion for finite, but normally small n. In the ACT-R context it predicts that the STACKED TYPE spreads activation. An alternative model would store each syntax chunk in declarative memory, effectively creating a probabilistic stack, because the most recently acquired goal would be the most accessible one and could be retrieved first. This variant predicts more processing difficulty in cases where production cannot (or does not) proceed incrementally, including retrieval errors of stacked items.

4. Evaluation

In this section, we will present a set of studies that evaluate the proposed model by simulating the results of experiments and corpus studies on syntactic priming.

Commonly, ACT-R models are evaluated against experimental data collected under controlled conditions. For instance, a production priming experiment may be designed as follows: the participant listens to a sentence (the prime), whose syntactic construction is manipulated. Then, they describe a picture, producing a sentence (the target). Syntactic priming is in evidence if the experiment shows that participants are more likely to generate a given target construction if the prime contained the same construction, compared to when it did not. To test an ACT-R model of priming against such an experiment, a modeler would have to implement a model that can comprehend and produce the experimental materials, observe the reactions of the model to those materials, and then compare these reactions to those obtained from human participants.

To compare a model to a corpus study, the method used for evaluation must be adapted in several ways. A priming experiment controls the semantics of the utterances obtained, disregarding utterances where the participant did not produce a semantically correct target sentence. In contrast, a corpus study has little semantic control over the materials, and there is no semantic coding of sufficient detail available to simulate the production of the data found in a corpus. Even with

---

[5]As a simplification, we use a number of stored combination patterns to specify valid combinations of categories, as not all combinations of deeply hierarchical categories may be tested in the precondition of an ACT-R production rule. This refers to cases where $A/(B/C)$ forward-combines with $(C/(D/E))$ to $A/B/D/E$, a situation where the internal structure of B/C is not accessible directly in the ACT-R rule.

a large-coverage grammar, semantic specifications are typically not constrained enough to reproduce a substantial number of the original utterances, particularly for speech (as opposed to written language). As a consequence, our computational model does not attempt to simulate the raw data (i.e., produce the sentences in the corpus). Instead, our aim is to capture priming by simulating the effects established in the literature. However, our model is designed so that it is capable in principle of producing realistic linguistic output; the syntactic framework underlying our model can cope with a broad range of syntactic phenomena found in corpus data (Hockenmaier & Steedman, 2007).

The ACT-R simulations presented in this section all used a uniform set of parameters. As is common in the ACT-R literature, we did not tune these parameters to maximize model fit; rather we relied on experimentally established default values. Furthermore, our simulations draw on corpus data to determine the relative frequency of lexical entries. Section A.5 in the Appendix gives details on parameter settings and frequency estimation used.

In Simulation 1, we test the basic hypothesis that memory access in ACT-R can provide an explanation for priming effects. In particular, we use a large set of (artificial) verbs to show that ACT-R's base-level learning mechanism produces the short-term priming of syntactic structure and the inverse frequency interaction.

In Simulation 2, we evaluate the full ACT-R model, including the language production algorithm. We simulate the production of a number of sentences, alternating double object and prepositional object realizations of the same semantics. We model a priming experiment in which a participant produces either a DO or a PO variant and is then free to choose either variant in the target elicitation that follows. This evaluation is designed to show that the model adapts to the syntactic variant produced as the prime, and that there is little noticeable decay of this priming (after the first few seconds).

Simulation 3 shows that our ACT-R model can also account for cumulativity effects in priming, i.e., the finding that the strength of the priming effect increases with the number of primes that a speaker is exposed to (Kaschak et al., 2006; Kaschak & Borregine, 2008; Kaschak, 2007; Jaeger & Snider, 2008; Snider & Jaeger, 2009).

In Simulation 4, we extend our model with an associative learning function and demonstrate that it is able to simulate the lexical boost effect for data involving the dative alternation.

Finally, we also present the results of a corpus study (Experiment 1) that tests a prediction that our model makes. Due to the role of spreading activation assumed by our model, it predicts that the lexical boost effect in priming should not be limited to lexical heads; rather priming should be boosted by any type of lexical overlap between the prime and the target. We present corpus data to evaluate this prediction of our model.

4.1. Simulation 1: Learning and short-term priming

This simulation used simulated language data to determine whether base-level learning can reproduce short-term priming effects and model the inverse frequency interaction.

4.1.1. Method

The study was structured as an *acquisition phase* and a *priming phase*. During acquisition, we simulated base-level learning on a set of abstract, artificial syntactic decisions with varying frequencies, not yet using the full language production model. Recall that syntactic decisions are represented in our model by memory retrievals of combinatorial categories, which encode syntactic choices (e.g., whether to realize a sentence with a PO or DO frame). Thus, these decisions are

encoded in ACT-R chunks. Then, during the priming phase, we simulated priming of a syntactic construction through the retrieval of the syntactic decision and examined the resulting activation levels of the chunks.

In the *acquisition phase*, syntactic decisions occurred randomly over a time period chosen to cover short-term priming at a wide range of realistic chunk activations (50,000 seconds). The probability of each rule's occurrence was defined according to its frequency, which was randomly sampled from a Zipfian (power law) distribution. Throughout this period, we simulated exposure to the syntactic decisions. Each decision increased the rule's base-level activation, and this activation decayed over time. The decay is defined by ACT-R's base-level learning function (see Section 2.4). After acquisition, a temporal lag is enforced to let activations settle.

The acquisition phase provided the base-level activation of each syntactic rule. We would expect such activation as a result of the normal learning that occurs as the language processor is exposed to syntactic rules.

During the *priming phase*, we simulated the exposure to the rule that serves as the prime, as it would occur in any utterance in an actual corpus. Then, lag is simulated, ranging from $d = 0$ to 15 seconds, before activation levels are sampled. This mirrors a corpus-based methodology for estimating short-term priming levels, using base-level activation instead of the probability of rule repetition. This is in line with ACT-R's approach, in which a chunk's base-level activation correlates with the retrieval probability of the chunk.

We used a linear mixed-effects model (Agresti, 2002) to analyze the results of the simulation; the model mimics the decay-based method used in prior corpus experiments investigating the priming of general syntactic rules (Reitter, Moore, & Keller, 2006b). The mixed-effects model used activation level as outcome (dependent variable); the predictors (independent variables) were prime-target lag (log-transformed), syntactic rule frequency (log-transformed), and the interaction between lag and frequency.

In this and in all other models reported in this paper, we centered all predictors to reduce collinearity. We checked for any remaining collinearity by determining the correlations between all fixed factors and all interactions in the model. We assume that the absence of collinearity is evident if these correlation coefficients are < 0.2.

In the current simulation, we expect to find a main effect of lag, based on the fact that corpus studies on syntactic priming have reported a negative correlation of lag and repetition probability (Gries, 2005b; Szmrecsanyi, 2005; Reitter, Moore, & Keller, 2006b; Reitter, 2008). Furthermore, we expect an interaction between lag and rule frequency, mirroring the inverse frequency interaction found in priming experiments.

4.1.2. Results

We obtained a significant main effect of log-transformed prime-target lag ($\beta = -0.12$, $p < 0.0001$), indicating that the repetition probability decays logarithmically with time. We also found an interaction of lag with rule frequency ($\beta = 0.0026$, $p < 0.0001$), indicating that decay diminishes with rule frequency. A main effect of frequency was also present ($\beta = 0.0026$, $p < 0.0001$). Detailed model statistics are given in Section B in the Appendix.

4.1.3. Discussion

Our simulation successfully demonstrated syntactic priming, and in addition found that the priming effect is weaker for high-frequency rules. These effects are similar in magnitude to ones
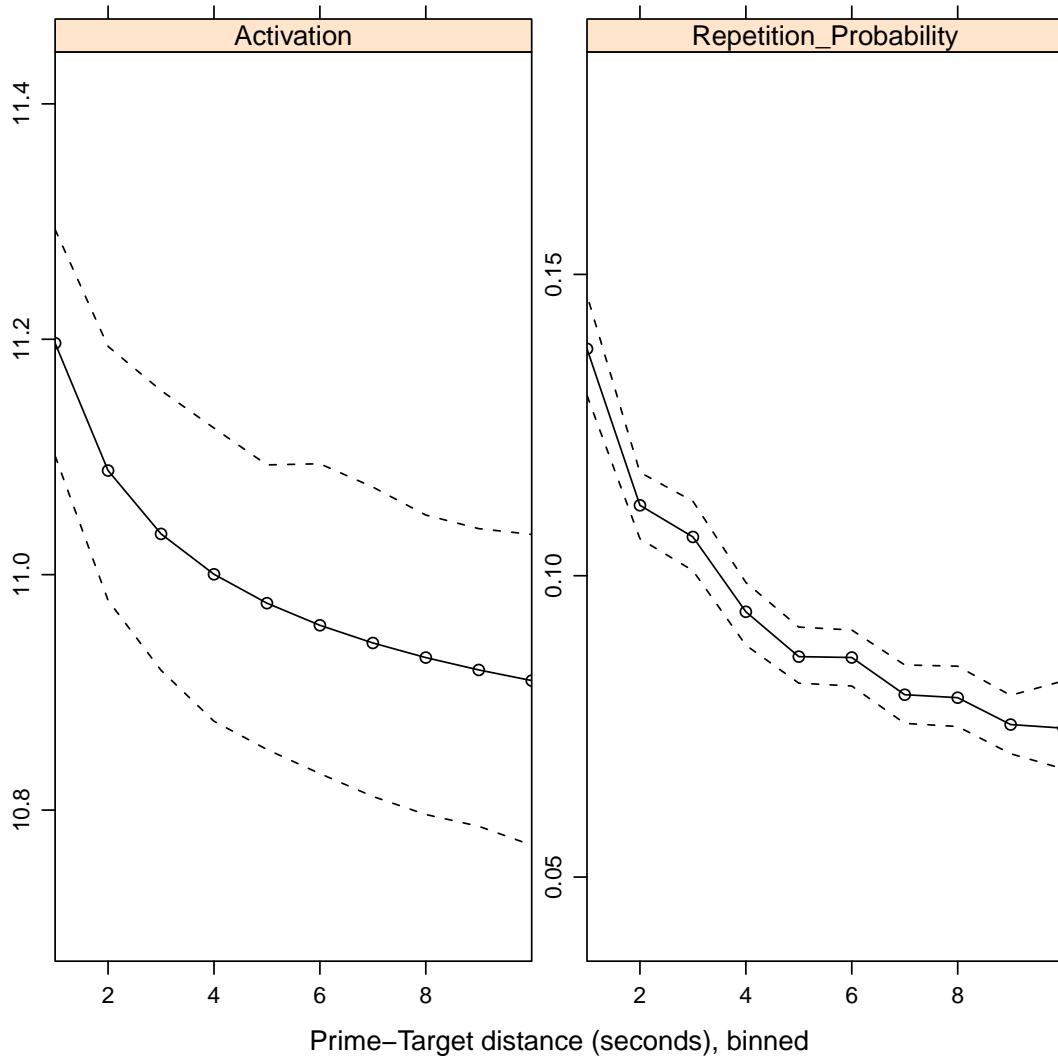
Figure 4. Simulation 1: Base-level activation in the simulation (left panel) and repetition probability for phrase-structure rules in production-production priming in the Switchboard corpus (right panel), for lower-half rule frequencies. Dashed lines: 95% confidence intervals obtained by bootstrapping.

found in corpus-derived data. The basic priming effect in our simulation is due to ACT-R's base-level learning function, which is not unexpected as this function describes a logarithmic decay. Figure 4 (left panel) shows the base-level activation over 1 to 10 seconds (grouped into 1 second bins), beginning from the simulated prime presentation in the model. Figure 4 (right panel) compares this to the repetition probability for the same prime-target lags obtained empirically from the Switchboard dialogue corpus (Marcus et al., 1994). In Switchboard, we measured repetition probability for the lower-frequency half of all phrase structure rule occurrences for utterances produced by the same speakers (production-production priming).

