Complexity Metrics in an Incremental Right-corner Parser

Stephen Wu **Asaf Bachrach**[†] **Carlos Cardenas**[*] **William Schuler**[°]
Department of Computer Science, University of Minnesota
[†] Unit de Neuroimagerie Cognitive INSERM-CEA
[*] Department of Brain & Cognitive Sciences, Massachussetts Institute of Technology
[°] University of Minnesota and The Ohio State University
swu@cs.umn.edu [†]asaf@mit.edu [*]cardenas@mit.edu [°]schuler@ling.ohio-state.edu

Abstract

Hierarchical HMM (HHMM) parsers make promising cognitive models: while they use a bounded model of working memory and pursue incremental hypotheses in parallel, they still achieve parsing accuracies competitive with chart-based techniques. This paper aims to validate that a right-corner HHMM parser is also able to produce *complexity metrics*, which quantify a reader's incremental difficulty in understanding a sentence. Besides defining standard metrics in the HHMM framework, a new metric, *embedding difference*, is also proposed, which tests the hypothesis that HHMM store elements represents syntactic working memory. Results show that HHMM surprisal outperforms all other evaluated metrics in predicting reading times, and that embedding difference makes a significant, independent contribution.

1 Introduction

Since the introduction of a parser-based calculation for surprisal by Hale (2001), statistical techniques have been become common as models of reading difficulty and linguistic complexity. Surprisal has received a lot of attention in recent literature due to nice mathematical properties (Levy, 2008) and predictive ability on eye-tracking movements (Demberg and Keller, 2008; Boston et al., 2008a). Many other complexity metrics have been suggested as mutually contributing to reading difficulty; for example, entropy reduction (Hale, 2006), bigram probabilities (McDonald and Shillcock, 2003), and split-syntactic/lexical versions of other metrics (Roark et al., 2009).

A parser-derived complexity metric such as surprisal can only be as good (empirically) as the model of language from which it derives (Frank, 2009). Ideally, a psychologically-plausible language model would produce a surprisal that would correlate better with linguistic complexity. Therefore, the specification of how to encode a syntactic language model is of utmost importance to the quality of the metric.

However, it is difficult to quantify linguistic complexity and reading difficulty. The two commonly-used empirical quantifications of reading difficulty are eye-tracking measurements and word-by-word reading times; this paper uses reading times to find the predictiveness of several parser-derived complexity metrics. Various factors (i.e., from syntax, semantics, discourse) are likely necessary for a full accounting of linguistic complexity, so current computational models (with some exceptions) narrow the scope to syntactic or lexical complexity.

Three complexity metrics will be calculated in a Hierarchical Hidden Markov Model (HHMM) parser that recognizes trees in right-corner form (the left-right dual of left-corner form). This type of parser performs competitively on standard parsing tasks (Schuler et al., 2010); also, it reflects plausible accounts of human language processing as incremental (Tanenhaus et al., 1995; Brants and Crocker, 2000), as considering hypotheses probabilistically in parallel (Dahan and Gaskell, 2007), as bounding memory usage to short-term memory limits (Cowan, 2001), and as requiring more memory storage for center-embedding structures than for right- or left-branching ones (Chomsky and Miller, 1963; Gibson, 1998). Also, unlike most other parsers, this parser preserves the *arc-eager/arc-standard* ambiguity of Abney and John-

son (1991). Typical parsing strategies are arc-standard, keeping all right-descendants open for subsequent attachment; but since there can be an unbounded number of such open constituents, this assumption is not compatible with simple models of bounded memory. A consistently arc-eager strategy acknowledges memory bounds, but yields dead-end parses. Both analyses are considered in right-corner HHMM parsing.

The purpose of this paper is to determine whether the language model defined by the HHMM parser can also predict reading times — it would be strange if a psychologically plausible model did not also produce viable complexity metrics. In the course of showing that the HHMM parser does, in fact, predict reading times, we will define surprisal and entropy reduction in the HHMM parser, and introduce a third metric called *embedding difference*.

Gibson (1998; 2000) hypothesized two types of syntactic processing costs: *integration cost*, in which incremental input is combined with existing structures; and *memory cost*, where unfinished syntactic constructions may incur some short-term memory usage. HHMM surprisal and entropy reduction may be considered forms of integration cost. Though typical PCFG surprisal has been considered a forward-looking metric (Demberg and Keller, 2008), the incremental nature of the right-corner transform causes surprisal and entropy reduction in the HHMM parser to measure the likelihood of grammatical structures that were hypothesized before evidence was observed for them. Therefore, these HHMM metrics resemble an integration cost encompassing both backward-looking and forward-looking information.

On the other hand, embedding difference is designed to model the cost of storing center-embedded structures in working memory. Chen, Gibson, and Wolf (2005) showed that sentences requiring more syntactic memory during sentence processing increased reading times, and it is widely understood that center-embedding incurs significant syntactic processing costs (Miller and Chomsky, 1963; Gibson, 1998). Thus, we would expect for the usage of the center-embedding memory store in an HHMM parser to correlate with reading times (and therefore linguistic complexity).

The HHMM parser processes syntactic constructs using a bounded number of store states, defined to represent short-term memory elements; additional states are utilized whenever center-embedded syntactic structures are present. Similar models such as Crocker and Brants (2000) implicitly allow an infinite memory size, but Schuler et al. (2008; 2010) showed that a right-corner HHMM parser can parse most sentences in English with 4 or fewer center-embedded-depth levels. This behavior is similar to the hypothesized size of a human short-term memory store (Cowan, 2001). A positive result in predicting reading times will lend additional validity to the claim that the HHMM parser's bounded memory corresponds to bounded memory in human sentence processing.

