Expected surprisal and entropy

Brian Roark

August 25, 2011

Abstract

In this brief technical report, we build off of previous work in calculating surprisal and entropy measures from an incremental parser [1] and presented measures of Expected Surprisal, which are related to some of the entropy measures defined in the previous work.

1 Grammars, derivations, prefix probabilities[1]

A probabilistic context-free grammar (PCFG) $G = (V, T, S^\dagger, P, \rho)$ consists of a set of non-terminal variables V; a set of terminal items (words) T; a special start non-terminal $S^\dagger \in V$; a set of rule productions P of the form $A \rightarrow \alpha$ for $A \in V$, $\alpha \in (V \cup T)^*$; and a function ρ that assigns probabilities to each rule in P such that for any given non-terminal symbol $X \in V$, $\sum_\alpha \rho(X \rightarrow \alpha) = 1$.

For a given rule $A \rightarrow \alpha \in P$, let the function RHS return the right-hand side of the rule, i.e., RHS$(A \rightarrow \alpha) = \alpha$. Without loss of generality, we will assume that for every rule $A \rightarrow \alpha \in P$, one of two cases holds: either RHS$(A \rightarrow \alpha) \in T$ or RHS$(A \rightarrow \alpha) \in V^*$. That is, the right-hand side sequences consist of either (1) exactly one terminal item, or (2) zero or more non-terminals.

Let $W \in T^n$ be a terminal string of length n, i.e., $W = W_1 \dots W_n$ and $|W| = n$. Let $W[i, j]$ denote the substring beginning at word W_i and ending at word W_j of the string. Then $W_{|W|}$ is the last word in the string, and $W[1, |W|]$ is the string as a whole. Adjacent strings represent concatenation, i.e., $W[1, i]W[i{+}1, j] = W[1, j]$. Thus $W[1, i]w$ represents the string where $W_{i+1} = w$.

We can define a "derives" relation (denoted $\Rightarrow_G$ for a given PCFG G) as follows: $\beta A \gamma \Rightarrow_G \beta\alpha\gamma$ if and only if $A \rightarrow \alpha \in P$. A string $W \in T^*$ is in the language of a grammar G if and only if $S^\dagger \stackrel{+}{\Rightarrow}_G W$, i.e., a sequence of one or more derivation steps yields the string from the start non-terminal. A *leftmost* derivation begins with $S^\dagger$ and each derivation step replaces the leftmost non-terminal A in the yield with some α such that $A \rightarrow \alpha \in P$. For a leftmost derivation $S^\dagger \stackrel{*}{\Rightarrow}_G \alpha$, where $\alpha \in (V \cup T)^*$, the sequence of derivation steps that yield α can be represented as a tree, with the start symbol $S^\dagger$ at the root, and the "yield" sequence α at the leaves of the tree. A *complete* tree has only terminal items in the yield, i.e., $\alpha \in T^*$; a *partial* tree has some non-terminal items in the yield. With a leftmost derivation, the yield $\alpha = \beta\gamma$ partitions into an initial sequence of terminals $\beta \in T^*$ followed by a sequence of non-terminals $\gamma \in V^*$. For a complete derivation, $\gamma = \epsilon$; for a partial derivation $\gamma \in V^+$, i.e., one or more non-terminals. Let $\mathcal{T}(G, W[1, i])$ be the set of complete trees with $W[1, i]$ as the yield of the tree, given PCFG G.

A leftmost derivation D consists of a sequence of $|D|$ steps. Let D_i represent the i^{th} step in the derivation D, and $D[i, j]$ represent the subsequence of steps in D beginning with D_i and ending with D_j. Note that $D_{|D|}$ is the last step in the derivation, and $D[1, |D|]$ is the derivation as a whole. Each step D_i in the derivation is a rule in G, i.e., $D_i \in P$ for all i. The probability of the derivation and the corresponding tree

[1] Parts of these appendices are revisions of content that first appeared in [1].