The second result of this simulation is that base-level learning also explains the inverse frequency interaction: we found that low-frequency items show significantly more priming, which is in line with both corpus studies and experimental data on syntactic priming.

Taken on its own, this finding is compatible with an explanation of priming in terms of base-level learning. However, as we argued in Section 2.4.1, this is not sufficient as an explanation for priming effects: it fails to capture the lexical boost effect and is unable to explain important differences between short-term and long-term priming (rate of decay, susceptibility to lexical boost, task factors, and sequence priming). In the next simulation, we will therefore evaluate our complete priming model, which combines spreading activation with base-level learning to obtain a more realistic model of syntactic priming.

4.2. Simulation 2: Priming in DO/PO production

Simulation 1 provided a proof of concept that showed that base-level learning can account for priming in principle. Building on this, the current study tests our full ACT-R model by using it to simulate the generation of sentences. We elicit DO and PO primes by forcing the model to choose a particular syntactic structure given a semantics. Then, another semantic representation is given, and the model is free to choose any syntactic variant. The simulation is set up to be similar to an actual priming experiment. We aim to show that the model exhibits long-term priming similar to what has been observed in actual corpus data.

4.2.1. Method

The model has to generate sentences with semantic representations equivalent to *the doctor gave a flower to the policeman*. Two conditions were used: a prime condition ($\textsc{Primed}_1$), and a control condition ($\textsc{Primed}_0$). In both conditions, the model was first given the semantics to generate from. We alternated a constraint that forced the model to either choose prepositional object (primed) or double object constructions (control condition) in the prime sentence. The model was free to choose different lexical realizations (a number of synonymous noun phrases were given for the arguments).

We then simulated a random lag (60–1,000 seconds, uniformly distributed) with no activity in order to give any short-term effects a chance to decay.[6] Then, a target sentence was elicited (whose semantics was equivalent to that of *the cheerleader offered the seat to a friend*). This time, the model was free to chose a syntactic variant, i.e., *the cheerleader offered the seat to a friend* or *the cheerleader offered a friend the seat*. In each condition, 100 repeated trials were sampled.

---

[6]Simulating the production of intervening sentences as opposed to 'quiet' lag would not affect the activation of those particular syntactic types as long as the sentences do not make use of them. That is, the presentation of other syntactic material does not influence the activation of the PO type.

We did not vary the items used in our simulation. In an experiment with human participants, a set of different stimuli would be used, but given the model is implemented in ACT-R, the only source of variation is the general noise added to the system and the preexisting, corpus-acquired activation of the lexical material and their links to syntax chunks. Therefore, varying the stimuli would not yield different results.

We report the results of a χ^2 test to compare the proportions of DO and PO sentences produced. Our statistics are intended to generalize beyond the activation noise and the model's choices, but not beyond prime/target semantics and verbs. A mixed-effects model with logit link was also fitted to examine whether a number of trials comparable to the experiments could yield a decay effect of repetition probability over time. Such a decay is typical for short-term priming in corpus data, increasing the likelihood of a re-occurrence of a PO structure (in this case) shortly after a PO structure is used. To test for decay, we fit a mixed-effects model with logit-link with repetition as binary outcome (PO PO coded as repetition, DO PO as non-repetition) and the lag between prime and target (log-transformed) as predictor. A random intercept was entered, grouped by items. No collinearity remained after centering; the distribution of the predictor appears normal.

4.2.2. Results

In the control condition, the model produced prepositional object constructions in 23% of the trials. In the PO priming condition, the model produced PO constructions in 38% of all trials. This difference was significant ($\chi^2 = 4.6$, $p < 0.05$): for the given semantics and the verbs used, we obtained significant priming of prepositional object constructions.

No reliable effect could be found for a decay of the priming effect ($\beta = 0.088$, $p = 0.71$) in the mixed model for the simulation data. Detailed statistics for this model are given in Section B in the Appendix.

4.2.3. Discussion

This simulation showed that our model exhibits long-term priming, where long-term priming is defined as the increased repetition of a syntactic construction after a lag of at least 60 seconds following the prime.

Figure 5 compares our simulation results to data presented by Branigan, Pickering, Stewart, and McLean (2000), who tested DO and PO priming in a spoken language production experiment. Branigan et al. did not find an effect of lag between prime and target for DO/PO priming; they compared three conditions: no lag, one intervening sentence, and a temporal lag. In the model, an effect of prime-target lag should be present in theory, given the decay in underlying base-level activation. This decay, however, is too small to be detected if we assume realistic numbers of trials and standard noise levels, as this simulation shows. This is compatible with corpus studies investigating large prime-target lags, in which no effect of lag could be shown (e.g., Jaeger, 2006).

4.3. Simulation 3: Cumulative priming

() As detailed in Section 2.1, Jaeger and Snider (2008) present data suggest that priming is cumulative, i.e., that the strength of the priming effect increases with the number of primes that a speaker is exposed to. Their results concern two structures: optional *that* complementizers and relativizers and the active/passive alternation. For DO/PO structures, Kaschak et al. (2006) present experimental data that can be seen as evidence for cumulativity. In the present simulation study, we
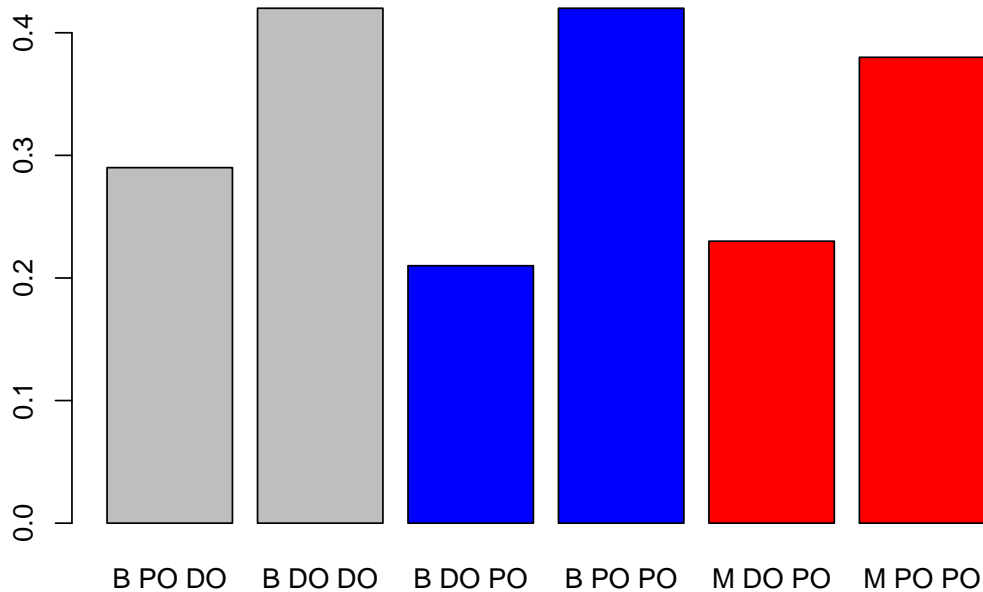
Figure 5. Simulation 2: Proportion of double object (DO) and prepositional object (PO) sentences produced in Branigan, Pickering, Stewart, & McLean (2000), for priming conditions (DO DO, PO PO) and non-priming conditions (PO DO, DO PO). B: Branigan et al. data; M: model data.

examine whether the cumulative priming effect can be replicated by the ACT-R model. We again use the PO/DO alternation as our test case.

4.3.1. Method

As in Simulation 2, we elicited target sentences for a given semantics. This time, we ran two simulations, one using DO primes, and another one using PO primes. We manipulated the number of prime sentences the model was exposed to ranging from 0–25. Fourteen trials were carried out for each number of prime sentences. A random pause was introduced between the prime sentences (5–30 seconds, uniformly distributed). As before, there was also a random lag between primes and targets (60–1,000 seconds, uniformly distributed). A total of 1,300 trials were produced in each simulation.

Again, a mixed-effects model with logit-link was fitted to the data, with the binary outcome coding the repetition or non-repetition of the prime structure in the target. As predictor, we entered the number of primes (centered). A random intercept grouped by items was entered to compensate for repeated measurements (items were repeated for each number of primes). No collinearity remained after centering.

4.3.2. Results

Figure 6 shows the repetition probabilities resulting from the PO and DO simulations. The statistical analysis included all trials with numbers of prime sentences above one, i.e., we only look at cases of priming, which is conservative, as it excludes the relatively strong contribution of the no-priming control case (data points at zero in Figure 6).

For PO primes, the mixed model shows a significant increase of repetition probability with increasing number of primes ($\beta = 0.024$, $p < 0.005$). For DO primes, this effect fails to reach significance ($\beta = 0.01$, $p = 0.35$). Detailed statistics for the mixed model are given in Section B in the Appendix.

4.3.3. Discussion

Long-term priming, according to our model, is cumulative for the prepositional object construction. This effect for PO realizations appears to be weaker than that of *that* complementizer deletion or passive constructions found in Jaeger and Snider's (2008) corpus study, in which they report log-odds of 0.170 for the number of preceding *that* realizations (within speakers) when predicting *that* realization in relative clauses, and 0.214 in complement clauses.

The failure to find reliable cumulativity of DO priming can be attributed to the higher probability of this construction (both relative to PO constructions and overall), which is the only relevant difference between PO and DO in our model. This constitutes an interesting interaction between cumulativity and frequency, perhaps akin to the more general inverse frequency interaction in priming, (the fact that more frequent forms prime less). Note that previous experimental studies of cumulative priming (Kaschak et al., 2006) indicate that imbalanced exposure to syntactic alternatives in the priming phase reduces the priming effect. However, this does not fully explain the findings in this simulation, as Kaschak et al. (2006) found an equal reduction of priming for both alternatives, rather than just for the more frequent one (DO in our simulation).

4.4. Simulation 4: Lexical boost

Simulations 1–3 provided a successful evaluation of our model, but did not deal explicitly with lexical boost effects. This is because simulating the lexical boost in our model requires *associative learning*, which computes the strength of spreading-activation links between chunks. However, associative learning is not implemented in the most recent version of the ACT-R framework (6.0). In Simulations 1–3 we therefore did not stipulate a particular associative learning function; our results hold as long as the associative learning function includes a strong decay (as does base-level learning).

However, this leaves us without a way of capturing lexical boost effects in our model. The current simulation study fills this gap. We extend our model with an associative learning function and demonstrate that it is able to simulate the lexical boost effect for data involving the dative alternation.

