An Implementation and Evaluation of the Original Generalized Phrase Structure Grammar System Proposal

Alejandro Velez (amva13) **Famien Koko** (fakoko) **Yang Dai** (yangdai)

Abstract

We provide an implementation-based replication of the Context-Free Grammar rules proposed by Gerald Gazdar in his article *Unbounded Dependencies and Coordinate Structure*, a founding document of the grammatical framework which came to be known as Generalized Phrase Structure Grammar. We extend Gazdar's proposals for phrase structure grammars handling certain sentences Gazdar's system was unable to parse and analyze the limitations of the system. In particular, we analyze the ambiguity Gazdar's system introduces, even when considering sentences in the original paper, and postulate some of it to be due to his Generalized Left-Branch condition. The inability of our system to tackle some sentences is also analyzed.

1 Introduction

Consider this sentence, proposed by Gazdar:

The Dodgers beat the Red Sox and the Dodgers were beaten by the Giants.

This type of sentence is an obvious passive transformation of the simpler sentence:

The Dodgers beat the Red Sox and the Giants beat the Dodgers.

A generative grammar should be able to handle both of these. If we remove the use of transformation in our grammar, however, we are left with only phrase structure grammars capable of generating only context-free languages. Gazdar claims removing the transformation operation will not only retain the ability to parse these sentences, but also could help explain the fact that humans process language very rapidly.

In his paper, *Unbounded Dependencies and Coordinate Structure* (1981), Gazdar poses these, and several other sentences involving constituent coordination and unbounded dependencies. He proposes simple, general schemas using only phrase structure grammars to correctly parse positive and negative examples of these sentences. While his schemas are able to handle a large portion of the examples, there are some which he identifies would need more work to solve. Two examples of sentences which show where his schema fails are shown below:

***John, who and whose friends you saw, is a fool.**

John, to who and to whose friends that letter was addressed, is a fool.

Both of these sentences would be generated under his proposed grammar, but the first is ungrammatical.

In this project we implement a Context-Free Grammar (CFG), building on Gazdar's work in the original paper, which can handle constituent coordination as well as unbounded dependencies. Our grammar specifically tackles the 90 sentences consisting of 61 positive examples and 29 negative examples which Gazdar lists. Out of the positive examples, Gazdar claims his rules should handle 52 of them. The majority of rules for our grammar are taken directly from the rules and metarules specified by Gazdar. The terminal symbols are taken from the example sentences. In doing so, we provide two main research objectives

1. An implementation of Gazdar's Generalized Phrase Structure Grammar system.

2. An analysis of the system's ambiguity and other limitations.

We were able to successfully parse the 61 positive examples and correctly failed 14 out of the 29 negative examples. We found that the Generalized Left Branch Condition (GLBC) over-generalized; it generated incorrect parse trees for certain sentences and generated parses for many of the negative examples. As an improvement, we replaced GLBC with several new rules that generate more linguistically sound parse trees and over-generalize fewer of the negative examples. We found, however, that coming up with a CFG that did not generate any of the negative examples was very difficult and is possibly a limitation of CFG itself.

2 Motivation

The article *Unbounded Dependencies and Coordinate Structure* was a founding document of the grammatical framework which came to be knows as Generalized Phrase Structure Grammar (GPSG). The goal of GPSG was to show that the syntax of natural languages could properly be described utilizing context-free-grammars, CFGs (Gazdar, 1985). Since the article, many experts, including Gazdar, have argued CFGs cannot properly describe natural languages (`www.informatics.susx.ac.uk`). Our motivation for the project is twofold. First, we are interested in constructing a robust CFG to handle complex sentences. In this project we do so by improving on Gazdar's proposals. Second, we wish to better understand the limitations of CFGs in being able to describe English. This is achieved through the analysis of the limitations of our system.

3 Related Works

Several other works attempted to argue for the possibility of CFGs properly describing natural languages. For example, Pullum's 1983 work *Context-freeness and the Computer Processing of Human Languages* supports the argument that CFGs are indeed expressive enough to describe human language and further argues that CFGs may actually be overly powerful. (6)

Several works since Gazdar's paper was published have significantly improved upon Gazdar's handling of constituent coordination, which lacks a way to handle valid sentences that conjoin constituents of different syntactic categories. Sag et al. (1985) (5) demonstrates that Gazdar's earlier assumption on the likeness of syntactic categories in coordination is too strong. Gazdar's assumption that only phrases in the same syntactic category can be conjoined excludes many valid sentences. Examples given include:

Pat is either stupid or a liar. [AP or NP]

Pat is a Republican and proud of it. [NP and AP]

Pat is healthy and of sound mind. [AP and PP]

Pat is either asleep or at the office. [AP or PP]

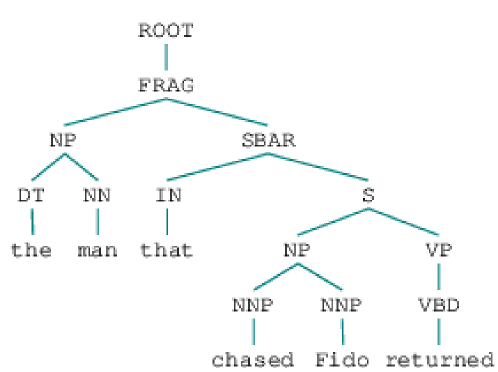
Sag's work handles coordination of unlike categories by making use of the representation of categories as feature-value pairs. Major categories (e.g. N, V, A, P) are represented as some combination of the binary features, N, V, BAR, and SUBJ. All other categories are extensions of the major categories, and a category, A, is defined as an extension of category only if the features of B is a subset of the features in A. With the use of extensible categories and features, Sag is able to build coordination rules by looking at syntactic function rather than just syntactic categories and is able to solve many of the issues of coordination in Gazdar's earlier paper.