The rest of this paper is organized as follows: Section 2 defines the language model of the HHMM parser, including definitions of the three complexity metrics. The methodology for evaluating the complexity metrics is described in Section 3, with actual results in Section 4. Further discussion on results, and comparisons to other work, are in Section 5.

2 Parsing Model

This section describes an incremental parser in which surprisal and entropy reduction are simple calculations (Section 2.1). The parser uses a Hierarchical Hidden Markov Model (Section 2.2) and recognizes trees in a right-corner form (Section 2.3 and 2.4). The new complexity metric, embedding difference (Section 2.5), is a natural consequence of this HHMM definition. The model is equivalent to previous HHMM parsers (Schuler, 2009), but reorganized into 5 cases to clarify the right-corner structure of the parsed sentences.

2.1 Surprisal and Entropy in HMMs

Hidden Markov Models (HMMs) probabilistically connect sequences of observed states o_t and hidden states q_t at corresponding time steps t. In parsing, observed states are words; hidden states can be a conglomerate state of linguistic information, here taken to be syntactic.

The HMM is an incremental, time-series structure, so one of its by-products is the *prefix probability*, which will be used to calculate surprisal. This is the probability that that words $o_{1..t}$ have been observed at time t, regardless of which syntactic states $q_{1..t}$ produced them. Bayes' Law and Markov independence assumptions allow this to

be calculated from two generative probability distributions.[1]

$$\mathsf{Pre}(o_{1..t}) = \sum_{q_{1..t}} \mathsf{P}(o_{1..t}\, q_{1..t}) \quad (1)$$

$$\stackrel{\text{def}}{=} \sum_{q_{1..t}} \prod_{\tau=1}^{t} \mathsf{P}_{\Theta_\text{A}}(q_\tau \mid q_{\tau-1}) \cdot \mathsf{P}_{\Theta_\text{B}}(o_\tau \mid q_\tau) \quad (2)$$

Here, probabilities arise from a *Transition Model* (Θ_A) between hidden states and an *Observation Model* (Θ_B) that generates an observed state from a hidden state. These models are so termed for historical reasons (Rabiner, 1990).

Surprisal (Hale, 2001) is then a straightforward calculation from the prefix probability.

$$\text{Surprisal}(t) = \log_2 \frac{\mathsf{Pre}(o_{1..t-1})}{\mathsf{Pre}(o_{1..t})} \quad (3)$$

This framing of prefix probability and surprisal in a time-series model is equivalent to Hale's (2001; 2006), assuming that $q_{1..t} \in \mathcal{D}_t$, i.e., that the syntactic states we are considering form derivations $\mathcal{D}_t$, or partial trees, consistent with the observed words. We will see that this is the case for our parser in Sections 2.2–2.4.

Entropy is a measure of uncertainty, defined as $H(x) = -\mathsf{P}(x) \log_2 \mathsf{P}(x)$. Now, the entropy H_t of a t-word string $o_{1..t}$ in an HMM can be written:

$$H_t = \sum_{q_{1..t}} \mathsf{P}(q_{1..t}\, o_{1..t}) \log_2 \mathsf{P}(q_{1..t}\, o_{1..t}) \quad (4)$$

and *entropy reduction* (Hale, 2003; Hale, 2006) at the t^th word is then

$$\text{ER}(o_t) = \max(0, H_{t-1} - H_t) \quad (5)$$

Both of these metrics fall out naturally from the time-series representation of the language model. The third complexity metric, embedding difference, will be discussed after additional background in Section 2.5.

In the implementation of an HMM, candidate states at a given time q_t are kept in a trellis, with step-by-step backpointers to the highest-probability $q_{1..t-1}$.[2] Also, the best q_t are often kept in a beam $\mathcal{B}_t$, discarding low-probability states. This mitigates the problems of large state spaces (e.g., that of all possible grammatical derivations). Since beams have been shown to perform well (Brants and Crocker, 2000; Roark, 2001; Boston et al., 2008b), complexity metrics in this paper are calculated on a beam rather than over all (unbounded) possible derivations $\mathcal{D}_t$. The equations above, then, will replace the assumption $q_{1..t} \in \mathcal{D}_t$ with $q_t \in \mathcal{B}_t$.

[1] Technically, a prior distribution over hidden states, $\mathsf{P}(q_0)$, is necessary. This q_0 is factored and taken to be a deterministic constant, and is therefore unimportant as a probability model.

[2] Typical tasks in an HMM include finding the most likely sequence via the Viterbi algorithm, which stores these backpointers to maximum-probability previous states and can uniquely find the most likely sequence.

2.2 Hierarchical Hidden Markov Models

Hidden states q can have internal structure; in Hierarchical HMMs (Fine et al., 1998; Murphy and Paskin, 2001), this internal structure will be used to represent syntax trees and looks like several HMMs stacked on top of each other. As such, q_t is factored into sequences of depth-specific variables — one for each of D levels in the HMM hierarchy. In addition, an intermediate variable f_t is introduced to interface between the levels.

$$q_t \stackrel{\text{def}}{=} \langle q_t^1 \ldots q_t^D \rangle \quad (6)$$

$$f_t \stackrel{\text{def}}{=} \langle f_t^1 \ldots f_t^D \rangle \quad (7)$$

Transition probabilities $\mathsf{P}_{\Theta_\text{A}}(q_t \mid q_{t-1})$ over complex hidden states q_t are calculated in two phases:

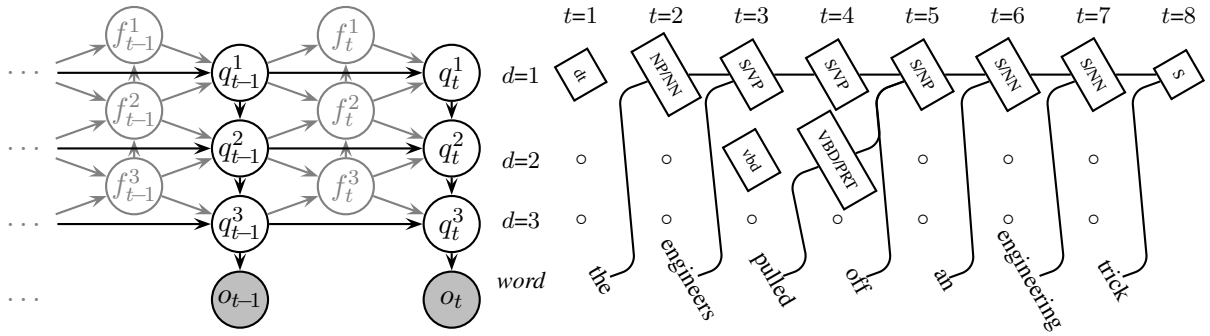
- *Reduce phase.* Yields an intermediate state f_t, in which component HMMs may terminate. This f_t tells "higher" HMMs to hold over their information if "lower" levels are in operation at any time step t, and tells lower HMMs to signal when they're done.
- *Shift phase.* Yields a modeled hidden state q_t, in which unterminated HMMs transition, and terminated HMMs are re-initialized from their parent HMMs.

Each phase is factored according to *level-specific* reduce and shift models, Θ_F and Θ_Q:

$$\mathsf{P}_{\Theta_\text{A}}(q_t | q_{t-1}) = \sum_{f_t} \mathsf{P}(f_t | q_{t-1}) \cdot \mathsf{P}(q_t | f_t\, q_{t-1}) \quad (8)$$

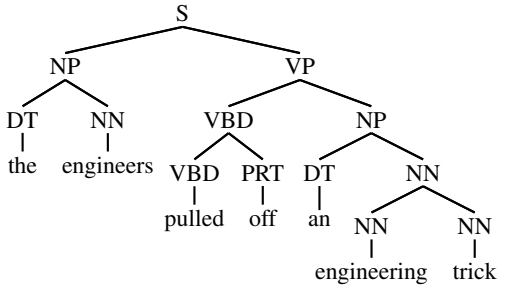
$$\stackrel{\text{def}}{=} \sum_{f_t^{1..D}} \prod_{d=1}^{D} \mathsf{P}_{\Theta_\text{F}}(f_t^d | f_t^{d+1} q_{t-1}^d q_{t-1}^{d-1}) \cdot \mathsf{P}_{\Theta_\text{Q}}(q_t^d | f_t^{d+1} f_t^d\, q_{t-1}^d q_t^{d-1}) \quad (9)$$

with f_t^{D+1} and q_t^0 defined as constants. Note that only q_t is present at the end of the probability calculation. In step t, f_{t-1} will be unused, so the marginalization of Equation 9 does not lose any information.

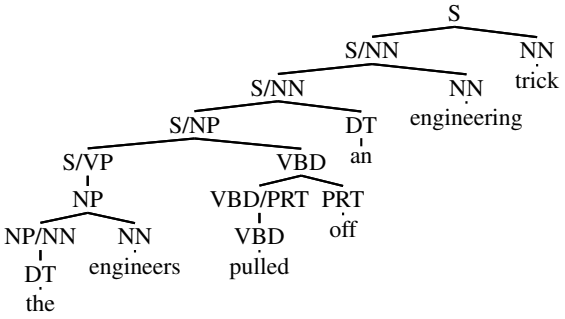


(a) Dependency structure in the HHMM parser. Conditional probabilities at a node are dependent on incoming arcs.

(b) HHMM parser as a store whose elements at each time step are listed vertically, showing a good hypothesis on a sample sentence out of many kept in parallel. Variables corresponding to q_t^d are shown.



(c) A sample sentence in CNF.



(d) The right-corner transformed version of (c).

Figure 1: Various graphical representations of HHMM parser operation. (a) shows probabilistic dependencies. (b) considers the q_t^d store to be incremental syntactic information. (c)–(d) demonstrate the right-corner transform, similar to a left-to-right traversal of (c). In 'NP/NN' we say that NP is the *active* constituent and NN is the *awaited*.

The Observation Model Θ_{B} is comparatively much simpler. It is only dependent on the syntactic state at D (or the deepest active HHMM level).

$$\mathsf{P}_{\Theta_{\mathrm{B}}}(o_t \mid q_t) \stackrel{\text{def}}{=} \mathsf{P}(o_t \mid q_t^D) \tag{10}$$

Figure 1(a) gives a schematic of the dependency structure of Equations 8–10 for $D = 3$. Evaluations in this paper are done with $D = 4$, following the results of Schuler, et al. (2008).

2.3 Parsing right-corner trees

In this HHMM formulation, states and dependencies are optimized for parsing right-corner trees (Schuler et al., 2008; Schuler et al., 2010). A sample transformation between CNF and right-corner trees is in Figures 1(c)–1(d).

Figure 1(b) shows the corresponding store-element interpretation[3] of the right corner tree in 1(d). These can be used as a case study to see what kind of operations need to occur in an HHMM when parsing right-corner trees. There is one unique set of HHMM state values for each tree, so the operations can be seen on either the tree or the store elements.