4.4.1. Method

The language production model formulated in ACT-R 6.0 was re-implemented using *ACT-UP*, a novel toolbox library. ACT-UP implements associative learning as described in this paper, using an approach based on mutual information (Anderson, 1993, see Section A.3 in the Appendix for details).
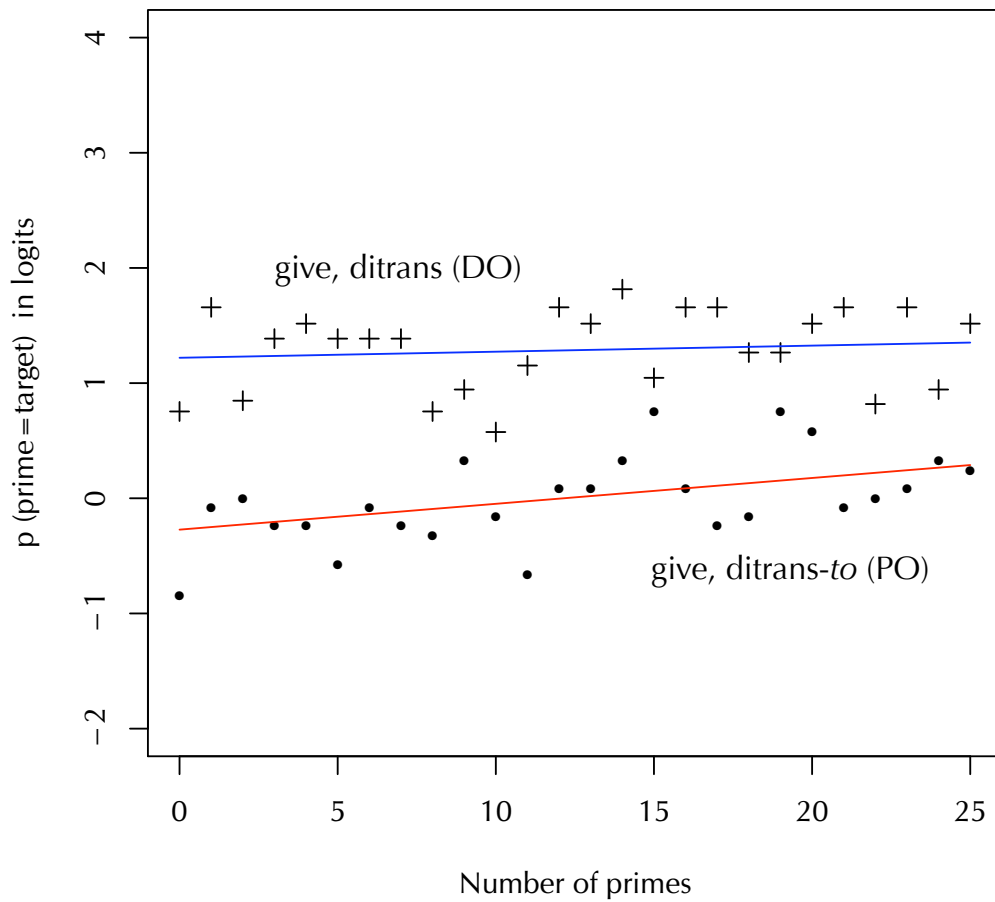
Figure 6. Simulation 3: Proportion of matched targets for 0–25 primes. Dots represent PO primes (and proportions of PO targets), crosses represent DO primes (and proportions of DO targets). The slopes of the two fits indicate cumulative priming, they exclude the no-priming condition (0 primes). Probability scale (y-axis) in logits.

The model used each of the semantic chunks that made up the semantics of the sentence produced as cues in the retrieval of syntactic choices; in ACT-R terms, these semantic chunks were present in buffers while the retrieval of the syntactic type was requested. After the retrieval, the combination of the retrieved syntactic chunk and each of the semantic components were treated as a joint presentation so that their associative weights could be updated.

The language production model was run with the semantics for the sentences (1) *The doctor gave the policeman a flower* and (2) *The girl offered the friend a seat*. Both sentences were tested in their DO and PO variants. We used corpus data to estimate base-level activations and associative links from lexical to syntactic items for the two verbs. The corpus used was the incrementalized CCG version of Switchboard (see Section A.5 in the Appendix), except for the case of *offered*, whose DO/PO distribution was acquired from the ICE-GB corpus of mixed written/spoken English (as reported in Gries, 2005a).[7]

Each trial in the simulation consisted of a prime sentence followed by a target sentence. The model was initialized to the base-level activations and link strengths estimated from the corpora before each trial. The model was able to freely chose a DO or a PO realization during the prime phase (uncontrolled primes). We compared three conditions: In the *semantics-same* condition, prime and target used the same semantic representation (i.e., either the semantics of sentence (1) or of sentence (2)). In the *verb-same* condition, prime and target shared the verbs, but the rest of the semantics was different. In the *semantics-different* condition, the prime used the semantics of sentence (1) and the target that of sentence (2), or vice versa.

A delay of four seconds was assumed after each prime presentation, which matches the conditions in a typical lab-based priming study. A total of $20,000$ trials were run in each condition.

4.4.2. Results

First, we consider DO priming. In the semantics-different condition, the model chose a DO realization in 89.7% of all cases when DO was the prime compared to 83.8% of the cases when PO was prime. In the verb-same condition (partial lexical boost), the model chose a DO realization in 99.98% of the cases in which DO was prime, compared to 9.8% when PO was prime. In the semantics-same condition, the model chose a DO realization in all (100%) DO prime trials compared to none (0%) of the PO prime trials.

Planned paired comparisons (χ^2 tests with continuity correction, not corrected for multiple tests) revealed a significant difference in priming strength between the verb-same and semantics-different conditions ($\chi^2 = 2269, p < 0.0001$), and a significant difference in priming strength between the semantics-same and the verb-same conditions ($\chi^2 = 473, p < 0.0001$).

We now turn to PO priming. In the semantics-different condition, the model chose a PO realization in 16.2% of all cases when PO was the prime compared to 10.3% of the cases when DO was prime. In the verb-same condition (partial lexical boost), the model chose a PO realization in 90.2% of the cases in which PO was prime, compared to 0.0002% when DO was prime. In the semantics-same condition, the model chose a PO realization in all (100%) PO prime trials compared to none (0%) of the DO prime trials.

Planned paired comparisons as before revealed a significant difference in priming strength between the verb-same and semantics-different conditions ($\chi^2 = 1956, p < 0.0001$), and no difference

---

[7]The PO and DO realizations of this verb were not present in sufficient quantities in Switchboard; the same holds for the other verbs in the materials of (Pickering & Branigan, 1998).

in priming strength between the semantics-same and the verb-same conditions ($\chi^2 = 0.003, p > 0.95$).

4.4.3. Discussion

This simulation used a small sample of lexical and semantic materials to show that associative learning can account for lexical boost effects. We find that verb repetition from prime to target significantly increases priming compared to cases in which the semantics of the prime and the target is completely different. This effect holds for both frequent structures (DO) and for infrequent ones (PO). In addition, we found that DO priming increases further compared to verb repetition if the whole semantics is repeated, i.e., there is maximal lexical overlap between the prime and the target. The simulation was therefore clearly able to model the lexical boost effect, and also provides some evidence that the lexical boost effect is not restricted to verb (or head) repetition, as indicated by recent experimental and corpus results (Raffray & Scheepers, 2009; Snider, 2008, 2009).

The results of this simulation demonstrate that associative learning can account for the lexical boost effects, but the form of associative learning we assumed (Anderson, 1993) is not the only realistic mechanism. In fact, a decay of associations, balancing recency and frequency, could also explain the apparent rapid decay of the lexical boost effect, as could the acquisition of lexemes as declarative chunks. There is insufficient psycholinguistic evidence at this time that could distinguish such variants, and the exact form of associative learning is similarly unclear from a cognitive perspective.

While the model showed behavior consistent with lexical boost effects, we note that the distribution of DO/PO realizations for each verb is a result of the specific structural frequencies found in the corpora used. The learning effects depend on the initial link weights between lexical and syntactic chunks. As a consequence, the magnitude of the learning effects differ for specific verbs (*give, offer*).

4.5. Experiment 1: General lexical boost

Most existing experimental results (e.g., Pickering & Branigan, 1998; Branigan et al., 1999; Cleland & Pickering, 2003), show that the lexical boost effect increases the priming of verb phrase structure if the verb is repeated; the priming of noun phrase structure is enhanced by repetition of the head noun. Thus the lexical boost seems to be driven by head repetition. However, the prediction of our model is more general: it predicts a boost whenever lexical material is repeated, whether it is the head or not. This boost effect emerges from the spread of activation from general lexical and semantic material present in the Goal buffer to syntactic material as it is being retrieved. Such lexical-syntactic associations are acquired independently of whether lexical material happens to serve as the head of a phrase. In Simulation 4, we found some evidence for such a general lexical boost: for DO targets, full lexical repetition increased the amount of priming compared to verb repetition only. The current corpus study has the aim of generalizing this finding from a specific alternation to arbitrary syntactic rules.

This experiment also aims to clear up another issue with the lexical boost, viz., an alternative explanation in terms of frame selection bias. Heads (e.g., the verbs in DO/PO structure) commonly show frame selection bias, i.e., they introduce a distribution of the possible syntactic variants of the structures they govern: For verbs such as *give* the DO realization is more likely than the alternative PO realization, while for other verbs such as *offer* DO and PO are approximately equally likely (see

Figure 3). It is possible that the lexical boost effect is explainable in terms of this selection bias (if the verb is repeated, so is the selection bias, thus increasing priming, see Section 2.1 for details).

In this experiment, we therefore ask two questions: (a) Is it the frame selection bias of the head that causes the lexical boost? (b) Can only heads lead to a short-term lexical boost? Note that these questions are of theoretical importance: accounts of language production that select the head and then plan the structure of the constituent (e.g., Pickering & Branigan, 1998) only predict the head to boost syntactic priming, while models of incremental production allow other lexical material to boost priming as well (e.g., de Smedt & Kempen, 1991; Hoffman, 1995; F. Ferreira and Swets, 2002).

4.5.1. Method

Examining the role of heads in a variety of structures necessitates large data sets. To test the prediction arising from our model, we turn to a corpus of recorded, spoken dialogue. We use the Switchboard corpus, which contains spontaneous telephone conversation between North-American speakers (see Section A.5 in the Appendix for details). We annotated the syntactic rules extracted from the corpus with lexical heads using head-finding rules listed by Collins (1999). We quantify syntactic priming by looking at the repetition probability of syntactic phrase structure rules. This method has been developed previously (Reitter, Moore, & Keller, 2006b) and also tested using CCG (Reitter, Hockenmaier, & Keller, 2006).

A corpus study affords us with the ability to observe even weak effects and their interaction through the use of a large data set (Bresnan, Cueni, Nikitina, & Baayen, 2007). Compared to specifically designed experiments with human subjects, the lack of control in corpora is compensated for not just through a large number of samples, but also through the post-hoc introduction of explanatory variables as predictors in the statistical model. This approach has been used successfully in corpus studies investigating priming of individual syntactic constructions (see for example Jaeger, 2006; Jaeger & Snider, 2008). In the present study, we are interested in priming in arbitrary syntactic rules (following Reitter, Moore, & Keller, 2006a); this means that it is more difficult to introduce control variables (as they differ from rule to rule). However, by using decay (prime-target lag) and rule frequency as a covariates in the model, we are able to control for frequency biases.

