Reading Comprehension Difficulty in the scope of Surprisal

Isaac Lee, *Ling 185B - Final Paper*

Abstract

Given a phrase-structural language model and a probabilistic parser, surprisal can be calculated to predict the word-by-word processing time of a human reader. However, while surprisal is a good indicator for predicting reading time, there are limits as to what processing phenomena its total parallelism approach can predict about the human parser which is assumed to be much more bounded. There's always the question of how accurately the limited sample grammar reflects the complex natural language syntax. That being said, in the cases of NP/Z ambiguity, surprisal and the underlying grammar structures may give additional information on the source of ambiguity.

1. Introduction

Predicting reading comprehension difficulty has long been a topic of interest among many scholars, and linguists over the years have come up with varying ways to gauge them. A better understanding of the topic can have a wide range of impacts, from helping solve oral language disorder to enhancing machine learning models. There are multiple ways to approach this problem, and thanks to new tools and formalisms introduced in the field, human sentence processing can be viewed with different levels of abstraction.

To better understand the topic and raise additional questions for further research, we revisit a few of the relevant literature in discussing surprisal and analyze the presented data.

2. Surprisal in Hale 2001

In the Hale 2001 paper, we are presented with an information-theoretical analysis technique, surprisal, to predict reading comprehension difficulty on difficult-to-process sentences. Surprisal can be used to predict the processing difficulty of a word given its context. Surprisal, with its roots in information theory, is measured in "bits," reflecting how much more memory is needed to encode the particular outcome (negative log of the conditional probability). If the outcome is very general and happens all the time, it will be very low. If the outcome is extremely rare and happens, say, 1 in a million, then the surprisal would be ~20 ($\log_2$ million).

Throughout the Hale 2001 paper, several data are generated from toy grammars to show how the surprisal predictions change based on certain syntactic variance. While the resulting data was used to prove that some grammatical structures are more difficult to process than others (increase in surprisal prediction), it stands to reason if the differences are due to the syntactic complications or simply from the sophistication of the toy grammar that was carefully carved to give the desired outcome. In the following sections, I present the original and the modified grammars to compare the results and analyze the source of incongruity.

2.1 Subject/Object asymmetry

In section 7 of the Hale 2001 paper, we are introduced with the following probabilistic-context-free grammar.

0.33	NP	→	SPECNP NBAR
0.33	NP	→	you
0.33	NP	→	me
1	SPECNP	→	DT
0.5	NBAR	→	NBAR S[+R]
0.5	NBAR	→	N
1	S	→	NP VP
0.868646	S[+R]	→	NP[+R] VP
0.131354	S[+R]	→	NP[+R] S/NP
1	S/NP	→	NP VP/NP
1	VP/NP	→	V NP/NP
1	VP	→	V NP
1	V	→	saw
1	NP[+R]	→	who
1	DT	→	the
1	N	→	man
1	NP/NP	→	ε

This grammar produces the following two sentences with the corresponding predictions to suggest that sentence (b), with the object relative clause, has a higher surprisal prediction than (a), subject relative clause, and therefore is harder for the reader to process.

		Mean[1]
(a)	the man who saw you saw me	~2.1
(b)	the man who you saw saw me	~5.0

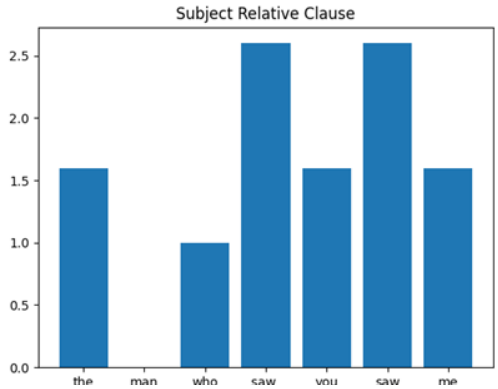
However, with a slight modification to the probabilities and the addition of extra rules, we can achieve a very different result. Here is the modified grammar.