At each time step t, a certain number of elements (maximum D) are kept in memory, i.e., in the store. New words are observed input, and the bottom occupied element (the "frontier" of the store) is the context; together, they determine what the store will look like at $t{+}1$. We can characterize the types of store-element changes by when they happen in Figures 1(b) and 1(d):

Cross-level Expansion (CLE). Occupies a new store element at a given time step. For example, at $t = 1$, a new store element is occupied which can interact with the observed word, "the." At $t = 3$, an expansion occupies the second store element.

In-level Reduction (ILR). Completes an active constituent that is a unary child in the right-corner tree; always accompanied by an in-level expansion. At $t = 2$, "engineers" completes the active NP constituent; however, the

[3] This is technically a pushdown automoton (PDA), where the store is limited to D elements. When referring to directions (e.g., up, down), PDAs are typically described opposite of the one in Figure 1(b); here, we push "up" instead of down.

level is not yet complete since the NP is along the left-branching trunk of the tree.

In-level Expansion (ILE). Starts a new active constituent at an already-occupied store element; always follows an in-level reduction. With the NP complete in $t=2$, a new active constituent S is produced at $t=3$.

In-level Transition (ILT). Transitions the store to a new state in the next time step at the same level, where the awaited constituent changes and the active constituent remains the same. This describes each of the steps from $t=4$ to $t=8$ at $d=1$.

Cross-level Reduction (CLR). Vacates a store element on seeing a complete active constituent. This occurs after $t = 4$; "off" completes the active (at depth 2) VBD constituent, and vacates store element 2. This is accompanied with an in-level transition at depth 1, producing the store at $t=5$. It should be noted that with some probability, completing the active constituent does not vacate the store element, and the in-level reduction case would have to be invoked.

The in-level/cross-level ambiguity occurs in the expansion as well as the reduction, similar to Abney and Johnson's arc-eager/arc-standard composition strategies (1991). At $t=3$, another possible hypothesis would be to remain on store element 1 using an ILE instead of a CLE. The HHMM parser, unlike most other parsers, will preserve this in-level/cross-level ambiguity by considering both hypotheses in parallel.

2.4 Reduce and Shift Models

With the understanding of what operations need to occur, a formal definition of the language model is in order. Let us begin with the relevant variables.

A shift variable q_t^d at depth d and time step t is a syntactic state that must represent the active and awaited constituents of right-corner form:

$$q_t^d \stackrel{\text{def}}{=} \langle g^{\text{A}}_{q_t^d}, g^{\text{W}}_{q_t^d} \rangle \tag{11}$$

e.g., in Figure 1(b), q_2^1=⟨NP,NN⟩=NP/NN. Each g is a constituent from the pre-right-corner grammar, G.

Reduce variables f are then enlisted to ensure that in-level and cross-level operations are correct.

$$f_t^d \stackrel{\text{def}}{=} \langle k_{f_t^d}, g_{f_t^d} \rangle \tag{12}$$

First, $k_{f_t^d}$ is a switching variable that differentiates between ILT, CLE/CLR, and ILE/ILR. This switching is the most important aspect of f_t^d, so regardless of what $g_{f_t^d}$ is, we will use:

- $f_t^d \in F_0$ when $k_{f_t^d} = \mathbf{0}$, (ILT/no-op)
- $f_t^d \in F_1$ when $k_{f_t^d} = \mathbf{1}$, (CLE/CLR)
- $f_t^d \in F_G$ when $k_{f_t^d} \in G$. (ILE/ILR)

Then, $g_{f_t^d}$ is used to keep track of a completely-recognized constituent whenever a reduction occurs (ILR or CLR). For example, in Figure 1(b), after time step 2, an NP has been completely recognized and precipitates an ILR. The NP gets stored in $g_{f_3^1}$ for use in the ensuing ILE instead of appearing in the store-elements.

This leads us to a specification of the reduce and shift probability models. The reduce step happens first at each time step. True to its name, the reduce step handles in-level and cross-level reductions (the second and third case below):

$$\mathsf{P}_{\Theta_{\text{F}}}(f_t^d \mid f_t^{d+1} q_{t-1}^d q_{t-1}^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_t^{d+1} \notin F_G & : [\![f_t^d = \mathbf{0}]\!] \\ \text{if } f_t^{d+1} \in F_G, f_t^d \in F_1 & : \tilde{\mathsf{P}}_{\Theta_{\text{F-ILR},d}}(f_t^d \mid q_{t-1}^d \, q_{t-1}^{d-1}) \\ \text{if } f_t^{d+1} \in F_G, f_t^d \in F_G & : \tilde{\mathsf{P}}_{\Theta_{\text{F-CLR},d}}(f_t^d \mid q_{t-1}^d \, q_{t-1}^{d-1}) \end{cases} \tag{13}$$

with edge cases q_t^0 and f_t^{D+1} defined as appropriate constants. The first case is just store-element maintenance, in which the variable is not on the "frontier" and therefore inactive.

Examining $\Theta_{\text{F-ILR},d}$ and $\Theta_{\text{F-CLR},d}$, we see that the produced f_t^d variables are also used in the "if" statement. These models can be thought of as picking out a f_t^d first, finding the matching case, then applying the probability models that matches. These models are actually two parts of the same model when learned from trees.