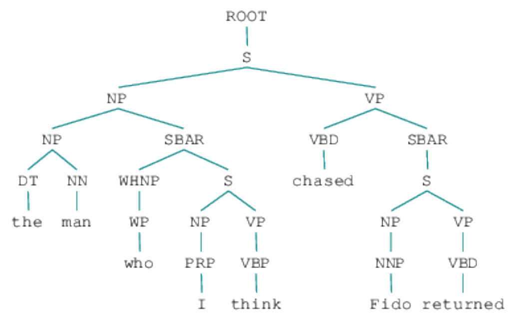
Additionally, Gawron et al. (1982) describes an implementation of GPSG that was linguistically based on Gazdar's work. Gawron specifically mentions that Gazdar's slash categories, which allows constituents with gaps to be treated as belonging in the same syntactic category as constituents with gaps, is able to successfully handle questions and relative clauses that were previously thought to be impossible within context-free grammars. (3)

Ristad (1987) proposes a revised GPSG that is more tractable and linguistically constrained and cites a previous paper, Ristad (1986), that shows that the metarule and feature components of GPSG can lead to computational intractability. (4)

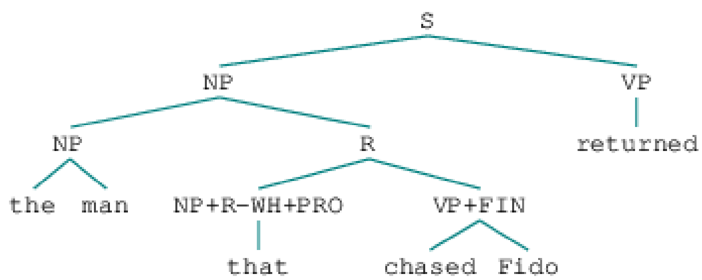
The works cited above show both successes and inadequacies of Gazdar's work and the use of context-free grammars to represent human language. The team has used these sources for ideas to build upon while implementing and critiquing Gazdar's context-free grammar.

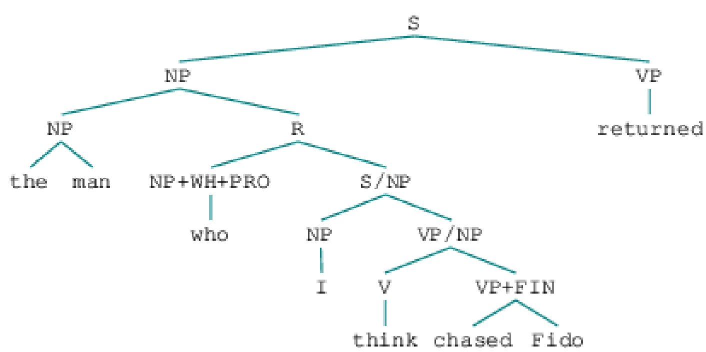
Additionally, since the peak of popularity for CFGs, several other parsing approaches have become very popular. The reknowned Stanford statistical parser makes use of lexicalized dependency parsing and is deemed to work rather well (https://nlp.stanford.edu/software/lex-parser.shtml). However, in many cases, CFGs are able to capture the semantics of parsed sentences better than such statistical parsers. For example, consider the following two parse trees generated by the Stanford parser for sentences we'll revisit later in the paper.





In both cases, the hierarchy of the parse tree output by the Stanford parser fails to adequately capture the semantics of the sentence. Compare them with the trees produced by our system:





The parse trees produced by our grammar capture the semantics of these sentences more appropriately than those produced by the Stanford parser. When building our system, we aimed to generate parse trees adequately capturing the semantics of the sentences parsed.

4 Methodology and Implementation

We engineered our grammar by, first, replicating Gazdar's context-free-grammar system and, then, improving on it by eliminating the Generalized Left-Branch Condition. We implemented Gazdar's 5 main proposals for tackling constituent coordination and unbounded dependencies. They were:

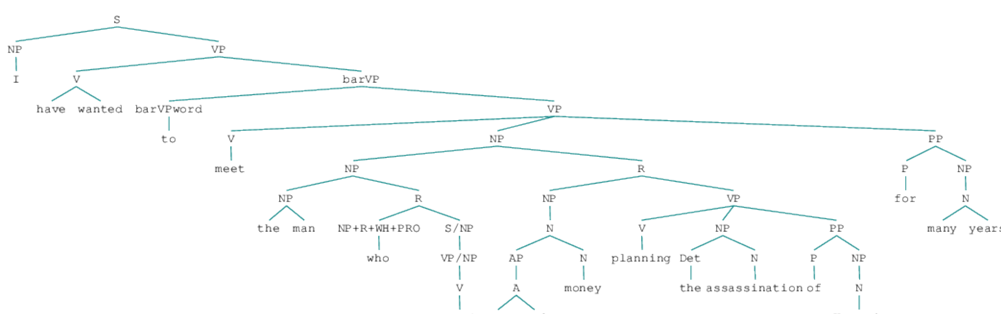
1. A schema for constituent coordination.
2. A schema allowing unbounded dependencies.
3. A generalization of Ross's Left Branch constraint.
4. A tensed VP analysis of subject dependencies.
5. A schema allowing rightward dependencies

In order to implement these proposals, we first created a base set of rules very similar to the base set proposed in Gazdar's article (Gazdar, 160). Then, we encoded Gazdar's proposed metarules into Python functions taking the base set of rules as input and appending the appropriate context-free grammar rules. This allowed for the parsing of the 10 sentences Gazdar explicitly parsed and displayed parse trees for in his paper. The team compared the obtained trees, shown in the Testing section of the paper, with Gazdar's and determined an adequate implementation of Gazdar's rules had been achieved. Our team was also able to generate all grammatical sentences discussed in Gazdar's paper.