is:

$$\rho(D) \quad = \quad \prod_{i=1}^{m} \rho(D_i) \tag{1}$$

The terminal yield of a leftmost derivation, which we will denote TermYield(D), is the sequence of terminal items at the leaves of the tree defined by D. Let $\mathcal{D}(G, W[1,i])$ be the set of all possible leftmost derivations D (with respect to G) such that TermYield(D) $= W[1,i]$ and RHS($D_{|D|}$) $= W_i$. These are the set of partial leftmost derivations with $W[1,i]$ as the terminal yield whose last step used a production with terminal W_i on the right-hand side. The prefix probability of $W[1,i]$ with respect to G is

$$\text{PrefixProb}_G(W[1,i]) \quad = \sum_{D \in \mathcal{D}(G,W[1,i])} \rho(D) \tag{2}$$

From this prefix probability, we can calculate the conditional probability of each word $w \in T$ in the terminal vocabulary, given the preceding sequence $W[1,i]$ as follows:

$$\begin{aligned} P_G(w \mid W[1,i]) &= \frac{\text{PrefixProb}_G(W[1,i]w)}{\sum_{w' \in T} \text{PrefixProb}_G(W[1,i]w')} \\ &= \frac{\text{PrefixProb}_G(W[1,i]w)}{\text{PrefixProb}_G(W[1,i])} \end{aligned} \tag{3}$$

This, in fact, is precisely the conditional probability that is used for language modeling for such applications as speech recognition and machine translation, which was the motivation for various syntactic language modeling approaches [2, 3, 4, 5].

As with language modeling, it is important to model the end of the string as well, usually with an explicit end symbol, e.g., </s>. For a string $W[1,i]$, we can calculate its prefix probability as shown above. To calculate its complete probability, we must sum the probabilities over the set of complete trees $\mathcal{T}(G, W[1,i])$. In such a way, we can calculate the conditional probability of ending the string with </s> given $W[1,i]$ as follows:

$$P_G(\text{</s>} \mid W[1,i]) = \frac{\sum_{D \in \mathcal{T}(G,W[1,i])} \rho(D)}{\text{PrefixProb}_G(W[1,i])} \tag{4}$$

2 Incremental top-down parsing

In this section, we review relevant details of the Roark [5] incremental top-down parser, as configured for use here. As presented in [6], the probabilities in the PCFG are smoothed so that the parser is guaranteed not to fail due to garden pathing, despite following a beam search strategy. Hence there is always a non-zero prefix probability as defined in Eq. 2.

The parser follows a top-down leftmost derivation strategy. The grammar is factored so that every production has either a single terminal item on the right-hand side or is of the form A $\rightarrow$ B A-B, where A,B $\in V$ and the factored A-B category can expand to any sequence of children categories of A that can follow B. This factorization of n-ary productions continues to nullary factored productions, i.e., the end of the original production A $\rightarrow$ $\text{B}_1 \ldots \text{B}_n$ is signaled with an empty production A-B_1-$\ldots$-B_n $\rightarrow \epsilon$.

The parser maintains a set of possible connected derivations, weighted via the PCFG. It uses a beam search, whereby the highest scoring derivations are worked on first, and derivations that fall outside of the beam are discarded. The reader is referred to Roark [5, 6] for specifics about the beam search.

The model conditions the probability of each production on features extracted from the partial tree, including non-local node labels such as parents, grandparents and siblings from the left-context, as well as c-commanding lexical items. Hence this is a lexicalized grammar, though the incremental nature precludes a general head-first strategy, rather one that looks to the left-context for c-commanding lexical items.

To avoid some of the early prediction of structure, the version of the Roark parser that we used performs an additional grammar transformation beyond the simple factorization already described – a selective left-corner transform of left-recursive productions [7]. In the transformed structure, slash categories are used to avoid predicting left-recursive structure until some explicit indication of modification is present, e.g., a preposition.

The final step in parsing, following the last word in the string, is to "complete" all non-terminals in the yield of the tree. All of these open non-terminals are composite factored categories, such as S-NP-VP, which are "completed" by rewriting to ϵ. The probability of these ϵ productions is what allows for the calculation of the conditional probability of ending the string, shown in Eq. 4.

One final note about the size of the non-terminal set and the intractability of exact inference for such a scenario. The non-terminal set not only includes the original atomic non-terminals of the grammar, but also any categories created by grammar factorization (S-NP) or the left-corner transform (NP/NP). Additionally, however, to remain context-free, the non-terminal set must include categories that incorporate non-local features used by the statistical model into their label, including parents, grandparents and sibling categories in the left-context, as well as c-commanding lexical heads. These non-local features must be made local by encoding them in the non-terminal labels, leading to a very large non-terminal set and intractable exact inference. Heavy smoothing is required when estimating the resulting PCFG. The benefit of such a non-terminal set is a rich model, which enables a more peaked statistical distribution around high quality syntactic structures and thus more effective pruning of the search space. The fully connected left-context produced by top-down derivation strategies provides very rich features for the stochastic parsing models. See Roark [5, 6] for discussion of these issues.