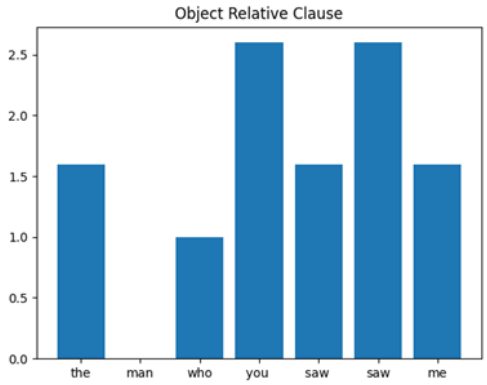
1	S	→	NP VP
0.5	S+R	→	NP+R VP
0.5	S+R	→	NP+R S/NP
0.33	NP	→	DT NBAR
0.33	NP	→	you
0.33	NP	→	me
0.5	NBAR	→	NBAR S+R
0.5	NBAR	→	man
1	S/NP	→	NP V
1	VP	→	V NP

[1] It took me some time to understand what 'mean' implied here in the paper; the 'mean' is the average of the prefix/current ratio at each word, not the average of surprisal.

0.33	V	→	saw
0.33	V	→	told
0.33	V	→	passed
1	NP+R	→	who
1	DT	→	the

This grammar was converted to conform to CNF (as presented in HW3) and I added a few extra rules with probability adjustments. With these adjustments, we get the prediction that is very much symmetric.





It is not a coincidence that these two predictions are identical. There are two primary reasons why the original grammar predicted higher surprisal for the subject relative clause compared to the object counterpart. One reason is the skewed distribution of probability for the S[+R] rule. The rule that projects SRC have a much higher probability than for the one that projects ORC. This will make the ORC interpretation more "surprising" and thus higher surprisal. Another reason is that there is only one verb compared to three nouns in the grammar. Because there is only 0.33 chance of selecting the correct noun, the surprisal is higher at the noun "you" in the ORC sentence[2].

[2] It is worth mentioning that the number of nouns/verbs does not affect the average of 'surprisal' but does change the 'mean' as presented in the Hale paper. In the paper, 'mean' is loosely used to compare the comprehension difficulty between two sentences.

So when we have probabilities evenly distributed, we get a result that does not predict a higher surprisal for the object relative clause than the subject relative clause, but rather we see an exact symmetry between them. Does this imply that the data in the paper is inaccurate? Not so. The toy grammar presented in the Hale paper reflects the probability distribution of natural language much closer than the one I've designed. For the probability distribution of SRCs and ORCs, it is much more skewed than even in the natural usage distribution, and while context may matter, there are generally more distinct nouns than verbs in English. So while the grammar in the paper may be simple and artificial, it does well to point out the asymmetry of SRC and ORC in the context of human sentence processing and surprisal prediction.

2.2 Limitation of surprisal

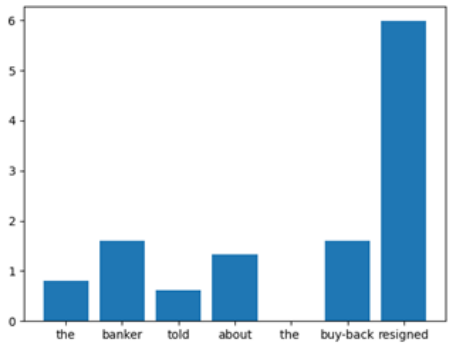
Now let's expand upon another example in Hale paper. The following example was used to compare the surprisal predictions of a subject relative clause and a reduced relative clause. In the paper, it concluded that reduced relative clause, *the banker told about the buy-back resigned,* was harder to process than the subject relative clause, *the banker who was told about the buy-back resigned*, because the surprisal prediction was higher on the disambiguating word *resigned* for RRC than SRC (6.68 over 5.88). I will follow a similar approach to get a contradictory result. As it was done before (homework 3), I modified the grammar in Hale paper to conform to Chomsky Normal Form (CNF), but the probability distribution it defines is the same.

