# Parsing with Analogical Substitutions: An Exemplar-based Model of the Emergence of Syntactic Structure

**Anonymous ACL submission**

## Abstract

We outline a partially specified, exemplar-based parsing method that, given a sentence and a raw training corpus, assigns scores to possible binary parse trees of the sentence by finding corpus phrases that are grammatically analogous to the constituents of the parse tree. The general idea behind the parsing method is this: in order to parse a sentence like *my cat jumped on the table*, we (1) split it up into two possible constituents, e.g. *my cat* and *jumped on the table*, then (2) find grammatically analogous corpus phrases for these constituents, e.g. *her dog* for *my cat* and *ran to John* for *jumped on the table*, and (3) examine the distributions of these corpus phrases in order to find evidence for the well-formedness of combining the original constituents. We give a brief introduction to exemplar-based inference and parsing methods in linguistics, we outline how the proposed method finds analogous phrases and compares their distributions, and we present examples of analogous words found by this method for words in a corpus. We describe how syntactic structure may emerge from analogical inference.

## 1 Introduction

Exemplar-based inference methods in linguistics make predictions about novel linguistic expressions by comparing them to attested expressions, or *exemplars*, such that a novel expression is predicted to behave like its most similar exemplars. Exemplar-based linguistic theories represent linguistic knowledge as a database of exemplars plus some exemplar-based inference method — notably, generalizations over exemplars are not explicitly represented but are implicit in the computation of a prediction, as opposed to rule-based theories (where generalizations are represented as rules) and connectionist theories (where generalizations are represented as the weighted connections of a neural network).

While recent advances in large language models have demonstrated the applicability of connectionist theories in natural language processing tasks, exemplar-based theories remain theoretically attractive for their interpretability. Whenever an exemplar-based inference method makes a prediction, it inherently identifies a similarity-weighted set of exemplars reflecting the contribution of each exemplar to the predicted behavior, allowing us to form hypotheses about why expressions behave the way they do.

In any exemplar-based theory of *parsing* (assigning syntactic structures to sentences), the exemplars that contribute to the predicted structure would intuitively include sentences that are grammatically similar to the input sentence. A recent example is Ambridge's (2019, p. 534) account of exemplar-based sentence processing, where a passive sentence like *the vase was broken by the hammer* is said to evoke, among other things, passive exemplar sentences like *the window was smashed by a ball*.

But neither Ambridge's account nor any other exemplar-based account of sentence processing (that we are aware of) includes an algorithm that parses an input sentence by finding the most grammatically similar exemplar sentences. Bod's (2009) Unsupervised Data-Oriented Parsing model is able to parse sentences based on raw corpus data but it does not do this by computing the similarity of sentences; rather, it treats the set of all possible parse trees of corpus sentences as its effective training set and assigns probabilities to the possible parse trees of an input sentence based on the frequencies of their subtrees within that set. Walsh et al.'s (2010) multilevel exemplar model can compute the similarity of sentences but it is not a parsing algorithm; it instead computes a kind of "local well-formedness" of an input sentence $s$ based on its similarity to some exemplar sentence $s'$, by checking if each $n$-gram of words in $s$ has a corresponding $n$-gram

of grammatically similar words in $s'$ (where the grammatical similarity of two words is calculated using the cosine similarities of their left and right probability distributions).

Although these models do not carry out parsing by identifying similar sentences, they both include mechanisms that could be adapted for this purpose. In sections 2 and 3 we outline a method that uses some of these mechanisms to parse sentences by computing the grammatical similarity — or more precisely, the grammatical substitutability — of relevant phrases.

## 2 Recursive analogical substitutions

Our parsing method is the byproduct of a solution to the problem of finding, given a possibly unattested phrase (string of words), the corpus phrases that can best predict its occurrence in arbitrary contexts, which we call its *analogical phrases*. We state the problem, describe our solution to it and then formulate our solution as a parsing method. We make a background assumption: we never come across words that did not occur in our training corpus.

Let $s$ be a possibly unattested phrase. Our task is to find corpus phrases $s'$ that are *substitutable* by $s$ to a high degree with respect to grammaticality, where the degree of substitutability of $s'$ by $s$ is defined as the degree of certainty with which we can assume that $s$ can grammatically occur in a context given that $s'$ can grammatically occur in that context.