To capture short-term priming, we compute the temporal decay of the probability that a phrase-structure rule is repeated after a prime. A case of rule repetition is defined as the double occurrence of a phrase-structure rule, first at time t (prime occurrence) and then again at time $t + d$ (target occurrence). We sample each target occurrence at $t + d$ and test whether repetition occurred previously in a one-second time window $[t - 0.5; t + 0.5]$. The model estimates the rule repetition probability based on the proportion of repetitions to non-repetition. It is thus able to predict repetition probability based on the lag (distance) d between prime and target. Under the non-priming null hypothesis, we would expect the rule repetition probability to be independent of the lag d, but previous corpus-based work (Reitter, Moore, & Keller, 2006b) observed a negative relationship between d and repetition, i.e., repetition probability decays as the lag increases. The strength of this decay is expressed in the statistical analysis as a covariate, lag. Using decay rather than conditional probabilities controls for any frame selection bias, which occurs when head lexical forms prefer certain structures and head repetition would confound syntactic repetition.

Our initial question was whether heads or lexical material in general boost priming. In our approach, such a boost is expressed as an interaction between the effect of lag and the proportion of repeated lexical material. A lexical boost favoring heads would show up as a negative interaction of
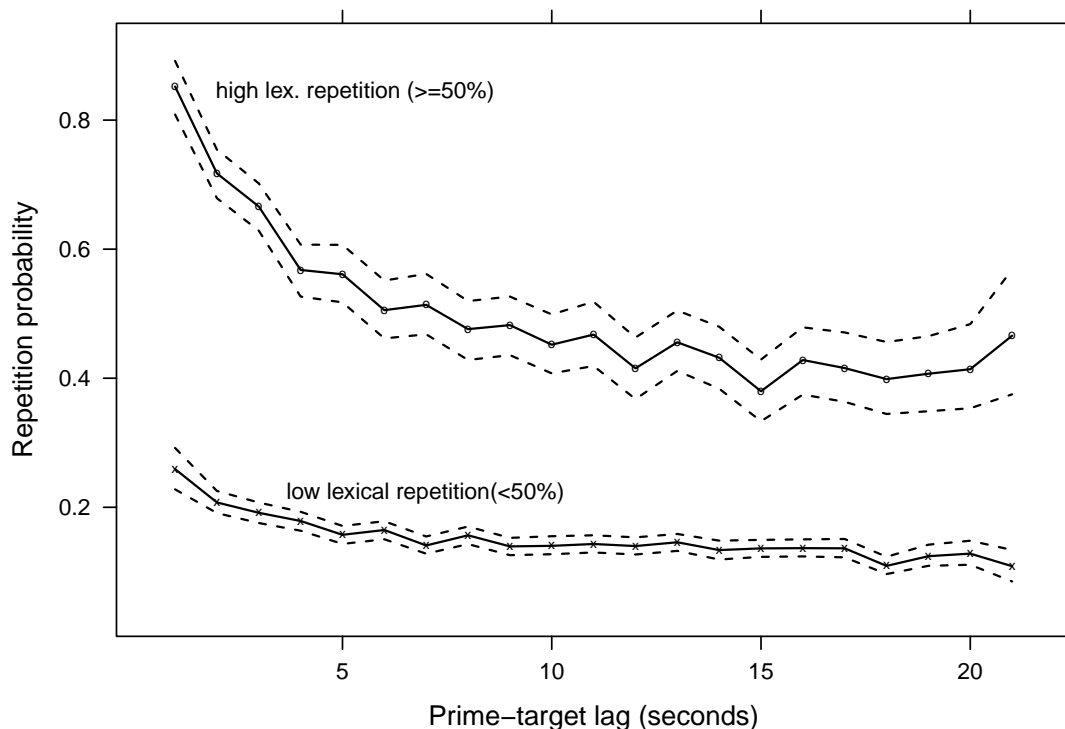
Figure 7. Experiment 1: Probability of rule repetition relative to the time lag between the prime and the target; separately for items with more or less than 50% of lexical material repeated between prime and target. Data source: Switchboard corpus. Dashed lines: 95% confidence intervals obtained by bootstrapping.

head repetition with lag, i.e., head repetition would strengthen the decay. Similarly, a lexical boost favoring general lexical repetition would show up as negative interaction of word repetition with lag. To measure lexical repetition, we make use of the notion of the *yield* of a rule: this is the sequence of words covered by the constituent that the rule defines. For instance, the yield of a simple noun phrase rule such as $NP \rightarrow DET\ N$ consists of all the words contained in the noun phrase. In the analysis, we only include data from cases of rule repetitions where at least one word, but not all words, were repeated between prime yield and target yield.

In a logistic linear mixed-effects model with logit-link, we again use syntactic rule repetition as binary outcome. As predictors, we use log-transformed lag (distance between prime and target in seconds), head repetition as a binary factor, as well as a measure of word repetition (logit-transformed): the proportion of repeated words between prime region and target constituent. If all the words in the yield of the target rule also occurred somewhere in the prime yield, the proportion would be one (though this case was excluded from our data, as we do not want to count verbatim repetitions of whole phrases). Utterances (items) were fitted as random slopes (for lag) to account for possible interdependencies of syntactic rule use (this is conservative compared to fitting random intercepts). No significant collinearity remained for any of the main effects or interactions reported in the results section (correlation coefficients < 0.2).

4.5.2. Results

We obtained a significant effect of lag ($\beta = -0.194$, $p < 0.0001$); the fact that the coefficient is negative indicates that repetition probability decreases with time, providing evidence for syntactic priming in our data set. We see no interaction of head repetition with lag ($\beta = -0.011$, $p = 0.73$). However, we observe a significant interaction of word repetition with lag ($\beta = -0.174$, $p < 0.0001$), indicating stronger decay the more words are repeated within a syntactic phrase. This suggests that any lexical repetition boosts priming rather than specifically head repetition. (See Appendix, table 4 for the model.)

4.5.3. Discussion

Figure 7 shows the development of rule repetition probability, comparing low lexical repetition with high lexical repetition. The short-term priming effect is clearly stronger in the case of high lexical repetition.

While consistent with the literature that finds a lexical boost for head repetition, the results of this experiment generalize the lexical boost effect to other lexical material, for arbitrary phrase structure rules. We found that once the (constant) frame selection bias is accounted for, heads play no special role compared to other lexical material. This is consistent with the small number of studies in the literature that have looked at non-head repetition: in two DO/PO priming experiments, Raffray and Scheepers (2009) found that priming is enhanced not only by verb repetition, but also by the repetition of other constituents in the sentence (representing the agent, theme, and recipient of the verb). In a corpus study, again using the DO/PO alternation, Snider (2008, 2009) found that priming increases with increased syntactic and semantic similarity between the prime and the target, which is compatible with a generalized lexical boost effect. The current corpus study extends these results by showing that the generalized lexical boost is not limited to DO/PO and other syntactic alternations, but applies to priming of syntactic rules in general.

As shown in Simulation 4, the lexical boost effect can be captured in terms of associative learning, which increases the links between lexical material (in a buffer) and a syntactic construction. This increase in link strength occurs with any lexical material present in the buffer (not just heads) and applies to all syntactic nodes that are retrieved while this material is still in the buffer. Stronger links lead to stronger lexical and syntactic associations. An underlying assumption is that learned association strengths decay in a way similar to base-level activation. The greater the lag between prime and target, the smaller is the effect of lexical repetition, because at greater lags, the learned associations have decayed more. Thus, this mechanism explains how spreading activation patterns are acquired and why the lexical boost occurs only over a short period of time after the prime. This is illustrated in Figure 7, which shows that lexical repetition does not make a (detectable) difference after approximately 10 seconds. Note that existing corpus studies investigating the general lexical boost (e.g., Snider, 2008, 2009) do not deal with decay.

5. General discussion

5.1. Comparison with other models

The focus of this article has been the development of a model of syntactic priming. However, in order to achieve this, we also had to model those aspects of the language production process as part of which priming emerges. In the following, we will therefore provide a brief comparison of our

model to standard models of language production. We will then contrast our model with existing computational models specifically designed to capture syntactic priming, following on from the discussion in Section 2.2.

A well-established model of human language production is Levelt's (1989) model Speaking. It assumes several modular processing components, which do their well-delineated work autonomously. Levelt's model provides a comprehensive architecture for language production (rather than a model of priming in the syntactic realization process as in the model presented here). It generates speech incrementally, but does not give an incremental account of syntactic processing like our model. Levelt's model distinguishes lexical and syntactic encoding more than lexicalized models. It makes a distinction between the lemma (the lexical entry defined in syntactic and semantic terms) and the form (the lexeme defined in phonological terms). This distinction is compatible with our model: information can be stored on the lexical level, i.e., in the lexical form chunks, or it can be represented in syntax chunks. While Speaking focuses mostly on the nature of the lexicon, our model is more concerned with how syntactic information guides the combination of words into phrases and sentences, rather than on specifying the overarching architecture.

Performance Grammar (de Smedt & Kempen, 1991; Vosse & Kempen, 2000; Bond, 2005) models lexical-syntactic processes in comprehension and production. In their model, the retrieval of information from memory shares properties with retrieval in ACT-R, as does the merging of information, for instance during lexical retrieval and syntactic attachment. These unification processes are non-recursive, just like in our model. A crucial difference, however, is how syntactic composition takes place. Performance Grammar presupposes a network of interconnected lexical nodes, i.e., it goes beyond our assumption of a very limited working memory during incremental generation (made possible by CCG's combinatorial properties). Essentially, Performance Grammar resembles models that assume temporary storage in a dedicated memory component, rather than in data structures equivalent to ACT-R's buffers.[8]

The network model of language production proposed by Roelofs (1992, 1993) specifies an encoding of syntactic preferences for verb forms that is similar to our spreading activation account within ACT-R. This model has been extended by Pickering and Branigan (1998) to form a theory of syntactic sentence production. There, attributes such as tense, aspect, and number are encoded separately. Syntactic representations are separate from the word form, to account for the fact that lexical repetition increases priming. Syntactic variants are encoded as combinatorial categories such as NP, NP (forcing a DO construction) and NP, PP (forcing a PO construction). However, this model does not store syntactic knowledge in lexicalized categories, but keeps a separate representation of word categories (such as *Verb*). Priming follows from a pre-activation of the combinatorial categories. The resulting model is able to capture basic priming effects, but fails to explain a range of findings pertaining to short- and long-term priming, e.g., the fact that short-term priming shows a lexical boost, while long-term priming does not.

We will now turn to existing models that are specifically designed to account for syntactic priming and compare them to the model proposed in the current paper. The first such model is Chang

---

[8]As a reviewer points out, there are syntactic structures for which our limited memory assumption may lead to problems. An example are extraposed relative clauses in Dutch or German: the gender of the relative pronoun agrees with that of the antecedent head noun, but the main verb of the sentences intervenes between the two. It follows that our assumption that a single category is sufficient to describe the combinatorial properties of an incrementally generated phrase is not adequate for this structure, and more complicated storage mechanisms, such as the one posited by Performance Grammar, may be required.

et al.'s (2006) Dual Path Model, a connectionist model of syntactic priming based on a simple recurrent network architecture. As we pointed out in Section 2.2, the Dual Path Model has two key conceptual limitations: because of its architecture, it is unable to capture the qualitative differences between short-term priming and long-term priming. Furthermore, it does not have a way of capturing the hierarchical relations between words and phrases, as it represents the input as a sequence of linguistic units. Our model overcomes both of these limitations: it exploits ACT-R's distinction between base-level learning and spreading activation to distinguish long-term and short-term priming. Hierarchical syntactic representations are built into our model as it uses CCG structures to simulate language production. On the empirical side, the Dual Path Model fails to model lexical boost effects and is unable to simulate production-to-production priming (because it uses error-driven learning). Again, our model does not share these limitations: it captures the lexical boost effect (as shown in Simulation 4, see Section 4.4), and it can in principle model both comprehension-to-production priming and production-to-production priming (though we have focused on the latter in the present paper).