Probabilities in the shift step are also split into cases based on the reduce variables. More maintenance operations (first case) accompany transitions producing new awaited constituents (second case below) and expansions producing new active constituents (third and fourth case):

$$\mathsf{P}_{\Theta_{\text{Q}}}(q_t^d \mid f_t^{d+1} f_t^d \, q_{t-1}^d q_t^{d-1}) \stackrel{\text{def}}{=} \begin{cases} \text{if } f_t^{d+1} \notin F_G & : [\![q_t^d = q_{t-1}^d]\!] \\ \text{if } f_t^{d+1} \in F_G, \, f_t^d \in F_0 & : \tilde{\mathsf{P}}_{\Theta_{\text{Q-ILT},d}}(q_t^d \mid f_t^{d+1} \, q_{t-1}^d \, q_t^{d-1}) \\ \text{if } f_t^{d+1} \in F_G, \, f_t^d \in F_1 & : \tilde{\mathsf{P}}_{\Theta_{\text{Q-ILE},d}}(q_t^d \mid f_t^d \, q_{t-1}^d \, q_t^{d-1}) \\ \text{if } f_t^{d+1} \in F_G, \, f_t^d \in F_G & : \tilde{\mathsf{P}}_{\Theta_{\text{Q-CLE},d}}(q_t^d \mid q_t^{d-1}) \end{cases} \tag{14}$$

FACTOR	DESCRIPTION	EXPECTED
Word order in narrative	For each story, words were indexed. Subjects would tend to read faster later in a story.	negative slope
Reciprocal length	Log of the reciprocal of the number of letters in each word. A decrease in the reciprocal (increase in length) might mean longer reading times.	positive slope
Unigram frequency	A log-transformed empirical count of word occurrences in the Brown Corpus section of the Penn Treebank. Higher frequency should indicate shorter reading times.	negative slope
Bigram probability	A log-transformed empirical count of two-successive-word occurrences, with Good-Turing smoothing on words occuring less than 10 times.	negative slope
Embedding difference	Amount of change in HHMM weighted-average embedding depth. Hypothesized to increase with larger working memory requirements, which predict longer reading times.	positive slope
Entropy reduction	Amount of decrease in the HHMM's uncertainty about the sentence. Larger reductions in uncertainty are hypothesized to take longer.	positive slope
Surprisal	"Surprise value" of a word in the HHMM parser; models were trained on the Wall Street Journal, sections 02–21. More surprising words may take longer to read.	positive slope

Table 1: A list of factors hypothesized to contribute to reading times. All data was mean-centered.

A final note: the notation $\tilde{\mathsf{P}}_\Theta(\cdot \mid \cdot)$ has been used to indicate probability models that are empirical, trained directly from frequency counts of right-corner transformed trees in a large corpus. Alternatively, a standard PCFG could be trained on a corpus (or hand-specified), and then the grammar itself can be right-corner transformed (Schuler, 2009).

Taken together, Equations 11–14 define the probabilistic structure of the HHMM for parsing right-corner trees.

2.5 Embedding difference in the HHMM

It should be clear from Figure 1 that at any time step while parsing depth-bounded right-corner trees, the candidate hidden state q_t will have a "frontier" depth $d(q_t)$. At time t, the beam of possible hidden states q_t stores the syntactic state (and a backpointer) along with its probability, $\mathsf{P}(o_{1..t}\, q_{1..t})$. The *average embedding depth* at a time step is then

$$\mu_{\text{EMB}}(o_{1..t}) = \sum_{q_t \in \mathcal{B}_t} d(q_t) \cdot \frac{\mathsf{P}(o_{1..t}\, q_{1..t})}{\sum_{q'_t \in \mathcal{B}_t} \mathsf{P}(o_{1..t}\, q'_{1..t})} \tag{15}$$

where we have directly used the beam notation. The *embedding difference* metric is:

$$\text{EmbDiff}(o_{1..t}) = \mu_{\text{EMB}}(o_{1..t}) - \mu_{\text{EMB}}(o_{1..t-1}) \tag{16}$$

There is a strong computational correspondence between this definition of embedding difference and the previous definition of surprisal. To see this, we rewrite Equations 1 and 3:

$$\mathsf{Pre}(o_{1..t}) = \sum_{q_t \in \mathcal{B}_t} \mathsf{P}(o_{1..t}\, q_{1..t}) \tag{1'}$$

$$\text{Surprisal}(t) = \log_2 \mathsf{Pre}(o_{1..t-1}) - \log_2 \mathsf{Pre}(o_{1..t}) \tag{3'}$$

Both surprisal and embedding difference include summations over the elements of the beam, and are calculated as a difference between previous and current beam states.

Most differences between these metrics are relatively inconsequential. For example, the difference in order of subtraction only assures that a positive correlation with reading times is expected. Also, the presence of a logarithm is relatively minor. Embedding difference weighs the probabilities with center-embedding depths and then normalizes the values; since the measure is a weighted average of embedding depths rather than a probability distribution, μ_{EMB} is not always less than 1 and the correspondence with Kullback-Leibler divergence (Levy, 2008) does not hold, so it does not make sense to take the logs.

Therefore, the inclusion of the embedding depth, $d(q_t)$, is the only significant difference between the two metrics. The result is a metric that, despite numerical correspondence to surprisal, models the HHMM's hypotheses about memory cost.

3 Evaluation

Surprisal, entropy reduction, and embedding difference from the HHMM parser were evaluated against a full array of factors (Table 1) on a corpus of word-by-word reading times using a linear mixed-effects model.

The corpus of reading times for 23 native English speakers was collected on a set of four narratives (Bachrach et al., 2009), each composed of sentences that were syntactically complex but constructed to appear relatively natural. Using Linger 2.88, words appeared one-by-one on the screen, and required a button-press in order to advance; they were displayed in lines with 11.5 words on average.