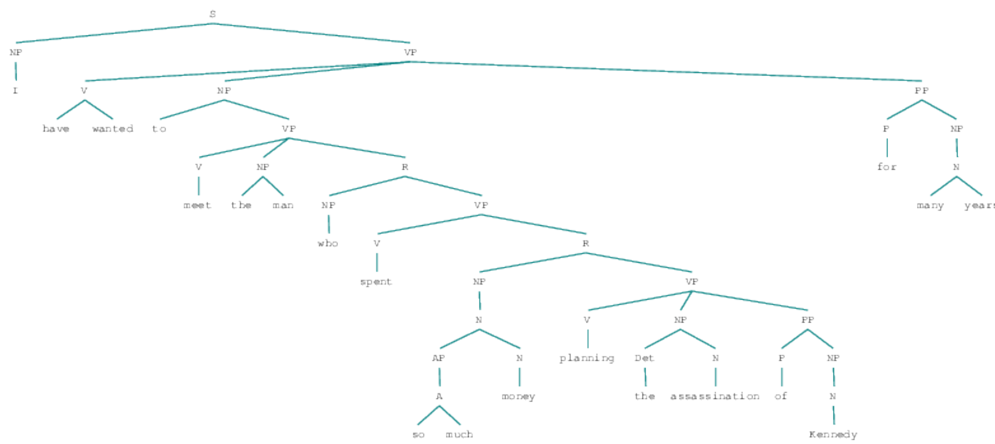
After implementing Gazdar's rules, our team tested our grammar against the sentences, both grammatical and ungrammatical, in Gazdar's paper. We had problems with over-generation and noticed undesirable parse trees in many of the grammatical sentences. Upon inspection, we traced the source of many of our problems to GLBC rules and, therefore, decided to eliminate it from our grammar. We believe this phenomenon is a specific case of a larger failure in Gazdar's proposal. Namely, when he evaluates the parsing or non-parsing of a sentence, he considers only a subset of the rules at any given point. In many cases, Gazdar fails to account for the interactions between all his rules for a given sentence. One such example is presented below with others being presented later in the paper.

4.1 Generalized Left Branch Condition and Ambiguity

Gazdar's Generalized Left-Branch Condition results in several introductions of ambiguity into the system. The sentence below is an example of a parse utilizing the GLBC.



This is clearly not the desired parse for the sentence but it is being enabled by the GLBC. Notice the 'for many years' phrase is paired as a prepositional phrase modifying the verb 'meet'. The desired parse would have it modifying the verb-phrase 'have wanted'. Furthermore, the noun-phrase 'so much money' is not being paired as the patient for the action 'spent'. Specifically, the desired tree would look like the below tree:



This parse tree was generated after removing the GLBC from the grammar rule set. We determined the GLBC was responsible for several examples of over-generation. Furthermore, it introduced many ambiguous cases of sentences being parsed with semantically incorrect parse trees. For these reasons, we removed rules derived from GLBC and adjusted rules in our grammar to continue parsing all grammatical sentences in Gazdar's article. With this change, we were able to improve the parse tree structures generated by our system and improve on the over-generation problems associated with the GLBC.

5 Testing

For testing, we wish to assert all of Gazdar's proposed rules are implemented correctly. Our main method of testing consisted of using the 10 parse trees provided in the article as a gold standard for comparison. Since our grammar was able to generate parse trees identical to Gazdar's we consider it to be an adequate implementation of Gazdar's proposals. What follows is a display of the rules proposed by Gazdar and the parse trees obtained upon implementation. These trees were compared with Gazdar's presented parse trees when evaluating our initial system. The symbols used in our grammar can be found in the appendix.

5.1 Testing Rule Implementation

To test our implementation of Gazdar's rules, we compared our parse trees with those presented by Gazdar for each rule. Gazdar's first rules tackle problems of constituent coordination and consist of the following 2 metarules:

$$(1)\alpha \rightarrow \alpha_1, ..., \alpha_{n[\beta]}$$
$$(2)\alpha_{[\beta]} \rightarrow \beta\alpha$$

Where β is a conjunction α is any syntactic category.

He also proposes a set of derived rules as follows from this meta-rule:

$$(3)\ D(\beta, G) = \{[_{\alpha/\beta}\ \sigma_1\ ...\ \sigma_i/\beta\ ...\ \sigma_n] : \alpha \rightarrow [\sigma_1\ ...\ \sigma_i\ ...\ \sigma_n]$$

Where β, α, and σ are any syntactic category.

He provides no specific parses for any sentences derived from these rules but does provide diagrams for potential subtrees derived from these rules. We used these to judge the accuracy of the first set of parse trees.

The sentences analyzed in this portion were:

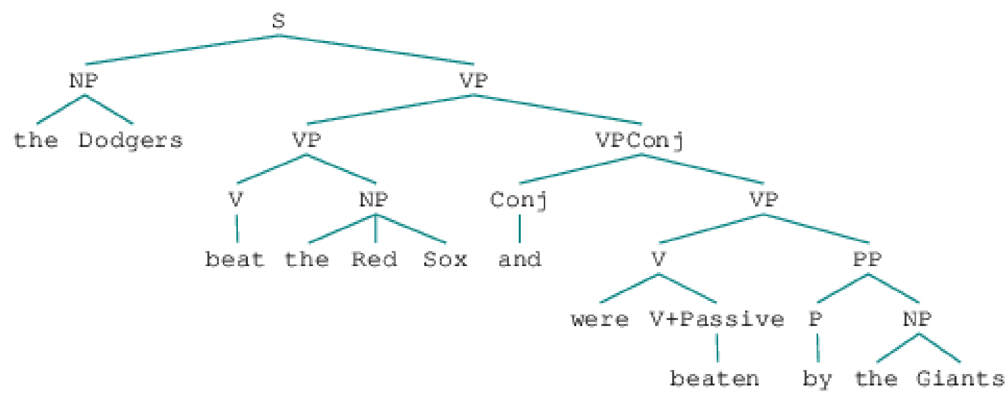
The Dodgers beat the Red Sox and were beaten by the Giants.