0.574928	S	→	NP VP
0.046941	S	→	VBD PP
0.059979	S	→	VBD NPPP
0.091273	S	→	AUX VP
0.206017	S	→	VBN PP
0.015503	S	→	told
0.00536	S	→	resigned
1	SBAR	→	WHNP VP
0.804124	NP	→	DT NN
0.082474	NP	→	NP SBAR
0.113402	NP	→	NP VP
0.11043	VP	→	VBD PP
0.141104	VP	→	VBD NPPP
1	NPPP	→	NP PP
0.214724	VP	→	AUX VP
0.484663	VP	→	VBN PP
0.036471	VP	→	told
0.012609	VP	→	resigned
1	PP	→	IN NP
1	WHNP	→	who
1	DT	→	the
0.33	NN	→	boss
0.33	NN	→	banker
0.33	NN	→	buy-back
0.5	IN	→	about
0.5	IN	→	by
1	AUX	→	was
0.743094	VBD	→	told
0.256906	VBD	→	resigned

1 VBN → told

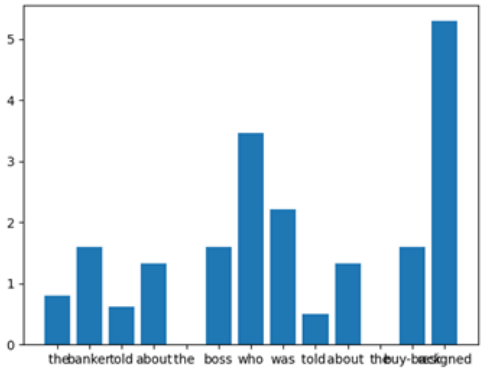
This grammar generates the reduced relative clause (a). I reproduced[3] the surprisal values in figure 5 of the Hale paper below.

(a) the banker told about the buy-back resigned



In this example, we see the high surprisal prediction on *resigned* (~5.98). This is to be expected given the ambiguity of the word *told*. Now for a moment think about the following sentence (b) and compare it with the sentence (a).

(b) the banker told about the boss who was told about the buy-back resigned



The sentence (b) at first glance might not make much sense, but at its core, we are given the same ambiguity presented in the sentence (a). The difference between the two sentences is the length of the reduced relative clause.

told about the buy-back vs
told about the boss who was told about the buy-back

We can tell that (b) is much harder to process than (a)[4]. With the human reader, there will be a greater comprehension difficulty at "resigned" for (b) than (a). However, if we compared the surprisal prediction of both sentence, we get that (b) has a lower surprisal prediction than (a) (5.29 vs 5.98).

2.3 Total parallelism vs bounded parallelism

The reason why the processing difficulty in (b) is not captured with surprisal prediction is that surprisal is fully information-theoretical analysis grounded in total parallelism (*full* parallelism in Levy 2008). In total parallelism, all the possible tree structures are in play until told otherwise, and in the case (b) above, lengthening the relative clause does not add any new possibilities when finally confronting the disambiguating word "resigned." In our example, it is the contrary, lengthening the relative clause have canceled out a few possible structures (not related to the ambiguating word "told").

Given that the increased comprehension difficulty in (b) is real, we then wonder if the human parser instead commits to bounded-parallelism, where the number of possible structures maintained is limited in the incremental reading process.

3. NP/Z ambiguity

The type of ambiguity we are dealing with is called NP/Z ambiguity (Levy 2008). The bases of NP/Z is an ambiguity caused by a verb that can either be transitive or intransitive. Take a look at the below sentence with NP/Z ambiguity.

(1) Although the man **stopped** the car still hit the boy.

In this sentence, **stopped** causes NP/Z ambiguity because it initially can be either transitive with the complement *the car* or intransitive with no complement. In our example with *banker*, although the structure is different, the sentence is still classified as NP/Z sentence because the main verb is initially thought by the reader to be a VBD where in reality it is VBN[5].

3.1 Source of ambiguity

What surprisal tells us is the level of difficulty in comprehension while incrementally reading the text word-by-word, left-to-right. Surprisal predicts **where** the reader has trouble

[3] To be honest, I couldn't quite replicate the values in the hale paper. While all other values are correct, the surprisal prediction at the last word "resigned" comes out to be 5.98 on my end but is 6.68 in Hale paper. I'm not sure where the discrepancy arises from; it may be that CNF conversion was done incorrectly or my code has a bug. I could not find the reason and was starting to wonder maybe the calculation in the paper was wrong. Either way, the slight difference is not relevant in this context.