We give our solution as a recursive algorithm. The base case is that $s$ occurs enough times in our corpus so that we can find the substitutable phrases $s'$ simply by comparing their distributions to $s$. Our preliminary solution to the base case is given in section 3, where we suggest a method for determining substitutability and demonstrate it on corpus data. For now, let us assume that we have solved the base case by using some method to return corpus phrases $s'$ with a high degree of substitutability by $s$. For the sake of conciseness, if $s'$ is substitutable by $s$ to a high degree, throughout this paper we will write $s' \Rightarrow s$ (expressing that we can "rewrite" $s'$ as $s$) and simply say that $s'$ is substitutable by $s$. (We will later define the degree of substitutability as a continuous variable.)

The recursive case is that $s$ is unattested or there is too little distributional information about it for our method above to find phrases other than $s$ itself that are substitutable by it. If $s$ is a single word, we just return it — there is no way for us to find other substitutable corpus phrases. Otherwise $s$ is a multi-word phrase: let us proceed to describe this case with $s =$ *my friend saw the rainbow* and illustrate the process in Figure 1 below.

We split $s$ up into two possible constituents $s_1 =$ *my friend* and $s_2 =$ *saw the rainbow*, and recursively call our algorithm on them to get corpus phrases $a_1 =$ *our teacher* and $a_2 =$ *heard it* such that $a_1 \Rightarrow s_1$ and $a_2 \Rightarrow s_2$. We then apply our base case algorithm to $a_1$ and $a_2$ to find corpus phrases $s'_1 =$ *she* and $s'_2 =$ *arrived* such that $s'_1 \Rightarrow a_1$, $s'_2 \Rightarrow a_2$ and $s'_1 s'_2$ occurs in the corpus. For each possible binary split of $s$ we collect these concatenations $s'_1 s'_2$ and return them as its analogical corpus phrases.

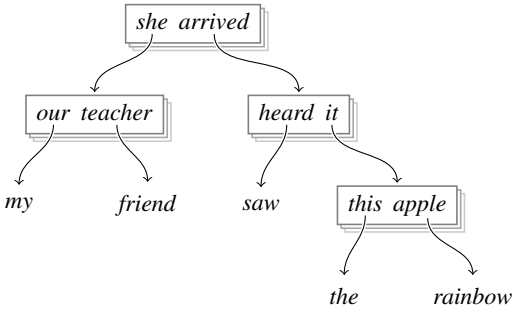


Figure 1: Finding analogical evidence for the sentence *my friend saw the rainbow* through recursive analogical substitutions. (Arrows indicate substitutability, e.g. *she* ⇒ *our teacher*; stacks indicate multiple corpus phrases.)

The assumptions that conceptually underlie the recursive process are that substitutability is *transitive* and *compatible* with the concatenation operation, the latter meaning that for any four strings $s_1, s_2, s'_1, s'_2$:

if $s'_1 \Rightarrow s_1$ and $s'_2 \Rightarrow s_2$, then $s'_1 s'_2 \Rightarrow s_1 s_2$.[1]

E.g. in Figure 1 we deduce *she* ⇒ *my friend* in two steps: (1) we use compatibility to deduce *our teacher* ⇒ *my friend* from *our* ⇒ *my* and *teacher* ⇒ *friend*, and (2) we use transitivity to deduce *she* ⇒ *my friend* from *she* ⇒ *our teacher* and *our teacher* ⇒ *my friend*.

[1] If we understood "⇒" as *full* substitutability, transitivity and compatibility could be proved instead of having to be assumed; but we understand "⇒" as a *high degree* of substitutability, in which case neither property can be proved.

### 2.1 Parsing with analogies

Figure 1 illustrates the inference process for a particular choice of bracketing for the sentence *my friend saw the rainbow*; we might find different analogical phrases if we split up e.g. the phrase *saw the rainbow* into the phrases *saw the* and *rainbow*. In fact, we predict that we would find *fewer* analogical phrases using this latter bracketing, since there is no word that is substitutable by the phrase *saw the* to the degree that e.g. the word *it* is substitutable by the phrase *the rainbow*.