3 Parser and grammar derived measures

3.1 Surprisal

The *surprisal* at word W_i is the negative log probability of W_i given the preceding words. Using prefix probabilities, this can be calculated as:

$$S_G(W_i) \;=\; -\log \frac{\text{PrefixProb}_G(W[1,i])}{\text{PrefixProb}_G(W[1,i-1])} \tag{5}$$

Substituting equation 2 into this, we get

$$S_G(W_i) \;=\; -\log \frac{\sum_{D \in \mathcal{D}(G,W[1,i])} \rho(D)}{\sum_{D \in \mathcal{D}(G,W[1,i-1])} \rho(D)} \tag{6}$$

If we are using a beam-search parser, some of the derivations are pruned away. Let $\mathcal{B}(G, W[1,i]) \subseteq \mathcal{D}(G, W[1,i])$ be the set of derivations in the beam. Then the surprisal can be approximated as

$$S_G(W_i) \;\approx\; -\log \frac{\sum_{D \in \mathcal{B}(G,W[1,i])} \rho(D)}{\sum_{D \in \mathcal{B}(G,W[1,i-1])} \rho(D)} \tag{7}$$

Any pruning in the beam search will result in a *deficient* probability distribution, i.e., a distribution that sums to less than 1. Roark's thesis (2001) showed that the amount of probability mass lost for this particular approach is very low, hence this provides a very tight bound on the actual surprisal given the model.

3.2 Lexical and Syntactic surprisal

High surprisal scores result when the prefix probability at word W_i is low relative to the prefix probability at word W_{i-1}. Sometimes this is due to the identity of W_i, i.e., it is a surprising word given the context. Other times, it may not be the lexical identity of the word so much as the syntactic structure that must be created to integrate the word into the derivations. One would like to tease surprisal apart into "syntactic surprisal" versus "lexical surprisal", which would capture this intuition of the lexical versus syntactic dimensions to the score. Our solution to this has the beneficial property of producing two scores whose sum equals the original surprisal score.

The original surprisal score is calculated via sets of partial derivations at the point when each word W_i is integrated into the syntactic structure, $\mathcal{D}(G, W[1,i])$. We then calculate the ratio from point to point in sequence. To tease apart the lexical and syntactic surprisal, we will consider sets of partial derivations *immediately before* each word W_i is integrated into the syntactic structure, i.e., $D[1, |D|-1]$ for $D \in \mathcal{D}(G, W[1,i])$. Recall that the last derivation move for every derivation in the set is from the POS-tag to the lexical item. Hence the sequence of derivation moves that excludes the last one includes all structure except the word W_i. Let $S(\mathcal{D}) = \{D[1, |D|-1] : D \in \mathcal{D}\}$, i.e., the set of derivations achieved by removing the last step of all derivations in $\mathcal{D}$. Then the syntactic surprisal is calculated as:

$$\text{Syn}S_G(W_i) = -\log \frac{\sum_{D \in S(\mathcal{D}(G, W[1,i]))} \rho(D)}{\sum_{D \in \mathcal{D}(G, W[1,i-1])} \rho(D)} \tag{8}$$

and the lexical surprisal is calculated as:

$$\text{Lex}S_G(W_i) = -\log \frac{\sum_{D \in \mathcal{D}(G, W[1,i])} \rho(D)}{\sum_{D \in S(\mathcal{D}(G, W[1,i]))} \rho(D)} \tag{9}$$

Note that the numerator of $\text{Syn}S_G(W_i)$ is the denominator of $\text{Lex}S_G(W_i)$, hence they sum to form total surprisal $S_G(W_i)$. As with total surprisal, these measures can be defined either for the full set $\mathcal{D}(G, W[1,i])$ or for a pruned beam of derivations $\mathcal{B}(G, W[1,i]) \subseteq \mathcal{D}(G, W[1,i])$.