Malhotra's (2009) connectionist model of syntactic priming overcomes some of the limitations of the Dual Path Model by using a dynamic systems approach, rather than relying on error-driven learning. It is able to distinguish short-term and long-term priming and successfully captures the lexical boost, the inverse frequency interaction, and cumulative priming. However, a key conceptual limitation remains: Malhotra's model is not able to represent hierarchical syntactic structure; it is only able to deal with a limited number of pre-determined syntactic alternations, which are implemented as a look-up table. Also, simulating production-to-production priming is not straightforward in Malhotra's model, all of his simulations are for comprehension-to-production priming.

Snider's (2008) exemplar-based model of syntactic priming takes a different approach from both Chang et al. (2006) and Malhotra (2009). It combines a spreading activation model with all-subtree syntactic representations familiar from Data-oriented Parsing (DOP; Bod, 1992). It is able to capture the inverse frequency effect and the lexical boost effect, as well as the neighborhood density effect (not dealt with in the present paper). However, it is unclear how Snider's model can capture decay in priming or account for the difference between long-term and short-term priming, as it only has a single mechanism at its disposal, viz., spreading activation. Other limitations include the absence of a learning mechanism that derives association strengths for DOP-style representations. It also inherits from DOP the problem that all-subtree representations are very memory intensive, while human performance is characterized by efficient memory storage and retrieval. These limitations are not shared by the ACT-R model proposed in the current paper, which uses base-level learning in addition to spreading activation and is thus able to distinguish long-term and short-term priming, and capture decay in short-term priming. Our model inherits the ACT-R learning mechanisms for learning base-level probabilities and spreading activation weights (see Section 2.4 and Section A.3 in the Appendix). Finally, in contrast to DOP representations, the CCG representations used by our model abstract over syntactic structures and thus enable efficient storage and retrieval.

Recently, Snider and Jaeger (2009) proposed an alternative model that is able to integrate various sources of probabilistic information in priming. They assume that the activation of a syntactic alternative is correlated with its surprisal, which they define as the linear interpolation of the prior probability of this alternative and a probability based on recent experience with the alternative. The model also incorporates an ACT-R-style activation decay and is thus able to account for the fact that priming decays. It is able to simulate the data from an experiment on the interaction of cumulativity and the inverse frequency effect in DO/PO priming, revealing recent experience is weighed more

strongly than prior probability. It is important to note that Snider and Jaeger's (2009) model is not a model of language production: the authors only provide a description of how activation strength is computed, they do not commit to any architectural or representational assumptions (and it seems their model is not actually implemented in ACT-R). Also, their model differs from ours in that it does not make reference to ACT-R's distinction between base-level learning and spreading activation learning (see Sections 2.4.1 and 2.4.2). It seems that the authors assume a uniform learning mechanism for both prior probabilities and experience-based probabilities (presumably base level learning, though the details are not spelled out by Snider & Jaeger, 2009). Furthermore, the decay function used by Snider and Jaeger (2009) applies equally to prior probabilities and experience-based probabilities, which means that their model cannot readily explain the fact that short-term priming decays, while long-term priming does not. Also, it is unclear how their model would capture lexical boost effects; this requires spreading activation in addition to base-level learning, as we argued in Section 2.4.2.

5.2. *Limitations of the lexicalized, memory-based model*

In the model presented here, we hypothesize syntactic priming to be a result of more general cognitive phenomena affecting syntactic processing (see Bock & Loebell, 1990; Hartsuiker, Kolk, & Huiskamp, 1999; Pickering & Branigan, 1998; Pickering, Branigan, & McLean, 2002; Cleland & Pickering, 2003). Syntactic priming does not primarily arise from effects operating on semantics or on the surface level, i.e., lexical choice or the sequencing of words or part-of-speech categories. Rather, we assume that priming applies directly to representations of syntactic structure.

The proposed model relies on combinatorial rules to encode syntactic structure; this means it always combines immediate dominance (the hierarchical organization of phrases) and linear precedence (the sequence in which phrases appear on the surface). This is compatible with claims by Pickering et al. (2002), who argue that there is no explicit linearization phase in language production, based on experiments that investigate priming in heavy NP shift. They find that shifted constructions such as *the racing driver showed to the helpful mechanic the problem with the car* fail to prime non-shifted constructions such as *the racing driver showed the extremely dirty and badly torn overall to the mechanic*, even though both constructions share the same immediate dominance relations. This leads to the conclusion that immediate dominance and linear precedence are computed in a single step, i.e., only constructions that share both prime each other, which is also the assumption underlying our model. Recent evidence from crosslinguistic priming studies, on the other hand, suggests that immediate dominance and linear precedence may be represented separately (Shin & Christianson, 2009); we will return to this point in Section 5.4 below.

Short-term priming emerges, in the model, as a consequence of two decay functions: base-level learning and associative learning. Short-term priming is primarily a contextualization effect, caused by semantic material present in a buffer in order to generate a constituent or a whole utterance. Associations between semantic and syntactic choices are learned before they decay rapidly; this property causes the lexical boost. The exact formulation of a learning function for associations is less clear; we have tentatively assumed the mutual information criterion proposed by Anderson (1993), which makes it possible to estimate associative learning from corpora. While Simulation 4 provided tentative evidence for this approach, future work will have to compare this with other possible ways of formalizing associative learning.

It is important to note a possible alternative explanation for the short-livedness of lexical boost effects. If semantic and syntactic material is retained in the buffer across utterances, it would

spread activation, making repeated syntactic choices more likely. This implies strong lexicalization, as lexicalization means that syntactic and semantic material are retained together. Support for such a retention view comes from coherence phenomena in discourse. Without discussing the details of coherence models, their essence is that sentences aim to continue the topic of a preceding sentence, placing referents presented late in the previous sentence early in the current one. Centering, a prominent theory of discourse coherence, posits: "Sequences of continuation are preferred over sequences of retaining; and sequences of retaining are to be preferred over sequences of shifting." (Grosz, Joshi, & Weinstein, 1995, p. 214, rule 2). In that case, we would say that the short-lived enhancement causing strong short-term priming and lexical boost effects are based on the same semantic retention effect that causes coherence. A testable prediction is a correlation of the effects: sentences between which a topic is continued would be more likely to show short-term priming and lexical boost effects.

It is generally unclear whether short-term priming shows decay over time, or whether the loss of priming is correlated with syntactic, or semantic processes. A semantic account, e.g., would predict that intervening semantic activity would lead to stronger activation decay. In this case, less coherent pairs of sentences would show less priming than more coherent ones: for instance, priming would be weaker when the topic shifts between the sentences. The ACT-R model assumes temporal decay of base-level activation.

Our model posits that lexical choice precedes and thus influences the syntactic decision. The chosen syntax then determines the argument order. Thus, it predicts no argument order priming isolated from lexical of syntactic repetition. Empirically, however, argument order preferences may influence syntactic choice, e.g., in order to satisfy information structure related conventions such as the theme-rheme ordering specifying a preference to present known information early in the sentence, and new information late (Halliday, 1967). This represents a challenge for the current model. Still, such a bias could be modeled as spreading activation in ACT-R. In that case, a semantic chunk still present in a buffer from the latter part of the previous utterance will act as a cue and lead to an early selection of the same material in the following sentence (as a syntactic subject). The selection of the syntactic ("functor") chunk will then be delayed until after the selection of the subject.

5.3. Explanations for comprehension priming

Syntactic types are stored and retrieved from declarative memory; their representations are shared between comprehension and production processes. This predicts priming between the two (e.g., Bock et al., 2007), albeit at different magnitudes. For general priming, we need to consider that learning in ACT-R's declarative memory depends on rehearsal; the processes associated with comprehension may involve fewer rehearsals. For short-term priming and lexical boost effects, comprehension and production priming could differ due to due to differences in retrieval cues. This is because associations between the lexical entry and retrieval cues may be acquired differently (cues are representations of the written or spoken word in comprehension and of the semantics in production).

5.4. Predictions for crosslinguistic priming

There is a growing body of work that investigates syntactic priming across languages (see Pickering & Ferreira, 2008, for an overview). Such experiments test bilingual speakers, and present the prime in one language, and the target in the other language. Most studies find not only that there

is crosslinguistic priming, but also that the magnitude of the effect is comparable to monolingual priming. Furthermore, there is a lexical boost in crosslinguistic priming if the heads of the prime and the target are translation equivalents.

Pickering and Ferreira (2008) assume that crosslinguistic priming can be explained by the fact that bilingual speakers maintain linguistic representations that are shared across languages if the two languages have matching constructions. If we follow this assumption, then we can readily accommodate crosslinguistic priming within the model proposed in this article. If we assume that syntax chunks (e.g., 'ditrans' and 'ditrans-to' in Figure 3) can be shared across languages, then this explains how crosslinguistic priming is possible in our model: through the same mechanism as monolingual priming. We also assume that lexical forms, which contain the surface realizations of words (e.g., 'give-lexform' and 'offer-lexform' in Figure 3), differ across languages; however, translation equivalent forms are connected by spreading activation links (e.g., a link between the English and the Spanish version of 'give-lexform'). In this scenario, the lexical boost effects for translation equivalent words can be explained by activation that spreads from the lexical form in one language to the equivalent form in the other language.

A prediction that follows from our model is that crosslinguistic priming can only occur in structures that share both linear precedence and immediate dominance relationships between languages. (The reasoning is the same as for monolingual priming, as detailed in Section 5.2.) This prediction seems to be borne out in the experimental literature on crosslinguistic priming: for instance, Loebell and Bock (2003) found no priming between English sentences such as *The river was poisoned by the chemical waste* and the German equivalent *Der Fluss wurde von dem chemischen Abfall vergiftet*. Both sentences contain a passive structure, but their linear order differs, as the main verb is sentence final in German. A number of studies have found similar results: Bernolet, Hartsuiker, and Pickering (2007) found no crosslinguistic priming between English and Dutch NPs (which differ in word order), but they did find it between German and Dutch NPs (which share the same word order). Similarly, Salamoura and Williams (2007) found that Greek PO/DO constructions prime English ones, but the effect disappears if the prepositional phrase complement in the Greek prime precedes the noun phrase complement (i.e., PP-NP structures did not prime NP-PP structures).

More recently, however, contradictory evidence has been presented by Shin and Christianson (2009). They found priming between Korean and English constructions even if the linear precedence relationship differed between prime and target (Korean, like Greek, allows flexible word order). This result is at odds with previous findings and poses a problem for our model (and for any account in which priming requires both linear order and immediate dominance to be shared, from Pickering & Branigan, 1998, onwards).