In general, we can evaluate a binary bracketing of a phrase by the degrees of substitutability of the analogical phrases we can find through it, and define the parse tree of the phrase to be the bracketing that yields the best analogical phrases — in other words, the most successful path in the recursive analogy-finding process can be considered to be the syntactic structure of the phrase. Our theory of language thus gives rise to syntactic structure naturally, without resorting to stipulation: our thesis is that the syntactic structure of a phrase emerges from the process of analogical inference from exemplars.[2]

Let us define an evaluation method for parse trees. Given a phrase $s$, a corpus phrase $s'$, and a parse tree $T$ for $s$, we define the substitutability of $s'$ by $s$ according to $T$ recursively as the function $\mathcal{A}_T(s' \Rightarrow s)$. The base case is that $s$ is a frequently occurring phrase or a word, in which case we just take the substitutability of $s'$ by $s$ according to our substitutability function $D$ defined in section 3:

$$\mathcal{A}_T(s' \Rightarrow s) = D(s' \Rightarrow s) \tag{1}$$

In the recursive case, $s$ is an infrequently occurring multi-word phrase; then $s'$ must also be a multi-word phrase to function as an analogical phrase for $s$, due to the "compositional" way that our analogical phrase finding algorithm above works. For each bracketing $s'_1 s'_2 = s'$, we calculate the substitutabilities of $s'_1$ and $s'_2$ by any corpus phrases $a_1$ and $a_2$, then recursively calculate the substitutability of $a_1$ and $a_2$ by the left and right constituent phrases $s_1$ and $s_2$ (according to $T$) of $s$, and sum the products of these scores:

$$\mathcal{A}_T(s' \Rightarrow s) = \sum_{s'_1, s'_2, a_1, a_2} D(s'_1 \Rightarrow a_1) \cdot \mathcal{A}_T(a_1 \Rightarrow s_1) \cdot D(s'_2 \Rightarrow a_2) \cdot \mathcal{A}_T(a_2 \Rightarrow s_2) \tag{2}$$

[2]The best parse tree of a phrase may coincide with the standard syntactic structure associated with the phrase, as in Figure 1, or it may not.

(At least conceptually, $D$ will be defined in section 3 as a probability distribution, so that this formula can be interpreted as rewriting $s'$ as $s$ by independently rewriting its constituents $s'_1$ and $s'_2$ as $s_1$ and $s_2$; we thank Márton Makrai p.c. for this intuition.)

We finally define the goodness of a parse tree $T$ for a sentence $s$ as the sum of the substitutability scores of $s'$ by $s$ according to $T$ for all corpus phrases $s'$:

$$G(T, s) = \sum_{s'} \mathcal{A}_T(s' \Rightarrow s) \tag{3}$$

Assigning scores to parse trees in this manner is consonant with Bod's (2009, p. 760) notion of "maximizing structural analogy" between a novel sentence and exemplar sentences.

As implementations of our analogical phrase finding algorithm have only occasionally yielded intuitively acceptable parse trees, we will not present examples of the results here; we consider this algorithm a starting point for developing an explicitly defined exemplar-based theory of language.

## 3 Determining substitutability

Suppose that a phrase $s$ occurs with sufficient frequency in a corpus. We describe a method for determining the degree to which a corpus phrase $s'$ can be substituted by $s$. We first give a conceptual definition and then change it due to testing results.

We conceive of the substitution of $s'$ by $s$ as the conjunction of two independent events: (1) randomly drawing a context $c$ according to the conditional probability $P(c \mid s')$, and (2) randomly drawing the phrase $s$ according to the conditional probability $P(s \mid c)$ (where the conditional probabilities are the maximum likelihood estimates). We sum the joint probabilities of these events over all contexts $c$ to define the degree of substitutability of $s'$ by $s$:

$$D(s' \Rightarrow s) = \sum_{c} P(c \mid s') \cdot P(s \mid c) \tag{4}$$

If the contexts $c$ are non-overlapping, $D$ is a probability distribution in the mathematical sense; but we have not found a way to handle overlapping contexts, so we do not consider it a probability distribution. An important property of $D$ is that it is non-symmetric, which conforms to our intuition that substitutability is non-symmetric: *she* $\Rightarrow$ *our teacher* but *our teacher* $\not\Rightarrow$ *she*, since *I like our teacher* is well-formed but **I like she* is not.

In practice, we have found that computing $D$ as above tends to overestimate the substitutability of phrases that have similar distributions only on one side: e.g. *small* will be considered substitutable by *the* because they are both often followed by nouns. Since our parsing algorithm requires that analogical phrases be substitutable by their target phrases in arbitrary contexts, we will have to estimate "bilateral" substitutability differently.