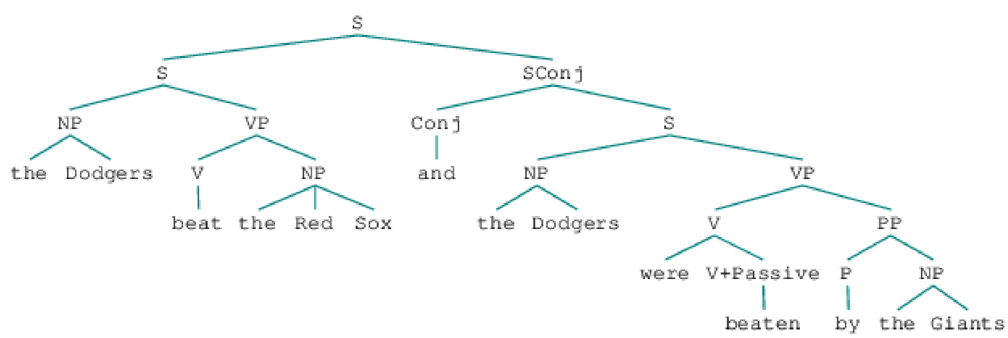
The Dodgers beat the Red Sox and The Dodgers were beaten by the Giants.

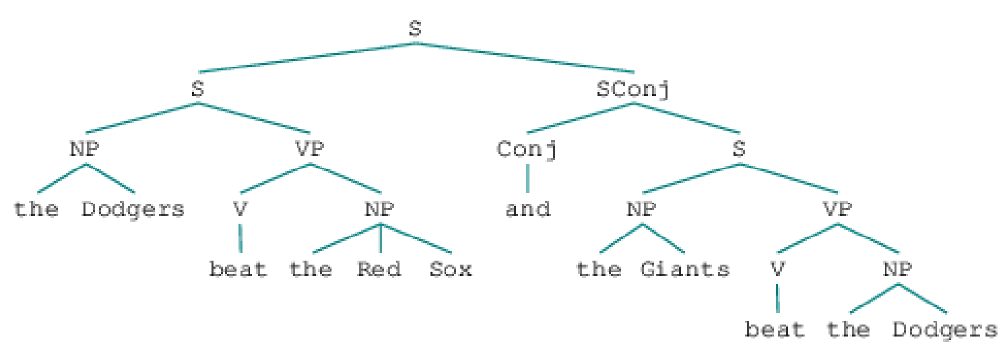
The Dodgers beat the Red Sox and The Giants beat the Dodgers.

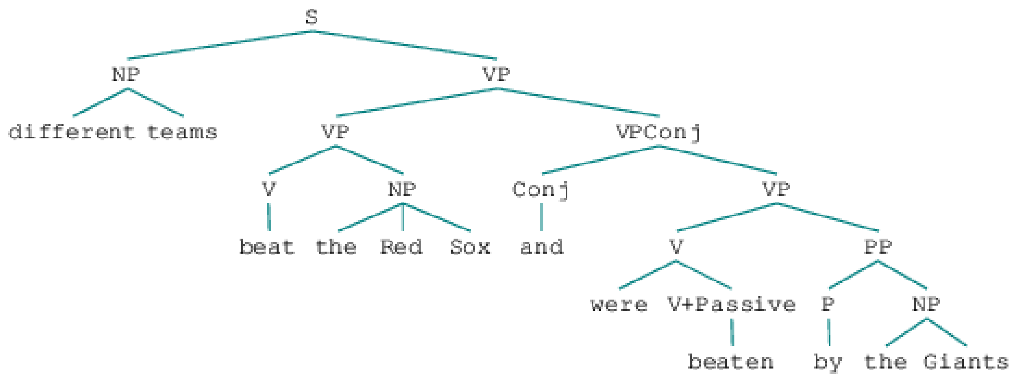
Different teams beat the Red Sox and were beaten by the Giants.

Our system produced the following parse trees:









Here Gazdar's rules for coordination between 'like' syntactic categories is employed and its effectiveness at handling these sentences is noticeable. Gazdar then introduces the Generalized Left Branch Condition (GLBC) seen below:

$$(4)\alpha/\beta \rightarrow \sigma/\beta$$

where α and σ are any node labels and β is NP.

As discussed previously, GLBC was tested and removed from our final grammar. GLBC is clearly a very bold rule. Furthermore, it adds large degrees of ambiguity to many parses in the system and, as will be discussed, was responsible for many over-generations. A set of linking rules was also added to the grammar. They are described by the metarule:

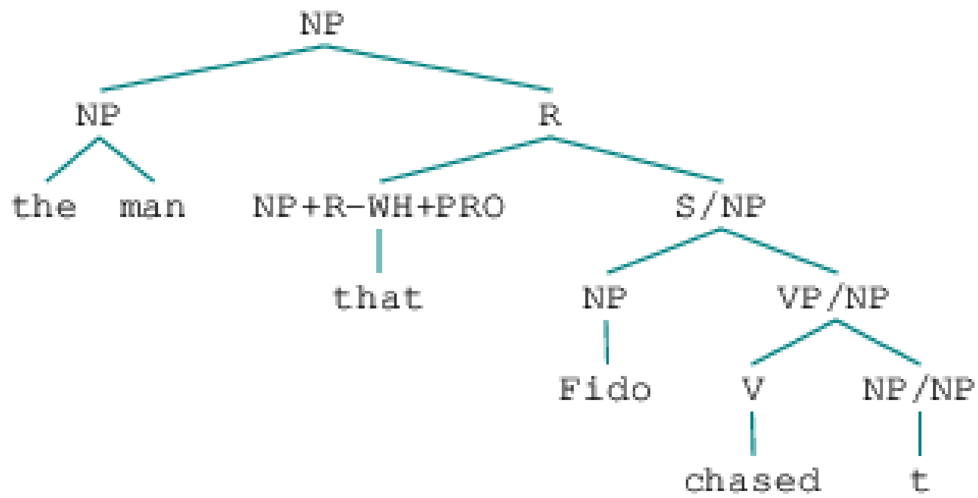
$$(5)\alpha/\alpha \rightarrow t$$

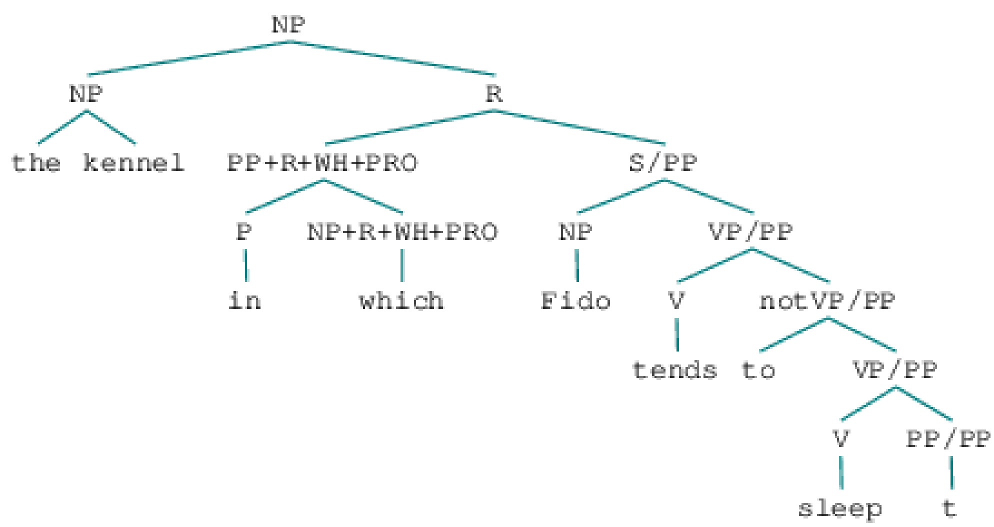
Here we use the lone letter t to represent a trace. Gazdar also defines sentential categories to dominate the behavior of relative clauses. These are: C(omplement), R(elative), and Q(interrogative) (Gazdar, 162). We added rules for Relative and interrogative clauses in our grammar. Using these rules, the following noun-phrases can be generated:

the man that Fido chased t .

the kennel in which Fido tends to sleep t .

For which we obtained the trees:

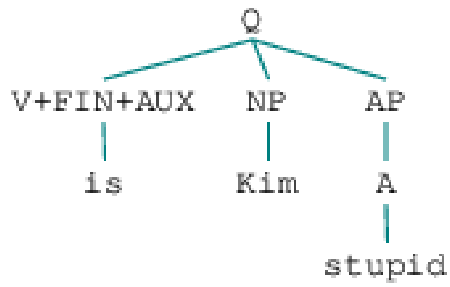




A rule for Subject-Auxiliary inversion in English is then introduced.

$$(6)\ VP[+FIN + AUX] \rightarrow V\ X \implies Q \rightarrow V[+FIN + AUX]\ NP\ X$$

Implementing this rule in our grammar allows us to generate the following tree for an interrogative clause.

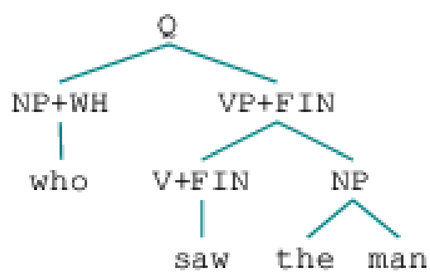


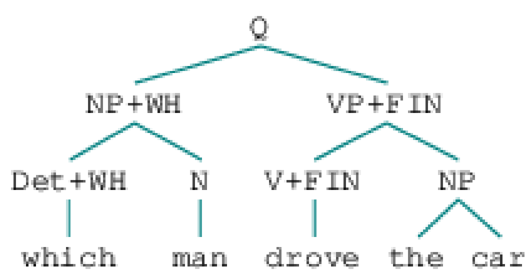
Two metarules for root constituency interrogatives are then added to the grammar. These are:

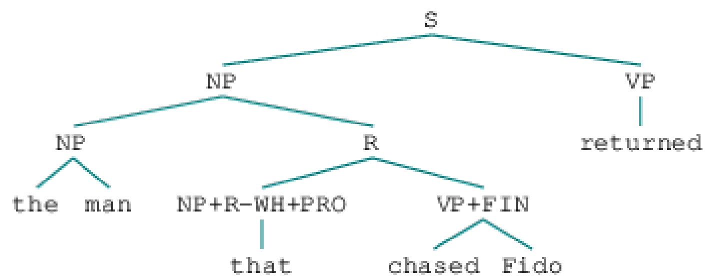
$$(7)\ \mathrm{Q} \rightarrow \alpha[\mathrm{+WH}]\ \mathrm{Q}/\alpha$$
$$\text{where } \alpha \ \epsilon\ \{\mathrm{NP, PP, AP, AdvP}\}$$
$$(8)\ \alpha \rightarrow \mathrm{X} \sum[-C]/\mathrm{NP}\ ...\ \Longrightarrow\ \ \alpha \rightarrow \mathrm{X\ VP[+FIN]}\ ...$$

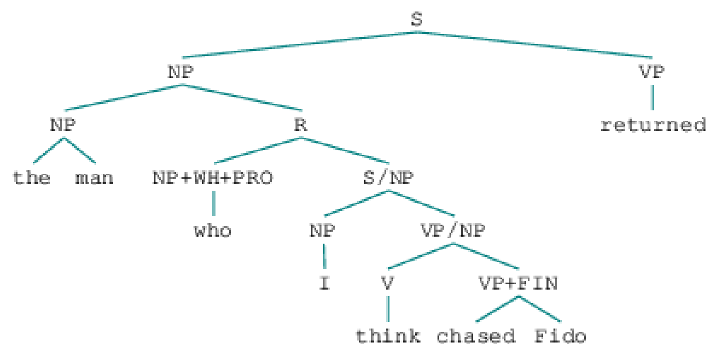
where X is any major category symbol, α is anything and $\sum$ ranges over sentential categories

and they allow for the generation of the following trees:







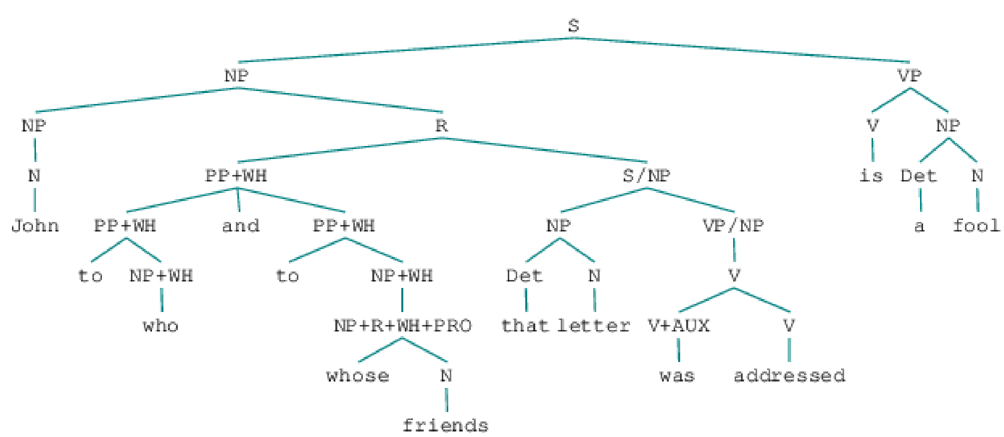


With this, Gazdar's schema for leftward dependencies is complete. However, Gazdar notes several exceptions his rules are unable to handle. For example, the difference between the following two sentences, the first ungrammatical and the second grammatical, is not captured as they are both generated.

John, who and whose friends you saw, is a fool.

John, to who and to whose friends that letter was addressed, is a fool.

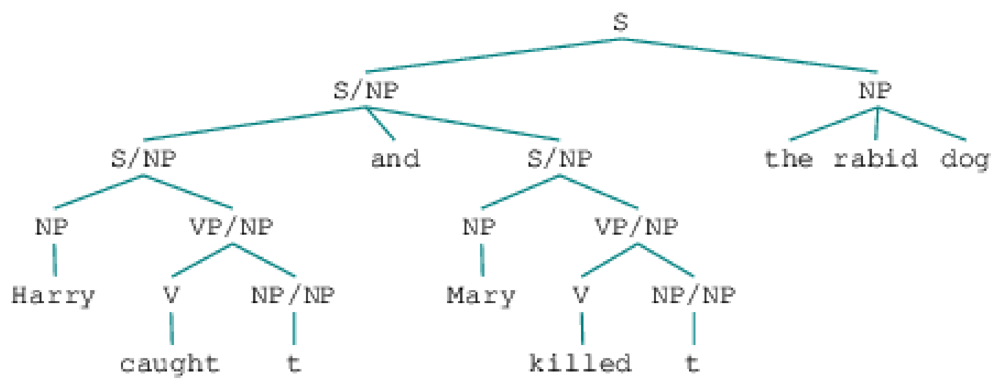
Our team was able to engineer a grammar accounting for this difference by improving on rules for relative clauses and who-forms to account for noun-phrases pairing "whose" with a noun. We properly rejected the first sentence. A tree for the second sentence can be seen below.



In addition to rules for leftward dependencies, Gazdar proposed a schema for rightward dependencies summarized by the following metarule.

(9) $\alpha \rightarrow \alpha/\beta \;\; \beta$
where α ranges over the clausal categories and β ranges over clausal and phrasal categories.

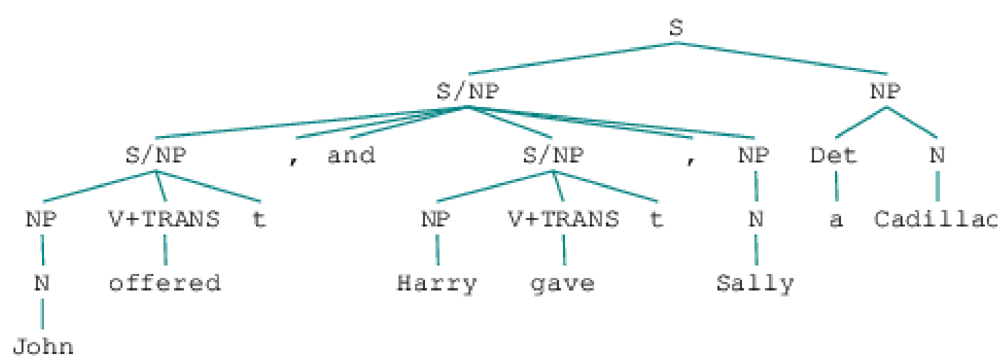
Implementing this schema allows for the generation of the following trees:



However, this schema fails to take into account that constituents are not necessarily present only in the right-most β position. As such, it fails to generate the following sentence, as noted by Gazdar:

John offered, and Harry gave, Sally a Cadillac.

Our grammar was able to parse this sentence and achieve a desirable parse tree. However, we are unsure about whether such a rule allowing for more than one right-branch is the general solution to the problem or about overgeneration resulting from this rule. No over-generation resulting from this rule was detected in the Gazdar corpus. The parse tree for this sentence can be seen below.



6 Evaluation

In his paper, Gazdar gives 90 example sentences. Of these, he claims 61 are grammatical and 29 are not. We evaluated our system by measuring our grammar's ability to parse the 61 grammatical sentences and reject the 29 ungrammatical sentences. Since replication and improvement of Gazdar's system is the goal of this paper, comparison with Gazdar's system is the chosen method of evaluation. Our system is able to correctly parse all of the grammatical examples (including those Gazdar himself was unable to parse), and correctly rejects 14 of the 29 negative examples. Below is a discussion on our problems with over-generation.

6.1 Overgeneration

Gazdar lists 29 different ungrammatical sentences in his paper. Our system correctly rejects 14 of them. The other 15 broadly fall into 2 categories: sentences that we believe are actually grammatical, and actually should be parsed by the grammar, and those which are ungrammatical and, therefore, our system should reject.

6.1.1 Grammatical sentences

First we examine sentences which we think are correct but Gazdar labels as ungrammatical. One class of these sentences arises because of Gazdar's "likeness" assumption when dealing with constituent coordination. As noted previously, this assumption has been debunked and we've chosen not to make this assumption. We, therefore, believe our system is behaving appropriately with these sentences despite achieving a result different from Gazdar's proposal. This accounts for sentences such as:

I wonder who Bill liked and saw Mary.