6. Conclusion

In this article, we introduced a model of syntactic priming that is able to explain key empirical properties of priming in terms of well-established general memory retrieval mechanisms realized in the ACT-R cognitive architecture. Our model does not claim to be a full account of the language production process, but focuses on those aspects of language production that are involved in syntactic priming. Nevertheless, our model is able to generate English sentences, starting from simple semantic representations encoded as ACT-R chunks. The implementation of the model we presented focused on the dative alternation, a syntactic alternation that has traditionally been used to show priming effects in both experimental and corpus data. The syntactic framework used, Com-

binatory Categorial Grammar, is flexible enough to describe a wide range of syntactic phenomena as they occur in natural text and dialogue, in data sets such as Switchboard. Crucially, CCG also supports incremental language production without the need to store large amounts of information during processing.

In a series of simulations, we showed that the model accounts for key findings in the priming literature, viz., the inverse frequency interaction, the absence of a decay in long-term priming, and the cumulativity of long-term priming. Furthermore, we argued that the model can also explain the lexical boost effect and the fact that it only occurs in short-term priming. Finally, we presented corpus data that verify a prediction of the model, i.e., that the lexical boost is driven by all lexical material, rather than just heads. In more general terms, our model embodies the claim that priming applies to syntactic structure, in the form of combinatorial categories as syntactic descriptions of subcategorization properties. This is in line with corpus evidence showing that priming is sensitive to constituent structure, and that lexical and non-lexical categories of the same type prime each other. Our model gives an explanation for such effects that is missing in prior models of syntactic priming.

The proposed model is compatible with a flexible notion of incrementality. It composes syntactic structure incrementally by default, even though planning is possible. To date, the exact extent of incrementality in production has not been investigated experimentally using syntactic priming. The model's ability to plan is limited by cognitive resources, with incremental production being the most efficient way to construct the syntax of sentences. Therefore, the model predicts that sentences which require non-incremental production, will take longer to produce and yield more errors.

The memory-based model we presented may also provide an explanation for priming at other levels, such as lexical and phonological priming. A more detailed model of language processing would incorporate comprehension as well as production. With such a model, we may be able to explain mutual priming on prosodic levels and the development of alignment between interlocutors over the course of dialogues. Such alignment is likely to be modulated by non-linguistic factors such as affect. Some interlocutors may serve as a stronger source of contextualization, while others, with whom a speaker does not endeavor to associate, may be deliberately kept further away. An alternative hypothesis would state that all alignment effects are based on mechanistic priming, to the exclusion of non-linguistic factors. This is a topic for future research.

To summarize, our main contribution is to demonstrate that syntactic priming can be explained by a combination of two learning effects: the learning of individual syntactic representations, and the acquisition of links between these syntactic representations and lexical or semantic material. Syntactic priming is neither due to a specialized pre-activation property of information in memory, nor is it due to a single implicit learning effect. Instead, priming emerges from two learning mechanisms that are based on well-established general cognitive principles.

References

Agresti, A. (2002). *Categorical data analysis*. New York, NY: Wiley.

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Quin, Y. (2004). An integrated theory of mind. *Psychological Review*, *111*, 1036-1060.

Bernolet, S., Hartsuiker, R. J., & Pickering, M. J. (2007). Shared syntactic representations in bilinguals: Evidence for the role of word-order repetition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *33*(5), 931–949.

Bock, J. K. (1986). Syntactic persistence in language production. *Cognitive Psychology*, *18*, 355-387.

Bock, J. K., Dell, G. S., Chang, F., & Onishi, K. H. (2007). Persistent structural priming from language comprehension to language production. *Cognition*, *104*(3), 437-458.

Bock, J. K., & Griffin, Z. (2000). The persistence of structural priming: Transient activation or implicit learning? *Journal of Experimental Psychology: General*, *129*, 177-192.

Bock, J. K., & Loebell, H. (1990). Framing sentences. *Cognition*, *35*, 1-39.

Bock, J. K., Loebell, H., & Morey, R. (1992). From conceptual roles to structural relations: Bridging the syntactic cleft. *Psychological Review*, *99*(1), 150–71.

Bod, R. (1992). A computational model of language performance: Data oriented parsing. In *Proceedings of the 14th International Conference on Computational Linguistics* (pp. 855–859). Nantes: Association for Computational Linguistics.

Bond, A. H. (2005). A psycholinguistically and neurolinguistically plausible system-level model of natural-language syntax processing. *Neurocomputing*, *65-66*, 833-841.

Branigan, H. P., Pickering, M. J., & Cleland, A. A. (1999). Syntactic priming in language production: Evidence for rapid decay. *Psychonomic Bulletin and Review*, *6*(4), 635-640.

Branigan, H. P., Pickering, M. J., & Cleland, A. A. (2000). Syntactic co-ordination in dialogue. *Cognition*, *75*, B13-25.

Branigan, H. P., Pickering, M. J., Stewart, A., & McLean, J. F. (2000, 1297-1302). Syntactic priming in spoken production: Linguistic and temporal interference. *Memory and Cognition*, *28*(8).

Bresnan, J., Cueni, A., Nikitina, T., & Baayen, R. H. (2007). Predicting the dative alternation. In G. Boume, I. Kraemer, & J. Zwarts (Eds.), *Cognitive foundations of interpretation* (p. 69-94). Amsterdam: Royal Netherlands Academy of Science.

Chang, F., Bock, J. K., & Goldberg, A. E. (2003). Can thematic roles leave traces of their places? *Cognition*, *99*(1), 29–49.

Chang, F., Dell, G., & Bock, J. K. (2006). Becoming syntactic. *Psychological Review*, *113*(2), 234-272.

Cleland, A. A., & Pickering, M. J. (2003). The use of lexical and syntactic information in language production: Evidence from the priming of noun-phrase structure. *Journal of Memory and Language*, *49*, 214-230.

Collins, M. (1999). *Head-driven statistical models for natural language parsing*. Unpublished doctoral dissertation, University of Pennsylvania.

de Smedt, K., & Kempen, G. (1991). Segment grammar: A formalism for incremental sentence generation. In C. L. Paris, W. R. Swartout, & W. C. Mann (Eds.), *Natural language generation in artificial intelligence and computational linguistics* (p. 329-349). Boston, MA: Kluwer.

Dubey, A., Keller, F., & Sturt, P. (2008). A probabilistic corpus-based model of syntactic parallelism. *Cognition*, *109*(3), 326-344.

Estival, D. (1985). Syntactic priming of the passive in English. *Text*, *5*, 7-21.

Ferreira, F., & Swets, B. (2002). How incremental is language production? Evidence from the production of utterances requiring the computation of arithmetic sums. *Journal of Memory and Language*, *46*, 57-84.

Ferreira, V. S. (2003). The persistence of optional complementizer production: Why saying "that" is not saying "that" at all. *Journal of Memory and Language*, *48*(2), 379-398.

Ferreira, V. S., & Bock, J. K. (2006). The functions of structural priming. *Language and Cognitive Processes*, *21*(7-8), 1011-1029.

Gries, S. T. (2005a). New perspectives on old alternations. In J. E. Cihlar, A. L. Franklin, D. W. Kaiser, & I. Kimbara (Eds.), *Papers from the 39th Regional Meeting of the Chicago Linguistics Society.* Chicago: Chicago Linguistics Society.

Gries, S. T. (2005b). Syntactic priming: A corpus-based approach. *Journal of Psycholinguistic Research*, *34*(4), 365-399.

Grosz, B. J., Joshi, A. K., & Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, *2*(21), 203-225.

Grosz, B. J., & Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, *12*, 175-204.

Halliday, M. (1967). *Intonation and grammar in British English*. The Hague, Netherlands: Mouton.

Hart, B., & Risley, T. R. (1995). *Meaningful differences in the everyday experience of young American children*. Baltimore, MD: Paul H Brookes.

Hartsuiker, R. J., Bernolet, S., Schoonbaert, S., Speybroeck, S., & Vanderelst, D. (2008). Syntactic priming persists while the lexical boost decays: Evidence from written and spoken dialogue. *Journal of Memory and Language*, *58*(2), 214-238.

Hartsuiker, R. J., Kolk, H., & Huiskamp, P. (1999). Priming word order in sentence production. *Quarterly Journal of Experimental Psychology A: Human Experimental Psychology*, *52A*(1), 129-147.

Hartsuiker, R. J., & Kolk, H. H. (1998). Syntactic persistence in Dutch. *Language and Speech*, *41*, 143-184.

Hartsuiker, R. J., & Westenberg, C. (2000). Persistence of word order in written and spoken sentence production. *Cognition*, *75B*, 27-39.

Hockenmaier, J., & Steedman, M. (2007). CCGbank: A corpus of CCG derivations and dependency structures extracted from the Penn treebank. *Computational Linguistics*, *33*(3), 355-396.

Hoffman, B. (1995). Integrating free word order, syntax, and information structure. In *Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics* (p. 245-252). San Francisco, CA: Morgan Kaufmann.

Jaeger, T. F. (2006). *Redundancy and syntactic reduction in spontaneous speech*. Unpublished doctoral dissertation, Stanford University.

Jaeger, T. F., & Snider, N. (2008). Implicit learning and syntactic persistence: Surprisal and cumulativity. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society* (pp. 827–812). Washington: Cognitive Science Society.

Joshi, A. K., Levy, L. S., & Takahashi, M. (1975). Tree adjunct grammars. *Journal of Computing and System Sciences*, *10*(1), 136-163.

Kapatsinski, V. (2006). Towards a single-mechanism account of frequency effects. *LACUS Forum*, *32*, 325–335.

Kaschak, M. P. (2007). Long-term structural priming affects subsequent patterns of language production. *Memory and Cognition*, *35*, 925-937.

Kaschak, M. P., & Borregine, K. L. (2008). Is long-term structural priming affected by patterns of experience with individual verbs? *Journal of Memory and Language*, *58*(3), 862-878.

Kaschak, M. P., Loney, R. A., & Borregine, K. L. (2006). Recent experience affects the strength of structural priming. *Cognition*, *99*, B73-B82.

Krott, A., Schreuder, R., & Baayen, R. H. (2002). Linking elements in Dutch noun-noun compounds: Constituent families as analogical predictors for response latencies. *Brain and Language*, *81*(3), 708–722.

Levelt, W. J. M. (1989). *Speaking: From intention to articulation*. Cambridge, MA: MIT Press.

Levelt, W. J. M., & Kelter, S. (1982). Surface form and memory in question answering. *Cognitive Psychology*, *14*, 78-106.

Lewis, R. L., & Vasishth, S. (2005, May). An activation-based model of sentence processing as skilled memory retrieval. *Cognitive Science*, *29*, 1-45.

Loebell, H., & Bock, K. (2003). Structural priming across languages. *Linguistics*, *41*(5), 791—824.

Malhotra, G. (2009). *Dynamics of structural priming*. Unpublished doctoral dissertation, University of Edinburgh.

Marcus, M., Kim, G., Marcinkiewicz, M., MacIntyre, R., Bies, A., Ferguson, M., et al. (1994). The Penn treebank: Annotating predicate argument structure. In *Proceedings of the ARPA Human Language Technology Workshop*. Plainsboro, NJ: ARPA.

Melinger, A., & Dobel, C. (2005). Lexically-driven syntactic priming. *Cognition*, *98*, B11-B20.

Pickering, M. J., & Branigan, H. P. (1998). The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and Language*, *39*, 633-651.

Pickering, M. J., Branigan, H. P., & McLean, J. F. (2002). Constituent structure is formulated in one stage. *Journal of Memory and Language*, *46*, 586-605.