Following Roark et al.'s (2009) work on the same corpus, reading times above 1500 ms (for diverted attention) or below 150 ms (for button presses planned before the word appeared) were discarded. In addition, the first and last word of each line on the screen were removed; this left 2926 words out of 3540 words in the corpus.

For some tests, a division between open- and closed-class words was made, with 1450 and 1476 words, respectively. Closed-class words (e.g., determiners or auxiliary verbs) usually play some kind of syntactic function in a sentence; our evaluations used Roark et al.'s list of stop words. Open class words (e.g., nouns and other verbs) more commonly include new words. Thus, one may expect reading times to differ for these two types of words.

Linear mixed-effect regression analysis was used on this data; this entails a set of *fixed effects* and another of *random effects*. Reading times y were modeled as a linear combination of factors x, listed in Table 1 (fixed effects); some random variation in the corpus might also be explained by groupings according to subject i, word j, or sentence k (random effects).

$$y_{ijk} = \beta_0 + \sum_{\ell=1}^{m} \beta_\ell x_{ijk\ell} + b_i + b_j + b_k + \varepsilon \qquad (17)$$

This equation is solved for each of m fixed-effect coefficients β with a measure of confidence (t-value $= \hat{\beta}/\mathrm{SE}(\hat{\beta})$, where SE is the standard error). β_0 is the standard intercept to be estimated along with the rest of the coefficients, to adjust for affine relationships between the dependent and independent variables. We report factors as statistically significant contributors to reading time if the absolute value of the t-value is greater than 2.

Two more types of comparisons will be made to see the significance of factors. First, a model of data with the full list of factors can be compared to a model with a subset of those factors. This is done with a likelihood ratio test, producing (for mixed-effects models) a χ^2_1 value and corresponding probability that the smaller model could have produced the same estimates as the larger model. A lower probability indicates that the additional factors in the larger model are significant.

Second, models with different fixed effects can be compared to each other through various information criteria; these trade off between having a more explanatory model vs. a simpler model, and can be calculated on any model. Here, we use Akaike's Information Criterion (AIC), where lower values indicate better models.

All these statistics were calculated in R, using the `lme4` package (Bates et al., 2008).

4 Results

Using the full list of factors in Table 1, fixed-effect coefficients were estimated in Table 2. Fitting the best model by AIC would actually prune away some of the factors as relatively insignificant, but these smaller models largely accord with the significance values in the table and are therefore not presented.

The first data column shows the regression on all data; the second and third columns divide the data into open and closed classes, because an evaluation (not reported in detail here) showed statistically significant interactions between word class and 3 of the predictors. Additionally, this facilitates comparison with Roark et al. (2009), who make the same division.

Out of the non-parser-based metrics, word order and bigram probability are statistically significant regardless of the data subset; though reciprocal length and unigram frequency do not reach significance here, likelihood ratio tests (not shown) confirm that they contribute to the model as a whole. It can be seen that nearly all the slopes have been estimated with signs as expected, with the exception of reciprocal length (which is not statistically significant).

Most notably, HHMM surprisal is seen here to be a standout predictive measure for reading times regardless of word class. If the HHMM parser is a good psycholinguistic model, we would expect it to at least produce a viable surprisal metric, and Table 2 attests that this is indeed the case. Though it seems to be less predictive of open classes, a surprisal-only model has the best AIC (-7804) out of any open-class model. Considering the AIC on the full data, the worst model *with* surprisal

	FULL DATA			OPEN CLASS			CLOSED CLASS		
	Coefficient	Std. Err.	t-value	Coefficient	Std. Err.	t-value	Coefficient	Std. Err.	t-value
(Intcpt)	$-9.340\cdot10^{-3}$	$5.347\cdot10^{-2}$	-0.175	$-1.237\cdot10^{-2}$	$5.217\cdot10^{-2}$	-0.237	$-6.295\cdot10^{-2}$	$7.930\cdot10^{-2}$	-0.794
order	$-3.746\cdot10^{-5}$	$7.808\cdot10^{-6}$	-4.797*	$-3.697\cdot10^{-5}$	$8.002\cdot10^{-6}$	-4.621*	$-3.748\cdot10^{-5}$	$8.854\cdot10^{-6}$	-4.232*
rlength	$-2.002\cdot10^{-2}$	$1.635\cdot10^{-2}$	-1.225	$9.849\cdot10^{-3}$	$1.779\cdot10^{-2}$	0.554	$-2.839\cdot10^{-2}$	$3.283\cdot10^{-2}$	-0.865
unigrm	$-8.090\cdot10^{-2}$	$3.690\cdot10^{-1}$	-0.219	$-1.047\cdot10^{-1}$	$2.681\cdot10^{-1}$	-0.391	$-3.847\cdot10^{+0}$	$5.976\cdot10^{+0}$	-0.644
bigrm	$-2.074\cdot10^{+0}$	$8.132\cdot10^{-1}$	-2.551*	$-2.615\cdot10^{+0}$	$8.050\cdot10^{-1}$	-3.248*	$-5.052\cdot10^{+1}$	$1.910\cdot10^{+1}$	-2.645*
embdiff	$9.390\cdot10^{-3}$	$3.268\cdot10^{-3}$	2.873*	$2.432\cdot10^{-3}$	$4.512\cdot10^{-3}$	0.539	$1.598\cdot10^{-2}$	$5.185\cdot10^{-3}$	3.082*
etrpyrd	$2.753\cdot10^{-2}$	$6.792\cdot10^{-3}$	4.052*	$6.634\cdot10^{-4}$	$1.048\cdot10^{-2}$	0.063	$4.938\cdot10^{-2}$	$1.017\cdot10^{-2}$	4.857*
srprsl	$3.950\cdot10^{-3}$	$3.452\cdot10^{-4}$	11.442*	$2.892\cdot10^{-3}$	$4.601\cdot10^{-4}$	6.285*	$5.201\cdot10^{-3}$	$5.601\cdot10^{-4}$	9.286*

Table 2: Results of linear mixed-effect modeling. Significance (indicated by *) is reported at $p < 0.05$.