3.3 Entropy

Entropy scores of the sort advocated by Hale [8, 9] involve calculation over the set of complete derivations consistent with the set of partial derivations. Hale performs this calculation efficiently via matrix inversion, which explains the use of relatively small-scale grammars with tractably sized non-terminal sets. Such methods are not tractable for the kinds of richly conditioned, large-scale PCFGs that we advocate using here. At each word in the string, the Roark [5] top-down parser provides access to the weighted set of partial analyses in the beam; the set of complete derivations consistent with these is not immediately accessible, hence additional work is required to calculate such measures.

Let $H(\mathcal{D})$ be the entropy over a set of derivations $\mathcal{D}$, calculated as follows:

$$H(\mathcal{D}) = -\sum_{D \in \mathcal{D}} \frac{\rho(D)}{\sum_{D' \in \mathcal{D}} \rho(D')} \log \frac{\rho(D)}{\sum_{D' \in \mathcal{D}} \rho(D')} \tag{10}$$

If the set of derivations $\mathcal{D} = \mathcal{D}(G, W[1,i])$ is a set of partial derivations for string $W[1,i]$, then $H(\mathcal{D})$ is a measure of uncertainty over the partial derivations, i.e., the uncertainty regarding the correct analysis of what has already been processed. This can be calculated directly from the existing parser operations. If

the set of derivations are the complete derivations *consistent* with the set of partial derivations – complete derivations that could occur over the set of possible continuations of the string – then this is a measure of the uncertainty about what is yet to come. We would like measures that can capture this distinction between (a) uncertainty of what has already been processed ("current ambiguity") versus (b) uncertainty of what is yet to be processed ("predictive entropy"). In addition, as with surprisal, we would like to tease apart the syntactic uncertainty versus lexical uncertainty.

To calculate the predictive entropy after word sequence $W[1,i]$, we modify the parser as follows: the parser extends the set of partial derivations to include all possible next words (the entire vocabulary plus </s>), and calculates the entropy over that set. This measure is calculated from just one additional word beyond the current word, and hence is an approximation to Hale's conditional entropy of grammatical continuations, which is over complete derivations. We will denote this as $H^1_G(W[1,i])$ and calculate it as follows:

$$H^1_G(W[1,i]) \;=\; H(\bigcup_{w \in T \cup \{\text{</s>}\}} \mathcal{D}(G, W[1,i]w)) \tag{11}$$

This is performing a predictive step that the baseline parser does not perform, extending the parses to all possible next words. As a practical matter, these values are calculated within the Roark parser as follows. A "dummy" word is created that can be assigned every POS-tag, and the parser extends from the current state to this dummy word. (The beam threshold is greatly expanded to allow for many possible extensions.) Then every word in the vocabulary is substituted for the word, and the appropriate probabilities calculated over the beam. Finally, the actual next word is substituted, the beam threshold is reduced to the actual working threshold, and the requisite number of analyses are advanced to continue parsing the string. This represents a significant amount of additional work for the parser – particularly for vocabulary sizes that we currently use, on the order of tens of thousands of words.

3.4 Expected Surprisal

Unlike surprisal, entropy does not decompose straightforwardly into syntactic and lexical components that sum to the original composite measure. However, one version of a predictive entropy measure, termed "lexical entropy" in [1] does move away from the distribution over derivations towards the distribution over words, defined in terms of the conditional probabilities derived from prefix probabilities as defined in Eq. 3.

$$\text{Lex}H^1_G(W[1,i]) \;=\; -\sum_{w \in T \cup \{\text{</s>}\}} P_G(w \mid W[1,i]) \log P_G(w \mid W[1,i]) \tag{12}$$

Note, however, that this so-called "lexical entropy" is exactly the expected value of surprisal (Equation 6), which is what we will call it. In expected value notation, this is $\mathcal{E}[S_G(W_i)]$ and

$$\mathcal{E}[S_G(W_i)] \;=\; -\sum_{w \in T \cup \{\text{</s>}\}} P_G(w \mid W[1,i]) \log \frac{\sum_{D \in \mathcal{D}(G, W[1,i]w)} \rho(D)}{\sum_{D \in \mathcal{D}(G, W[1,i])} \rho(D)} \tag{13}$$