[4] Ideally, this is where I plug in some empirical data to prove that (b) is indeed more difficult for a human to process than (a). I wanted to build some sort of eye-tracker to get real data, but it was too time-consuming to incorporate. For the purpose of this paper, I will naively make the assumption and take this as a fact.

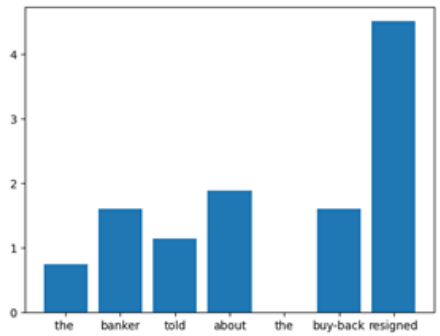
[5] Levy 2008 also has a sentence with the same structure classified as NP/Z.

parsing the sentence, but it does not show **what** caused such difficulty. The word with the highest surprisal prediction is the disambiguating word, which eliminates the initial (wrong) interpretation and leaves only the correct syntactic structures. However, this does not reveal where the confusion started.

Let's look at another example. Here is the grammar[6] and a sentence that has the NP/Z ambiguity.

0.59579	S	-->	NP VP
0.046941	S	-->	VBD PP
0.059979	S	-->	VBD NPPP
0.091273	S	-->	AUX VP
0.206017	S	-->	VBN PP
1	SBAR	-->	WHNP VP
0.804124	NP	-->	DT NN
0.082474	NP	-->	NP SBAR
0.113402	NP	-->	NP VP/N
0.22	VP	-->	VBD PP
0.25	VP	-->	VBD NPPP
1	NPPP	-->	NP PP
0.43	VP	-->	AUX VP
1	VP/N	-->	VBN PP
0.074309	VP	-->	told
0.025691	VP	-->	resigned
1	PP	-->	IN NP
1	WHNP	-->	who
1	DT	-->	the
0.33	NN	-->	boss
0.33	NN	-->	banker
0.33	NN	-->	buy-back
0.5	IN	-->	about
0.5	IN	-->	by
1	AUX	-->	was
0.743094	VBD	-->	told
0.256906	VBD	-->	resigned
1	VBN	-->	told

(1) the banker told about the buy-back resigned



[6] I've designed the grammar so that my algorithm for finding the ambiguating word works. This algorithm does not work on other grammars in the paper.

Based on these surprisal predictions, there is nothing that tells us that the word *told* is the source of ambiguity. Surprisal cannot track the source of ambiguity. In order to locate the "ambiguating" word, we need to question which structural analysis of the sentence is maintained and which are lost.

3.2 Algorithm to find the ambiguating word

I've written a python script to locate the source of NP/Z ambiguity given a pCFG and a string in question.

While there may be multiple ways to approach this problem, I came up with the following algorithm to locate the correct word(s).

1. Calculate the surprisal values for all prefixes.
2. Locate the word with the highest surprisal prediction. This is the disambiguating word.
3. Create two weighted CFGs; one intersected with the string up to the disambiguating word, and another one intersected with the string up to just before the disambiguating word.
4. Filter out unreachable rules in each CFGs.
5. Retrieve the terminal rules that output the same word at the same index with a different target NT symbol. Repeat for both CFGs.
6. Isolate the rules unique to one CFG (one will always be a subset of the other). The terminal symbols targeted by these rules are the ambiguating words[7].

Following this algorithm, we get the result that shows us that *told* is the ambiguating word and *resigned* is the disambiguating word.

This algorithm is by no means generic. There are many cases where this algorithm will fail. For the purpose of this paper, I will not dive deeper into this issue and leave the rest for future work.

4. Applying pTSL model to locality

The increase in comprehension difficulty we saw with "banker" is an example of what is called the digging-in effect. As the reader digs deeper into the ambiguous NP with the incorrect interpretation, the harder it is to retrieve the correct interpretation when arriving at the disambiguating word. As noted above, surprisal cannot account for the digging-in effects. In order to correlate predictions with reading difficulty, we need to account for the digging-in effects with other formalism.