	(Intr)	order	rlngth	ungrm	bigrm	emdiff	entrpy
order	.000						
rlength	-.006	-.003					
unigrm	.049	.000	-.479				
bigrm	.001	.005	-.006	-.073			
emdiff	.000	.009	-.049	-.089	.095		
etrpyrd	.000	.003	.016	-.014	.020	-.010	
srprsl	.000	-.008	-.033	-.079	.107	.362	.171

Table 3: Correlations in the full model.

(AIC=-10589) outperformed the best model *without* it (AIC=-10478), indicating that the HHMM surprisal is well worth including in the model regardless of the presence of other significant factors.

HHMM entropy reduction predicts reading times on the full dataset and on closed-class words. However, its effect on open-class words is insignificant; if we compare the model of column 2 against one without entropy reduction, a likelihood ratio test gives $\chi^2_1 = 0.0022, p = 0.9623$ (the smaller model could easily generate the same data).

The HHMM's average embedding difference is also significant except in the case of open-class words — removing embedding difference on open-class data yields $\chi^2_1 = 0.2739, p = 0.6007$. But what is remarkable is that there is any significance for this metric at all. Embedding difference and surprisal were relatively correlated compared to other predictors (see Table 3), which is expected because embedding difference is calculated like a weighted version of surprisal. Despite this, it makes an independent contribution to the full-data and closed-class models. Thus, we can conclude that the average embedding depth component affects reading times — i.e., the HHMM's notion of working memory behaves as we would expect human working memory to behave.

5 Discussion

As with previous work on large-scale parser-derived complexity metrics, the linear mixed-effect models suggest that sentence-level factors are effective predictors for reading difficulty — in these evaluations, better than commonly-used lexical and near-neighbor predictors (Pollatsek et al., 2006; Engbert et al., 2005). The fact that HHMM surprisal outperforms even n-gram metrics points to the importance of including a notion of sentence structure. This is particularly true when the sentence structure is defined in a language model that is psycholinguistically plausible (here, bounded-memory right-corner form).

This accords with an understated result of Boston et al.'s eye-tracking study (2008a): a richer language model predicts eye movements during reading better than an oversimplified one. The comparison there is between phrase structure surprisal (based on Hale's (2001) calculation from an Earley parser), and dependency grammar surprisal (based on Nivre's (2007) dependency parser). Frank (2009) similarly reports improvements in the reading-time predictiveness of unlexicalized surprisal when using a language model that is more plausible than PCFGs.

The difference in predictivity due to word class is difficult to explain. One theory may be that closed-class words are less susceptible to random effects because there is a finite set of them for any language, making them overall easier to predict via parser-derived metrics. Or, we could note that since closed-class words often serve grammatical functions in addition to their lexical content, they contribute more information to parser-derived measures than open-class words. Previous work with complexity metrics on this corpus (Roark et al., 2009) suggests that these explanations only account for part of the word-class variation in the performance of predictors.

Further comparsion to Roark et al. will show other differences, such as the lesser role of word length and unigram frequency, lower overall correlations between factors, and the greater predictivity of their entropy metric. In addition, their metrics are different from ours in that they are designed to tease apart lexical and syntactic contributions to reading difficulty. Their notion of entropy, in particular, estimates Hale's definition of entropy on whole derivations (2006) by isolating the predictive entropy; they then proceed to define separate lexical and syntactic predictive entropies. Drawing more directly from Hale, our definition is a whole-derivation metric based on the conditional entropy of the words, given the root. (The root constituent, though unwritten in our definitions, is always included in the HHMM start state, q_0.)

More generally, the parser used in these evaluations differs from other reported parsers in that it is not lexicalized. One might expect for this to be a weakness, allowing distributions of probabilities at each time step in places not licensed by the observed words, and therefore giving poor probability-based complexity metrics. However, we see that this language model performs well despite its lack of lexicalization. This indicates that lexicalization is not a requisite part of syntactic parser performance with respect to predicting linguistic complexity, corroborating the evidence of Demberg and Keller's (2008) 'unlexicalized' (POS-generating, not word-generating) parser.

Another difference is that previous parsers have produced useful complexity metrics without maintaining arc-eager/arc-standard ambiguity. Results show that including this ambiguity in the HHMM at least does not invalidate (and may in fact improve) surprisal or entropy reduction as reading-time predictors.

6 Conclusion

The task at hand was to determine whether the HHMM could consistently be considered a plausible psycholinguistic model, producing viable complexity metrics while maintaining other characteristics such as bounded memory usage. The linear mixed-effects models on reading times validate this claim. The HHMM can straightforwardly produce highly-predictive, standard complexity metrics (surprisal and entropy reduction). HHMM surprisal performs very well in predicting reading times regardless of word class. Our formulation of entropy reduction is also significant except in open-class words.

The new metric, embedding difference, uses the average center-embedding depth of the HHMM to model syntactic-processing memory cost. This metric can only be calculated on parsers with an explicit representation for short-term memory elements like the right-corner HHMM parser. Results show that embedding difference does predict reading times except in open-class words, yielding a significant contribution independent of surprisal despite the fact that its definition is similar to that of surprisal.