In the same way, we can define the expected value of our other two surprisal measures (lexical and syntactic). The expected syntactic surprisal $\mathcal{E}[\text{Syn}S_G(W_i)]$ is the sum over all possible following words (including end-of-string) of the probability of that word times the syntactic surprisal at that word. Similarly, the expected lexical surprisal $\mathcal{E}[\text{Lex}S_G(W_i)]$ is the sum over all possible following words (including end-of-string) of the

probability of the word times the lexical surprisal at that word. Recall that $S(\mathcal{D}) = \{D[1, |D|-1] : D \in \mathcal{D}\}$, i.e., the set of derivations achieved by removing the last step of all derivations in $\mathcal{D}$. Then

$$\mathcal{E}[\text{Syn}S_G(W_i)] = -\sum_{w \in T \cup \{\text{</s>}\}} P_G(w \mid W[1,i]) \log \frac{\sum_{D \in S(\mathcal{D}(G,W[1,i]))} \rho(D)}{\sum_{D \in \mathcal{D}(G,W[1,i-1])} \rho(D)} \quad (14)$$

and

$$\mathcal{E}[\text{Lex}S_G(W_i)] = -\sum_{w \in T \cup \{\text{</s>}\}} P_G(w \mid W[1,i]) \log \frac{\sum_{D \in \mathcal{D}(G,W[1,i])} \rho(D)}{\sum_{D \in S(\mathcal{D}(G,W[1,i]))} \rho(D)} \quad (15)$$

Once we have defined the expected surprisal measures, we can "normalize" the actual observed surprisal by subtracting the expected value before the word was seen.

$$\overline{S_G}(W_i) = S_G(W_i) - \mathcal{E}[S_G(W_{i-1})] \quad (16)$$
$$\overline{\text{Syn}S_G}(W_i) = \text{Syn}S_G(W_i) - \mathcal{E}[\text{Syn}S_G(W_{i-1})] \quad (17)$$
$$\overline{\text{Lex}S_G}(W_i) = \text{Lex}S_G(W_i) - \mathcal{E}[\text{Lex}S_G(W_{i-1})] \quad (18)$$

For these normalized measures, a value above zero means that the surprisal was greater than expected, and lower than zero means lower than expected. This may capture distinctions between contexts which make weak predictions about subsequent content from those which make strong predictions about subsequent content.

Note that the nice property that the lexical and syntactic components sum to the total surprisal continues to hold for the expected values and normalized measures:

$$\mathcal{E}[S_G(W_i)] = \mathcal{E}[\text{Syn}S_G(W_i)] + \mathcal{E}[\text{Lex}S_G(W_i)] \quad (19)$$
$$\overline{S_G}(W_i) = \overline{\text{Syn}S_G}(W_i) + \overline{\text{Lex}S_G}(W_i) \quad (20)$$

References

[1] B. Roark, A. Bachrach, C. Cardenas, and C. Pallier, "Deriving lexical and syntactic expectation-based measures for psycholinguistic modeling via incremental top-down parsing," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2009, pp. 324–333.

[2] F. Jelinek and J. Lafferty, "Computation of the probability of initial substring generation by stochastic context-free grammars," *Computational Linguistics*, vol. 17, no. 3, pp. 315–323, 1991.

[3] A. Stolcke, "An efficient probabilistic context-free parsing algorithm that computes prefix probabilities," *Computational Linguistics*, vol. 21, no. 2, pp. 165–202, 1995.

[4] C. Chelba and F. Jelinek, "Exploiting syntactic structure for language modeling," in *Proceedings of ACL-COLING*, 1998, pp. 225–231.

[5] B. Roark, "Probabilistic top-down parsing and language modeling," *Computational Linguistics*, vol. 27, no. 2, pp. 249–276, 2001.

[6] B. Roark, "Robust garden path parsing," *Natural Language Engineering*, vol. 10, no. 1, pp. 1–24, 2004.

[7] M. Johnson and B. Roark, "Compact non-left-recursive grammars using the selective left-corner transform and factoring," in *Proceedings of COLING*, 2000, pp. 355–361.

[8] J.T. Hale, "The information conveyed by words in sentences," *Journal of Psycholinguistic Research*, vol. 32, no. 2, pp. 101–123, 2003.

[9] J.T. Hale, "Uncertainty about the rest of the sentence," *Cognitive Science*, vol. 30, no. 4, pp. 643–672, 2006.