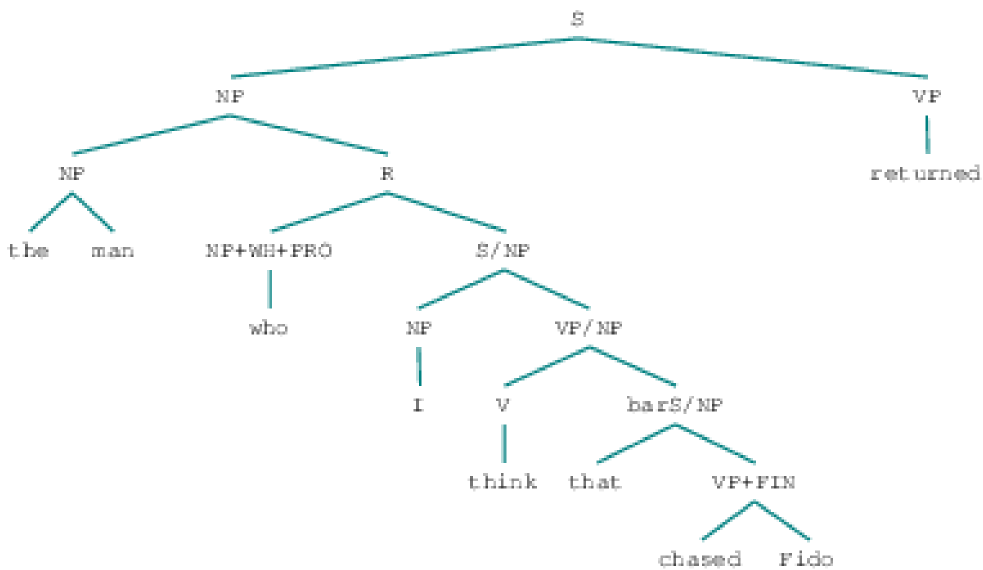
Gazdar rules out sentences such as these because it conjoins unlike syntactic categories, Bill liked and saw Mary, (S/NP & VP respectively). We believe sentences can conjoin unlike categories grammatically, and thus accept these sentences as valid, and our system parses them as such.

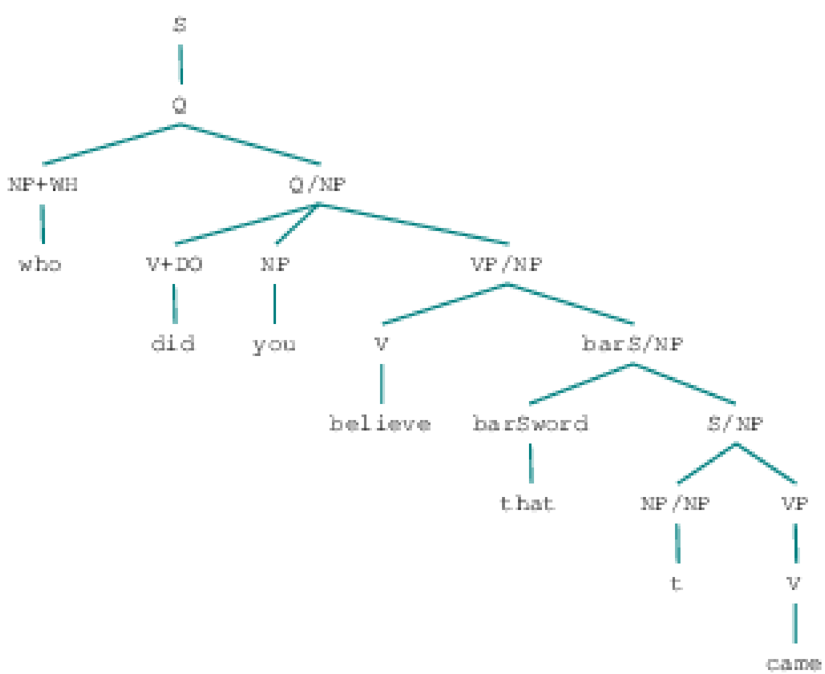
Another class of sentences deals with X-bar sentences including bar words such as that, for, and whether. A few examples are:

The man who I think that chased Fido returned.

Who did you believe that came.

The parse trees for these two sentences are shown below.





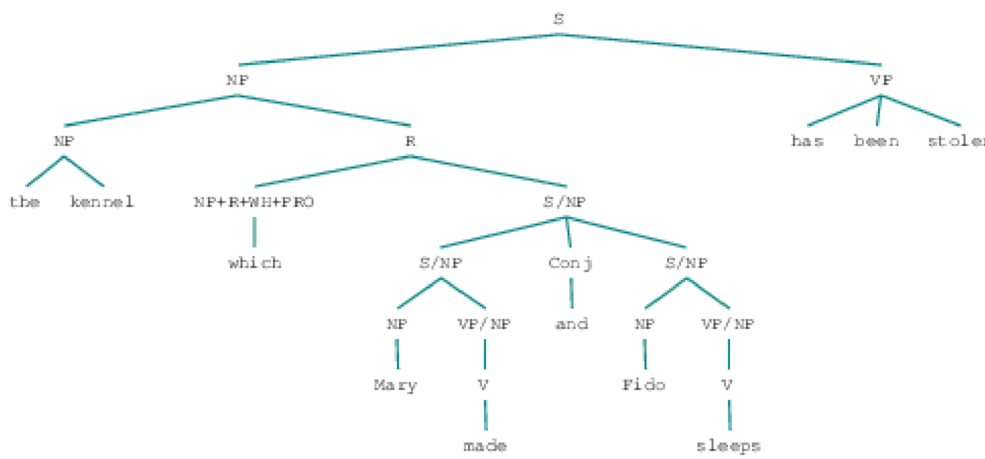
We decided to label these as grammatically correct since they made sense to us, and the semantics of the parse also made sense linguistically.