Acknowledgments

Thanks to Brian Roark for help on the reading times corpus, Tim Miller for the formulation of entropy reduction, Mark Holland for statistical insight, and the anonymous reviewers for their input. This research was supported by National Science Foundation CAREER/PECASE award 0447685. The views expressed are not necessarily endorsed by the sponsors.

References

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

Asaf Bachrach, Brian Roark, Alex Marantz, Susan Whitfield-Gabrieli, Carlos Cardenas, and John D.E. Gabrieli. 2009. Incremental prediction in naturalistic language processing: An fMRI study.

Douglas Bates, Martin Maechler, and Bin Dai. 2008. lme4: Linear mixed-effects models using S4 classes. R package version 0.999375-31.

Marisa Ferrara Boston, John T. Hale, Reinhold Kliegl, Umesh Patil, and Shravan Vasishth. 2008a. Parsing costs as predictors of reading difficulty: An evaluation using the Potsdam Sentence Corpus. *Journal of Eye Movement Research*, 2(1):1–12.

Marisa Ferrara Boston, John T. Hale, Reinhold Kliegl, and Shravan Vasishth. 2008b. Surprising parser actions and reading difficulty. In *Proceedings of ACL-08: HLT, Short Papers*, pages 5–8, Columbus, Ohio, June. Association for Computational Linguistics.

Thorsten Brants and Matthew Crocker. 2000. Probabilistic parsing and psychological plausibility. In *Proceedings of COLING '00*, pages 111–118.

Evan Chen, Edward Gibson, and Florian Wolf. 2005. Online syntactic storage costs in sentence comprehension. *Journal of Memory and Language*, 52(1):144–169.

Noam Chomsky and George A. Miller. 1963. Introduction to the formal analysis of natural languages. In *Handbook of Mathematical Psychology*, pages 269–321. Wiley.

Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24:87–185.

Matthew Crocker and Thorsten Brants. 2000. Wide-coverage probabilistic sentence processing. *Journal of Psycholinguistic Research*, 29(6):647–669.

Delphine Dahan and M. Gareth Gaskell. 2007. The temporal dynamics of ambiguity resolution: Evidence from spoken-word recognition. *Journal of Memory and Language*, 57(4):483–501.

Vera Demberg and Frank Keller. 2008. Data from eye-tracking corpora as evidence for theories of syntactic processing complexity. *Cognition*, 109(2):193–210.

Ralf Engbert, Antje Nuthmann, Eike M. Richter, and Reinhold Kliegl. 2005. SWIFT: A dynamical model of saccade generation during reading. *Psychological Review*, 112:777–813.

Shai Fine, Yoram Singer, and Naftali Tishby. 1998. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32(1):41–62.

Stefan L. Frank. 2009. Surprisal-based comparison between a symbolic and a connectionist model of sentence processing. In *Proc. Annual Meeting of the Cognitive Science Society*, pages 1139–1144.

Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Edward Gibson. 2000. The dependency locality theory: A distance-based theory of linguistic complexity. In *Image, language, brain: Papers from the first mind articulation project symposium*, pages 95–126.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 159–166, Pittsburgh, PA.

John Hale. 2003. *Grammar, Uncertainty and Sentence Processing*. Ph.D. thesis, Cognitive Science, The Johns Hopkins University.

John Hale. 2006. Uncertainty about the rest of the sentence. *Cognitive Science*, 30(4):609–642.

Roger Levy. 2008. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177.

Scott A. McDonald and Richard C. Shillcock. 2003. Low-level predictive inference in reading: The influence of transitional probabilities on eye movements. *Vision Research*, 43(16):1735–1751.

George Miller and Noam Chomsky. 1963. Finitary models of language users. In R. Luce, R. Bush, and E. Galanter, editors, *Handbook of Mathematical Psychology*, volume 2, pages 419–491. John Wiley.

Kevin P. Murphy and Mark A. Paskin. 2001. Linear time inference in hierarchical HMMs. In *Proc. NIPS*, pages 833–840, Vancouver, BC, Canada.

Joakim Nivre. 2007. Inductive dependency parsing. *Computational Linguistics*, 33(2).

Alexander Pollatsek, Erik D. Reichle, and Keith Rayner. 2006. Tests of the EZ Reader model: Exploring the interface between cognition and eye-movement control. *Cognitive Psychology*, 52(1):1–56.

Lawrence R. Rabiner. 1990. A tutorial on hidden Markov models and selected applications in speech recognition. *Readings in speech recognition*, 53(3):267–296.

Brian Roark, Asaf Bachrach, Carlos Cardenas, and Christophe Pallier. 2009. Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top-down parsing. *Proceedings of the 2009 Conference on Empirical Methods in Natural Langauge Processing*, pages 324–333.

Brian Roark. 2001. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276.

William Schuler, Samir AbdelRahman, Tim Miller, and Lane Schwartz. 2008. Toward a psycholinguistically-motivated model of language. In *Proceedings of COLING*, pages 785–792, Manchester, UK, August.

William Schuler, Samir AbdelRahman, Tim Miller, and Lane Schwartz. 2010. Broad-coverage incremental parsing using human-like memory constraints. *Computational Linguistics*, 36(1).

William Schuler. 2009. Parsing with a bounded stack using a model-based right-corner transform. In *Proceedings of NAACL*, pages 344–352, Boulder, Colorado.

Michael K. Tanenhaus, Michael J. Spivey-Knowlton, Kathy M. Eberhard, and Julie E. Sedivy. 1995. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634.