Pickering, M. J., & Ferreira, V. S. (2008). Structural priming: A critical review. *Psychological Bulletin*, *134*(4), 427–459.

Pickering, M. J., & Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, *27*, 169-225.

Pollard, C., & Sag, I. A. (1994). *Head-driven phrase structure grammar*. Chicago: University of Chicago Press.

Raffray, C. N., & Scheepers, C. (2009). *The lexical boost effect in syntactic priming is not indicative of verb-related syntactic frames.* Poster at the 22nd Annual CUNY Conference on Human Sentence Processing, Davis, CA.

Reitter, D. (2008). *Context effects in language production: Models of syntactic priming in dialogue corpora.* Unpublished doctoral dissertation, University of Edinburgh.

Reitter, D., Hockenmaier, J., & Keller, F. (2006). Priming effects in Combinatory Categorial Grammar. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (p. 308-316). Sydney, Australia: Association for Computational Linguistics.

Reitter, D., & Keller, F. (2007). Against sequence priming: Evidence from constituents and distituents in corpus data. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society (CogSci)* (p. 1421-1426). Nashville, TN: Cognitive Science Society.

Reitter, D., & Moore, J. D. (2007). Predicting success in dialogue. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)* (p. 808-815). Prague, Czech Republic: Association for Computational Linguistics.

Reitter, D., Moore, J. D., & Keller, F. (2006a). *Corpus-based evidence for syntactic priming as rule repetition.* Talk at the 12th Annual Conference on Architectures and Mechanisms for Language Processing. Nijmegen, Netherlands.

Reitter, D., Moore, J. D., & Keller, F. (2006b). Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society* (p. 685-690). Vancouver, Canada: Cognitive Science Society.

Roelofs, A. (1992). A spreading-activation theory of lemma retrieval in speaking. *Cognition*, *42*, 107-142.

Roelofs, A. (1993). Testing a non-decompositional theory of lemma retrieval in speaking: Retrieval of verbs. *Cognition*, *47*, 59-87.

Salamoura, A., & Williams, J. N. (2007). Processing verb argument structure across languages: Evidence for shared representations in the bilingual lexicon. *Applied Psycholinguistics*, *28*, 627-660.

Scheepers, C. (2003). Syntactic priming of relative clause attachments: Persistence of structural configuration in sentence production. *Cognition*, *89*, 179-205.

Schoonbaert, S., Hartsuiker, R. J., & Pickering, M. J. (2007). The representation of lexical and syntactic information in bilinguals: Evidence from syntactic priming. *Journal of Memory and Language*, *56*, 153-171.

Shin, J.-A., & Christianson, K. (2009). Syntactic processing in korean-english bilingual production: Evidence from cross-linguistic structural priming. *Cognition*, *112*(1), 175–180.

Snider, N. (2008). *An exemplar model of syntactic priming*. Unpublished doctoral dissertation, Stanford University.

Snider, N. (2009). Similarity and structural priming. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society* (pp. 815–820). Amsterdam: Cognitive Science Society.

Snider, N., & Jaeger, F. (2009). *Syntax in flux: Structural priming maintains probabilistic representations.* Poster at the 15th Annual Conference on Architectures and Mechanisms of Language Processing, Barcelona.

Steedman, M. (1999). Connectionist sentence processing in perspective. *Cognitive Science*, *23*, 615-634.

Steedman, M. (2000). *The syntactic process*. Cambridge, MA: MIT Press.

Szmrecsanyi, B. (2005). Creatures of habit: A corpus-linguistic analysis of persistence in spoken English. *Corpus Linguistics and Linguistic Theory*, *1*(1), 113-149.

$$
\begin{bmatrix}
\text{Goal buffer} & \\
\text{AGENTSEM} & \text{'policeman'} \\
\text{PREDSEM} & \text{'give'} \\
\text{THEMESEM} & \text{'speeding-fine'} \\
\text{GOALSEM} & \text{'driver'} \\
\text{STATE} & \text{'initial'}
\end{bmatrix}
\begin{array}{l}
\longrightarrow \\
\text{Rule:} \\
\textit{Select-} \\
\textit{Clause-} \\
\textit{Head}
\end{array}
\begin{bmatrix}
\text{Goal buffer} & \\
\text{AGENTSEM} & \text{'policeman'} \\
\text{PREDSEM} & \text{'give'} \\
\text{THEMESEM} & \text{'speeding-fine'} \\
\text{GOALSEM} & \text{'driver'} \\
\text{STATE} & \text{'retr-clause-head'}
\end{bmatrix}
$$

$$
\begin{bmatrix}
\text{Retrieval buffer} & \\
\text{IS-A} & \text{Lexical Form} \\
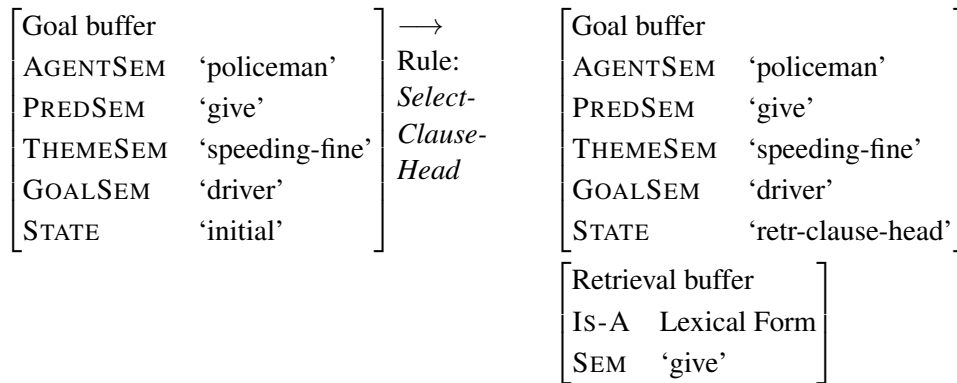\text{SEM} & \text{'give'}
\end{bmatrix}
$$

Figure 8. One step in the generation of a simple sentence, showing the states of the Goal and Retrieval buffers before and after the invocation of the rule *Select-Clause-Head*. This rule (shown in Figure 9) fills the Retrieval buffer to request a lexical form. The outcome of this request occurs later. Invoking the rule takes about 50 ms.

Vosse, T., & Kempen, G. (2000). Syntactic structure assembly in human parsing: A computational model based on competitive inhibition and a lexicalist grammar. *Cognition*, *75*, 105-143.

Weiner, E. J., & Labov, W. (1983). Constraints on the agentless passive. *Journal of Linguistics*, *19*, 29-58.

Wheeldon, L. R., & Lahiri, A. (1997). Prosodic units in speech production. *Journal of Memory and Language*, *37*, 356-381.

Wheeldon, L. R., & Smith, M. C. (2003). Phrase structure priming: A short-lived effect. *Language and Cognitive Processes*, *18*(4), 431-442.

Appendix A. Implementation details

A.1. Buffers and procedural rules in the ACT-R model

The model presented in this paper implements a procedure for language generation from a semantic representation to a surface form (sentence). A number of alternative representations and a range of algorithms could generate sentences and replicate the priming effects; in the following we describe those we used for Simulations 1–3. Simulation 4 was re-implemented in a formalism that allows the underspecification of procedural rules (while maintaining syntactic representations as chunks and the details of cue-based retrieval from declarative memory).

Buffers are a means for the different sub-systems of ACT-R to communicate with one another. The central control mechanism holds its state in the Goal buffer, but interacts with declarative memory. To retrieve a chunk from memory, the Retrieval buffer is filled, and in return, declarative memory augments the buffer with the information stored in the retrieved chunk. In Figure 8, the *Select-Clause-Head* rule uses this buffer to request a lexical form as the verb of the sentence. It tracks its state by changing the STATE attribute to 'retr-clause-head'.

Procedural knowledge is encoded in IF-THEN style production rules. A rule fires when its stated preconditions (IF) are met. Such preconditions check the contents of buffers: most commonly, they test whether a certain value is assigned to an attribute in the attribute-value pairs in a buffer. For instance, the *Select-Clause-Head* rule only fires in situations where the STATE attribute is set to 'initial'.

Once a rule has been selected, it may change the contents of the buffers. Often the THEN
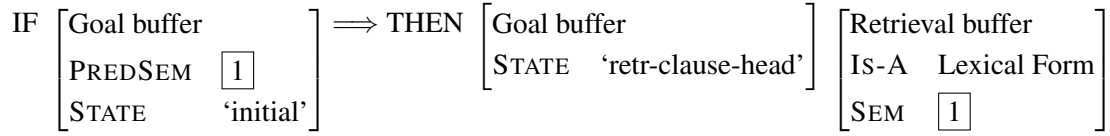
Select-Clause-Head:

$$\text{IF} \begin{bmatrix} \text{Goal buffer} & \\ \text{P{\small RED}S{\small EM}} & \boxed{1} \\ \text{S{\small TATE}} & \text{`initial'} \end{bmatrix} \Longrightarrow \text{THEN} \begin{bmatrix} \text{Goal buffer} & \\ \text{S{\small TATE}} & \text{`retr-clause-head'} \end{bmatrix} \begin{bmatrix} \text{Retrieval buffer} & \\ \text{I{\small S}-A} & \text{Lexical Form} \\ \text{S{\small EM}} & \boxed{1} \end{bmatrix}$$

Figure 9. The *Select-Clause-Head* rule requests a lexical form for the semantic predicate. Once the memory system has delivered the lexical form, another rule will deal with it. In the THEN side of the rule, we only show the changes that apply to the specific buffers. In the Goal buffer, all other information stays intact. By filling the Retrieval buffer, a new request is initiated.

part of the rule uses the Retrieval buffer to request a chunk from declarative memory. Figure 9 demonstrates such a case. The value of the P{\small RED}S{\small EM} attribute contains the name of the semantics of the predicate. It is read from the Goal buffer and copied to the Retrieval buffer to request a lexical form whose semantics match the predicate. (The $\boxed{1}$ indicates a variable local to this rule.) The request to the memory system contains constraints similar to the precondition in a rule: the name or some attribute values of the chunk requested are given. In the present example, all lexical form chunks for a specific semantics are eligible for retrieval.

Once the Retrieval buffer has been filled, the memory subsystem deals with retrieving the chunk. When the chunk has been retrieved, the next rule will match and copy the retrieved chunk into the Goal buffer.

Whereas the IF preconditions of all rules are matched in parallel, the actual execution of a single rule takes time; 50 ms is the default duration assumed in ACT-R. Rule invocation and memory retrieval account for the total sentence production time (without phonological and phonetic processes). The model predicts about four seconds for the production of a simple sentence with a ditransitive verb, which seems broadly plausible.

A.2. *Types of chunks in memory*

Each chunk is a attribute-value structure, with values referring to another chunk (but not containing it). The *type* of a chunk defines a set of attributes that can be contained in chunks of that type. The type information is stored in an I{\small S}-A attribute (see the chunks in Section 3.2 for examples). The following types of chunks are available in our model:

• Syntax chunk: these chunks represent syntactic categories in the CCG sense. For instance, there is a chunk for S\NP, containing the following attribute-value structure:

$$\begin{bmatrix} \text{`intrans'} & \\ \text{I{\small S}-A} & \text{Syntax Chunk} \\ \text{L{\small EFT}} & \text{`S'} \\ \text{C{\small OMBINATOR}} & \text{`\textbackslash'} \\ \text{R{\small IGHT}} & \text{`NP'} \end{bmatrix}$$

The syntax chunk contains the combinatorial components of the category, i.e., the combinators and the categories expected as arguments (an NP to its right) and the resulting category (and S). This is the syntactic category of an intransitive verb such as *laugh*.