A list of all the sentences we decided were actually grammatical can be found in the appendix.

6.1.2 Generated ungrammatical sentences

The ungrammatical sentences in Gazdar's proposal which our grammar generated were:

The kennel which Mary made and Fido sleeps has been stolen



This first sentence we parse, mostly because we did not have a sophisticated system of verb types. Our focus was mostly on high level rules and correct semantics, and thus we did not have a large array of verb type differentiation. In this case, our system does not know that made is a transitive verb while sleeps is not. It would not be difficult, but likely very time consuming, to extend the system to support this; we would need to specialize our rules based on different verb types (transitive, intransitive, number of arguments, future, present, past tense etc).

John is easy to please and to love Mary

The issue we had with this sentence is handling lack of coordination between the infinitive verb-phrase and 'love Mary'. We believe with more specialized rules for verb-phrase coordination may allow our system to reject this.

the man chased Fido returned

This sentence is parsed based on the set of Gazdar's derived rules from VP (Gazdar, 1981, p.160). Specifically, the derivation meta rule, rule (3) in the Testing section, led us to create the rule, S/VP $\rightarrow$ NP/VP VP. This rule allows back-to-back verb-phrases when combined with the right-branch meta rule. We believe the interaction of these two rules is prone to overgeneralizing. It would be worthwhile to investigate whether the derivation meta rule proposed by Gazdar can be applied across all phrasal categories.

6.2 Grammar Size

Since Gazdar's grammar is not implemented in the article, we were unable to compare our implementation to an alternative in terms of grammar size. However, we attempted to use the minimal amount of symbols while still achieving our desired functionality. All in all, our grammar consisted of a total of 108 symbols. This is a large grammar size, but should be expected as each feature and derived symbol (ie. VP/NP) is a new symbol.

7 Conclusions

Our team successfully replicated and improved upon Gazdar's originally proposed Generalized Phrase Structure Grammar via the implementation of a context-free grammar with rules derived from Gazdar's proposals. The major improvements were:

1. reduction in grammar ambiguity and over-generation through the elimination of the Generalized Left-Branch condition.

2. an extension of Gazdar's proposals to tackle problems noted by Gazdar with left- and rightward dependencies

We successfully parsed all grammatical sentences posed by Gazdar, including those he was unable to parse. Though we over-generated some ungrammatical sentences posed in the article, many of these parses used rules Gazdar himself proposed and cannot, therefore, be used as a basis of comparison. Rather, this reveals a problem of over-generation present in the system as proposed. On a related matter, some ambiguity is present in possible parses for some of the sentences in Gazdar's corpus. However, both over-generation and ambiguity are problems inherent to the CFG approach itself and, so, our work shows the limitations of CFGs when attempting to adequately describe natural languages. As next steps, it would be worthwhile to extend our system's treatment of coordination between verb-phrases and analyze the over-generation resulting from rule (3). This would help further reduce our grammar's over-generation and ambiguity.

8 Acknowledgements

We thank Professor Robert Berwick and the TAs of 6.863 for their mentorship throughout the project and for a great semester.

9 Grammar Availability

Our grammar is available on github as 'GRAMMAR.gr' in the following github repository:
`https://github.com/Famien/artiste-final-project/tree/master` For proper functionality, the start symbol of one's parser must be set to 'S'. Furthermore, the grammar format is that used for the competitive grammar writing competition of MIT 6.863: Natural Language Processing.

10 Appendix

Below is the table of symbols used in our grammar.

Symbol	Entity
S	sentence
Q	interrogative
R	relative
C	complement
barS, notS	sentence (special-form)
NP	noun-phrase
VP	verb-phrase
barVP, notVP	verb-phrase (special-form)
PP	prepositional-phrase
Conj	conjunction
AdvP	adverbial-phrase
AP	adjective-phrase
P, PRP	preposition
Det	determiner
Adv	adverb
A	adjective
N	noun
V	verb
MD	modal
DO	do-form
BE	be-form
CAN	can or cannot
PRO	pronoun
FIN	finite-form
Passive	passive-form
TRANS	transitive
AUX	auxiliary
WH	wh-form

Below are the sentences that Gazdar labeled as ungrammatical but we believe are actually grammatical:

1. the man I think that chased Fido returned
2. who did you believe that came
3. the man who I think that chased Fido returned
4. who did you believe that came
5. who did you wonder whether came
6. who did you wonder if came
7. John asked who and what bought
8. the table that I put Kim on surprised Kim
9. John saw more horses than Bill saw cows or Pete talked to
10. I know a man who Bill saw and liked Mary
11. I wonder who Bill saw and liked Mary
12. John offered and Harry gave Sally a Cadillac

References

1. Gazdar, G. 1981. Unbounded Dependencies and Coordinate Structure, Linguistic Inquiry, Vol. 12, No. 2 (Spring, 1981), pp. 155-184

2. Gazdar, Gerald; Ewan H. Klein; Geoffrey K. Pullum; Ivan A. Sag (1985). Generalized Phrase Structure Grammar. Oxford: Blackwell, and Cambridge, MA: Harvard University Press

3. Gawron J.M., King J., Lamping J., Loebner E., Paulson E.A., Pullum G.K., Sag I.A., Wasow T. Processing English with a Generalized Phrase Structure Grammar. Computer Research Center, Hewlett Packard Company.

4. Ristad, E. 1987. Revised Generalized Phrase Structure Grammar. MIT Artificial Intelligence Lab.

5. Sag I. A., Gazdar G., Wasow T., Weisler S. 1985. Coordination and How to Distinguish Categories, Natural Language and Linguistic Theory 3, pp. 117-171.

6. Pullum G. 1983. Context-freeness and the Computer Processing of Human Languages, ACL '83 Proceedings of the 21st annual meeting on Association for Computational Linguistics, pp. 1-6.

7. "Department of Informatics : University of Sussex". www.informatics.susx.ac.uk.

8. Williams, E. S. 1978. Across-the-Board Rule Application, Linguistic Inquiry 9, 31-43.