A straightforward idea would be to estimate bilateral substitutability similarly to Walsh et al. (2010, p. 561), by separately calculating substitutability for only left contexts and for only right contexts, and then combining these scores, as illustrated in Figure 2. The question remaining is how to combine the scores.

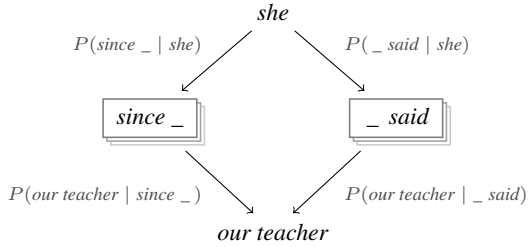


Figure 2: Computing the substitutability of *she* by *our teacher* using shared left and right contexts.

Surprisingly, when intuitively evaluating the analogical words found by the algorithm, we found that we get the best results when we not only take the minimum of the left and right degrees of substitutability, but also change the formula calculating these degrees so that it takes the minimum of the conditional probabilities that make up the paths from $s'$ to $s$ (instead of taking their product). That is, we define the degree of left or right substitutability of $s'$ by $s$ as follows, where $X$ is either the letter $L$ or the letter $R$, indicating that the contexts $c$ are either only left or only right contexts:

$$D_X(s' \Rightarrow s) = \sum_c \min\{P(c \mid s'), P(s \mid c)\} \quad (5)$$

And we redefine the degree of substitutability of $s'$ by $s$ as the minimum of their degrees of left and right substitutability:

$$D(s' \Rightarrow s) = \min\{D_L(s' \Rightarrow s), D_R(s' \Rightarrow s)\} \quad (6)$$

Our method thus becomes mathematically uninterpretable. However, when tested on a 1-million word corpus of excerpts from Project Gutenberg books collected by Peter Norvig (accessible here) we see that the best analogical words according to our method are in many cases not just grammatically but also semantically intuitively acceptable. Table 1 lists the five best analogical words found by our method for various words, with their frequencies in parentheses (degrees of substitutability are omitted because they are uninterpretable).

| ***walked* (98)** | ***think* (555)** | ***sofa* (81)** |
|---|---|---|
| *strode* (9) | *suppose* (57) | *chair* (126) |
| *stepped* (61) | *reflect* (15) | *couch* (14) |
| *went* (1005) | *say* (718) | *clavichord* (22) |
| *turned* (497) | *know* (1037) | *bed* (195) |
| *ran* (320) | *admit* (63) | *wagon* (22) |
| ***green* (53)** | ***unusual* (30)** | ***noise* (36)** |
| *gray* (74) | *peculiar* (82) | *sound* (214) |
| *white* (334) | *exceptional* (34) | *rattle* (15) |
| *blue* (109) | *ordinary* (99) | *sounds* (93) |
| *lilac* (6) | *extraordinary* (72) | *list* (44) |
| *brown* (54) | *important* (278) | *voice* (448) |

Table 1: Five best analogical words for various target words (in bold), with frequencies in parentheses.

Its mathematical uninterpretability notwithstanding, our method does suggest that when developing a measure of substitutability, we should aim to make it *conservative* in the sense that if there is a "weak link" in the process of substitution, other parts of the process should not be able to compensate for it — this may be why using the minimum yields intuitively good results even for relatively infrequent words of a relatively small corpus.

## 4 Conclusion

To conclude, we have outlined an exemplar-based parsing method that models syntactic structure as emerging from the process of recursively substituting unattested phrases by distributionally analogous corpus phrases. Although our current implementation rarely produces adequate full parses of sentences, we believe that with further research this method has the potential to yield insight into the nature of syntactic knowledge.

## References

Ben Ambridge. 2019. Against stored abstractions: A radical exemplar model of language acquisition. *First Language*, 40(5–6):509–559.

Rens Bod. 2009. From exemplar to grammar: A probabilistic analogybased model of language learning. *Cognitive Science*, 33(5):752–793.

Michael Walsh, Bernd Möbius, Travis Wade, and Hinrich Schütze. 2010. Multilevel exemplar theory. *Cognitive Science*, 34(4):537–582.