• Lexical form: these chunks store lexicon entries. The attribute S{\small EM} refers to the meaning,
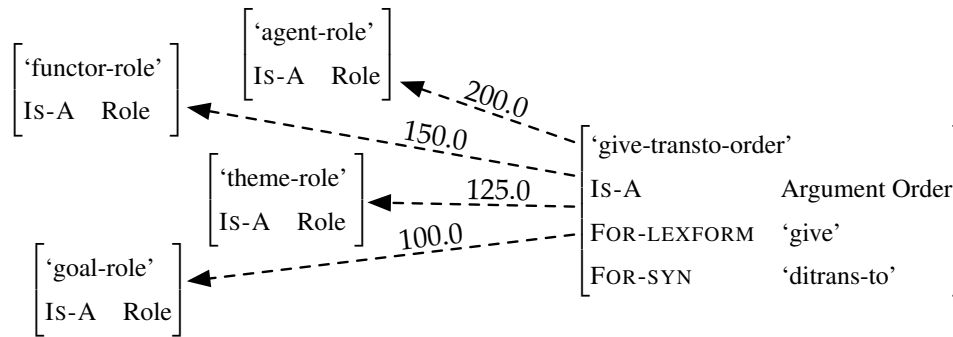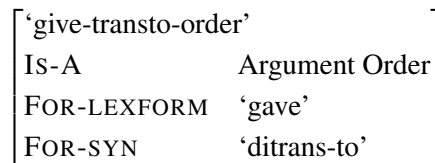
Figure 10. Order chunks define the order of thematic roles by spreading activation to Role chunks: the first role receives the most activation. Order chunks are specific to particular Lexical and Syntactic Forms. (Link values are manually chosen.)

and LEX to its linguistic realization. For instance, in the case of synonyms, we could have two lexical forms containing the LEX values "the doctor" and "the physician", but the same SEM value, 'doc' (referring to some other chunk which is not specified further in our model). As a simplifying assumption, our model stores fully lexicalized noun phrases. Lexical forms specify syntactic variants by means of spreading activation to syntax chunks (see Figure 3).

• Argument order: these chunks provide argument ordering for a given combination of lexical form and syntactic variant. The order they specify controls the sequencing of thematic roles throughout the incremental generation process. Each such chunk refers to a lexical form (attribute FOR-LEXFORM) and a syntax chunk (attribute FOR-SYN).

$$\begin{bmatrix} \text{'give-transto-order'} \\ \text{IS-A} \qquad\qquad \text{Argument Order} \\ \text{FOR-LEXFORM} \quad \text{'gave'} \\ \text{FOR-SYN} \qquad\quad \text{'ditrans-to'} \end{bmatrix}$$

By means of spreading activation to thematic roles (see Figure 10), the argument order chunks specify the order of complements, with most activation being spread to the first thematic role, for instance Agent.

• Role: these are atomic chunks named 'agent-role', 'theme-role', 'goal-role', and 'functor-role'. They receive spreading activation from the argument order chunks. Each Role chunk also spreads inhibitory activation to itself, preventing repeated retrieval. The 'functor' chunk identifies the semantic head of the clause, which is treated as an argument for the purpose of sequencing.

A.3. Spreading activation and associative learning

The acquisition of weighted links between chunks that define spreading activation is not specified in the current ACT-R theory (version 6.0). Clearly, the links that enable activation to spread between chunks must be acquired, and a method for *association learning* has been proposed (Anderson, 1993). In this method, learning occurs whenever a chunk i is requested (event N_i), while another chunk j is *in the context* (event C_j), i.e., it is in a buffer. The *empirical ratio*, defined

as:

$$E_{ji} = \frac{P_e(N_i|C_j)}{P_e(N_i)} \tag{2}$$

determines the positive adjustment of the association between i and j that results from the request of a chunk i when j is in a buffer; P_e is the estimated probability of the event. Equation (2) can be transformed to:

$$E_{ji} = \frac{P_e(N_i, C_j)}{P_e(C_j)P_e(N_i)} \tag{3}$$

which makes it clear that it is the degree of dependence between events N_i and C_j that creates the link between the chunks (formally, this is identical to the information-theoretic quantity of mutual information). This means the strength of learning is moderated by cognitive activity (e.g., number of productions matched) and thus predicts a decay of short-term priming with cognitive activity. (A time-based decay will have a similar effect, as time and production-based cognitive work are correlated.)

A.4. Argument order

Our generation algorithm in Section 3.4 starts by realizing the Agent role. How does the model decide about the order in which arguments are realized? From the model's point of view, there is little difference between the order of syntactic arguments (subject, objects) and the order of thematic roles. The order of arguments is retrieved as a chunk after the head (in our case: the verb) and its syntactic realization is chosen (see Section A.2 for a description of these chunks). Thus, we bind a sequence of arguments (as defined by the syntactic nodes) to a sequence of thematic roles.

Argument order is constrained by several factors. A combination of a lemma and a syntax chunk preselects a number of possible argument orders. Such argument orders are encoded in Argument Order chunks which are sensitive to priming upon retrieval. However, since the argument order is decided only after a lexical form of the head and its syntactic variant have been chosen, priming of argument order cannot influence decisions about lexical forms and syntactic variants. Thus, no priming or long-term priming of argument order is predicted unless lexical or syntactic forms are repeated between prime and target. This is consistent with experimental results that failed to find priming of thematic roles (Bock & Loebell, 1990; Bock, Loebell, & Morey, 1992).[9]

A.5. Initializing model parameters

For the simulations described in Section 4, we left ACT-R parameters at or near their defaults, with base-level learning decay set to 0.5 (default), the activation noise set to 0.4 (common choices are in the range 0.2–0.4) and the maximum associative strength to 50.0. No other parameter values were investigated and no attempt was made to systematically optimize model fit based on ACT-R parameter choices.

Some of the effects may depend on the general frequency of structures and lexical forms. ACT-R models such general exposure as the *base-level activation* of chunks, which strongly influences which chunks are retrieved from memory, and how quickly this can happen. To vary the

---

[9]However, more recent studies (Chang, Bock, & Goldberg, 2003) show that for certain syntactic alternation, there is priming of thematic roles, even if there is no repetition of lexical material. An explanation for this effect may be possible without invoking argument structure, e.g., in terms of priming of more general semantic properties such as animacy and concreteness, as suggested by Chang et al. (2003).

Covariate	β	SE	p_{MCMC}
Simulation 1: Learning and short-term priming			
Intercept	10.99	0.021	< 0.0001***
ln(DIST)	-0.1224	0.0018	< 0.0001***
ln(FREQ)	0.0204	0.0007	< 0.0001***
ln(DIST): ln(FREQ)	0.0026	0.00006	< 0.0001***

Table 2: The linear mixed-effects model fit to data resulting from Simulation 1. ACT-R activation was the dependent variable. P-values were obtained via MCMC sampling.

frequency of syntactic types and lexical forms stored in the lexicon, we initialized their base-levels with data acquired from the Switchboard corpus, which contains spontaneous, conversational speech (North-American English, $684,000$ words without disfluencies). To parametrize the model, the corpus was converted to incremental CCG derivations (see Section 2.3) following the procedure described by Hockenmaier and Steedman (2007) and Reitter, Hockenmaier, and Keller (2006).

From the corpus, we can derive the relative distributions of lexical entries. For a realistic model, we also need an estimate of the amount of language exposure that a speaker receives. In an extensive study of linguistic development and socioeconomic status, Hart and Risley (1995) assessed the linguistic exposure of children aged (up to) 3 years, i.e., the number of words addressed to the children was estimated from a study recording conversations in 42 families. They estimate a range of 10 million to about 35 million words, depending on the social class of the family, increasing linearly with age. Extrapolated, this translates to 50 million to 175 million words comprehended by the age of 15 (when we assume first-language acquisition to be complete). We expect further language exposure (e.g., through mass media) to be more intense. As priming effects are expected to be smaller for high-frequency structures, we chose the conservative assumption of 225 million words (comprehended *and* produced), scaled over 15 years, to set the base-level activations in the model according to ACT-R's base-level learning function. Each exposure is counted as one ACT-R chunk presentation for the purpose of activation calculation.

The distribution of syntactic choices given a lexical form is also estimated from the corpus. The link strengths from lexical forms to syntax chunks (as shown in Figure 3) result from the frequencies of syntactic forms for given lexical choices. The current ACT-R framework does not provide an account of associative learning; however, for Simulations 1–3, we made the assumption that link strength is correlated with (observable) conditional probabilities. Each strength is estimated as $0.5\phi(1+\hat{p}(syn|lex))$ for a given lexical form *lex* and a syntactic category *syn* in the corpus (where $\hat{p}$ is the maximum likelihood estimate of the probability, and $\phi = 75$ is a constant norming coefficient to determine the relative influence of link strengths and base-level activation in chunk retrieval). Thus association is derived from the conditional probability of a particular syntactic realization given the lexical form.

Appendix B. Details of regression models

The details of the linear mixed-effects models fitted for Simulation 1 is given in Table 2. Table 3 gives the details for the mixed effects models in Simulations 2 and 3; Table 4 gives the same information for Experiment 1.

Covariate	β	SE	z	$p(> |z|)$
Simulation 2: Priming in DO/PO production				
Intercept	−0.8243	0.1537	−5.364	< 0.0001***
ln(DIST)	0.0877	0.2355	0.372	0.71
Simulation 3: Cumulative priming for prepositional objects (PO)				
Intercept	0.0235	0.0579	0.406	0.68
#PRIMES	0.0243	0.0084	2.893	< 0.005**
Simulation 3: Cumulative priming for double objects (DO)				
Intercept	1.2756	0.0700	18.23	< 0.0001***
#PRIMES	0.0095	0.0101	0.94	0.347

Table 3: The logistic regression models fit to data resulting from Simulations 2 and 3. Repetition was the dependent variable (logit-link transformed). The same significance levels for variables of interest were also obtained via MCMC sampling from the corresponding linear mixed-effect models. Predictors were centered, collinearity was evaluated using correlation coefficients; no substantial collinearity remained for variables of interest (see main text).

Covariate	β	SE	z	$p(> |z|)$
Experiment 1: General lexical boost				
Intercept	−0.7403	0.0107	−69.38	< 0.0001***
log(RULEFREQUENCY)	0.7315	0.0069	106.32	< 0.0001***
ln(DIST)	−0.1940	0.0127	−15.24	< 0.0001***
HEADREPETITION	2.8668	0.0245	117.25	< 0.0001***
WORDREPETITIONLOGIT	0.5183	0.0089	58.29	< 0.0001***
log(RULEFREQUENCY):ln(DIST)	0.0740	0.0088	8.40	< 0.0001***
ln(DIST):HEADREPETITION	−0.0112	0.0324	−0.35	0.729
ln(DIST):WORDREPETITIONLOGIT	−0.1741	0.0122	−14.31	< 0.0001***

Table 4: The logistic regression model fit for the corpus data in Experiment 1. Repetition was the dependent variable (logit-link transformed). Significance, centering, and collinearity as in Table 3.