[7] There could be multiple words affecting the ambiguity.

3.1 Failed attempt

One of the ideas I tried to integrate into the concept of digging-in was using the probabilistic Tier-based Strictly Local grammar (pTSL; Mayer, 2021)[8]. I tried to use the weights of CFG at each prefix as the projection probability and somehow show that as more words with low projectable probabilities appear before the disambiguation, the more unlikely it is to predict the disambiguating word. However, no matter how I used this model, it always boiled down to simple multiplications/subtracts between the probabilities. Basically, I wrote a lot of code and introduced a rather complex formalism in pTSL only to use a very minimal subset of its functionalities (multiplication), and I could have arrived at the same results in a much simpler manner.

5. Conclusion

Surprisal theory uses calculations from the total probability of structural options (Hale 2001), which maintains all possible structures while incrementally reading the text word-by-word. This commits to total parallelism, which at times does not result in an ideal conclusion when considering the human parser, which likely is much more bounded. However, bounded parallelism is difficult to replicate, mainly because the "bound" is not something we can simply measure. Also, surprisal depends on CFG/PCFG, which is inadequate to appropriately model natural language in general cases (Shieber, 1985).

In the end, processing difficulties are due to the cognitive/memory loads, and there are many ways to define them. Surprisal can capture some important processing phenomena, but in other cases, other formalisms are needed to acquire better predictions. That being said, there may be more information that we can obtain in surprisal than simple word-by-word processing difficulty.

Appendix

1.1 How to run proj.py

proj.py is an extension of work done for homework 3, where I calculated the surprisal value of the last word in the sentence given a PCFG. All the grammar in the script is in Chomsky Normal Form. Please see the documentation in the script for a detailed explanation of each function.

List of major functions in proj.py:

intersect(pcfg, words): Intersect the given pcfg with the given sequence of words.

surprisal(pcfg, words): calculate surprisal.

ambiguity_finder(pcfg, string): find a list of ambiguiating words and disambiguating word.

pretty_print1,2,3: print the results in a presentable format.

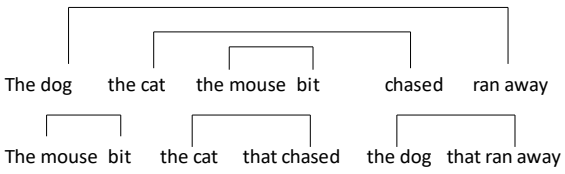
generate_barplot(): All the graphs used in this paper are generated with matplotlib library.

Here's an example output of *the banker told about the buy-back resigned* with the grammar shown in section 3.

```
Disambiguating word:
        resigned
List of ambiguating words
        "told" at index 3
```

1.2 Locality theories

In Levy 2008 paper, two locality-based processing theories are introduced. The first is Gibson's Dependency Locality Theory (DLT; Gibson, 1998), and Active filler Hypothesis. While it would be nice to analyze the digging-in effects with reference to these ideas, I could not make time to do enough research.



References

Graf, T., Monette, J., & Zhang, C. (2017). Relative clauses as a benchmark for Minimalist parsing. Journal of Language Modelling, 5(1), 57–106. https://doi.org/10.15398/jlm.v5i1.157

Hale, J. (2001). A probabilistic Earley parser as a psycholinguistic model. In Proceedings of NAACL (Vol. 2, pp. 159–166).

Hale, J. (2003). Grammar, Uncertainty and Sentence Processing. Ph.D. thesis, John Hopkins University.

Levy, Roger. (2007). Expectation-based syntactic comprehension. Cognition (Vol 106, pp. 1126-1177).

Mayer, Connor (2021) "Capturing gradience in long-distance phonology using probabilistic tier-based strictly local grammars," Proceedings of the Society for Computation in Linguistics: Vol. 4, Article 5.

[8] I devoted a lot of hours to come up with a reasonable proposal to integrate pTSL with prefix probabilities/surprisal to predict comprehension difficultly that correlates with digging-in effect, but ultimately could not find anything particularly meaningful. Maybe a dead-end?