

# Computational and Psycholinguistic Approaches to Structural Ambiguity: The Case of Garden Path Sentences

---

**Bajac, Petra**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, University of Zagreb, Faculty of Humanities and Social Sciences / Sveučilište u Zagrebu, Filozofski fakultet**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:131:051663>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom](#).

*Download date / Datum preuzimanja:* **2024-12-09**



Sveučilište u Zagrebu  
Filozofski fakultet  
University of Zagreb  
Faculty of Humanities  
and Social Sciences

*Repository / Repozitorij:*

[ODRAZ - open repository of the University of Zagreb  
Faculty of Humanities and Social Sciences](#)



UNIVERSITY OF ZAGREB  
FACULTY OF HUMANITIES AND SOCIAL SCIENCES  
DEPARTMENT OF ENGLISH  
ACADEMIC YEAR 2019/2020

Petra Bajac

**Computational and Psycholinguistic Approaches to  
Structural Ambiguity: The Case of Garden Path Sentences**

Master's thesis

Thesis advisor: dr. sc. Irena Zovko Dinković, Assoc. Prof.

Zagreb, June 2020

## **Abstract**

By approaching the phenomenon of structural ambiguity from two different perspectives – psycholinguistic and computational, this thesis shows how linguistic research can have practical applications in improving NLP systems. The analyses of specific sentences show breakdowns in processing, present the possible explanations for the reasons behind them through principles such as Minimal Attachment/Late closure and Lexical preference, and demonstrate the backtracking needed in order to achieve a full, successful parsing of ambiguous sentences. The types of sentences chosen for this thesis are called garden-path sentences, which induce a lot of difficulty in processing for both humans and machines, making them a perfect choice to demonstrate the similarities and differences between sentence processing in humans and machines. The research employs a combination of computational methods, both rule-based and statistical, and psycholinguistic hypotheses to explain the parsing process of garden-path sentences. The results show that in order for an NLP system to fully process highly ambiguous sentences such as these, it needs theoretical input to repair the partly parsed structures and successfully complete the parsing process. This implies a strong need for multidisciplinary research involving programmers, linguists, and cognitive scientists to succeed in emulating human intelligence in complex AI systems.

Keywords: Computational linguistics, psycholinguistics, garden-path sentences, natural language processing, multidisciplinary research

# Table of contents

<b>Abstract.....</b>	<b>1</b>
<b>Table of contents .....</b>	<b>2</b>
<b>List of tables.....</b>	<b>3</b>
<b>List of schematics .....</b>	<b>4</b>
<b>1 Introduction.....</b>	<b>5</b>
<b>2 Theoretical framework.....</b>	<b>7</b>
<b>2.1 Psycholinguistic approaches.....</b>	<b>8</b>
2.1.1 Right Association.....	8
2.1.2 Minimal Attachment and Late Closure.....	8
2.1.3 Lexical Preference .....	11
2.1.4 Parallel vs. Serial Processing and Competition .....	14
2.1.5 Reanalysis: Reparsing and Repair .....	15
<b>2.2 Computational approaches .....</b>	<b>18</b>
2.2.1 The Sausage Machine .....	18
2.2.2 Race-based Parsing .....	19
2.2.3 Augmented Transition Networks.....	21
2.2.4 Machine Learning and Frequency Based POS-tagging .....	23
<b>3 Methodology .....</b>	<b>26</b>
<b>4 Research and analysis.....</b>	<b>28</b>
<b>4.1 Race-based Parsing and Minimal Attachment/Right Association.....</b>	<b>28</b>
<b>4.2 Augmented Transition Networks and Minimal Attachment/Right Association ....</b>	<b>30</b>
<b>4.3 Augmented Transition Networks and Lexical Preference .....</b>	<b>33</b>
<b>4.4 Augmented Transition Networks and Machine Learning .....</b>	<b>38</b>

<b>5</b>	<b>Synthesis.....</b>	<b>44</b>
<b>6</b>	<b>Conclusion .....</b>	<b>49</b>
<b>7</b>	<b>Appendix.....</b>	<b>50</b>
	<b>7.1 List of analyzed sentences.....</b>	<b>50</b>
<b>8</b>	<b>References.....</b>	<b>51</b>
<b>9</b>	<b>Resources .....</b>	<b>53</b>

## List of tables

Table 1: Sample Grammar – adapted from Kempen (1994).....	32
Table 2: Production rules for (11), adapted from Du & Yu (2012).....	33
Table 3: Production rules for (12), adapted from Du & Yu (2012).....	35
Table 4: Different POS frequencies of <i>record</i> .....	38
Table 5: Production rules for (13), adapted from Du, Yu & Li (2014) .....	39
Table 6: Chi-square test for <i>houses</i> , adapted from Du, Yu & Li (2014) .....	42
Table 7: Production rules for (14).....	44
Table 8: Different POS frequencies of <i>man</i> .....	47

## List of schematics

Tree Schematic 1: Since she [jogs a mile] as a direct object .....	10
Tree Schematic 2: [Since she jogs] [a mile] as a subject of the second clause.....	10
Tree Schematic 3: [ <i>The boy got</i> ] <i>Fat</i> as an adjective.....	12
Tree Schematic 4: [ <i>The boy got</i> ] <i>Fat</i> as a noun .....	12
Schematic 5: Race-based Parsing System Overview - adapted from McRoy & Hirst (1990) .....	20
Schematic 6: ATN Overview – taken from Kempen (1994) .....	22
Tree Schematic 7: option a) <i>to</i> as a start of a PP modifier to the noun <i>letter</i> .....	29
Tree Schematic 8: option d) <i>to</i> as start of a PP denoting a BENEFICIARY to the verb <i>read</i> .....	29
Schematic 9: A new NP <sub>0</sub> replacing the default NP <sub>0</sub> – taken from Kempen (1994).....	31
Schematic 10: ATN for (11) – taken from Du & Yu (2012) .....	34
Schematic 11: ATN for (12) – taken from Du & Yu (2012) .....	36
Schematic 13: ATN for (13) – taken from Du, Yu & Li (2014).....	41
Tree Schematic 14: <i>The old man the boat</i> , with <i>man</i> as a noun .....	45
Tree Schematic 15: <i>The old man the boat</i> , with <i>man</i> as a verb .....	46
Schematic 16: Processing flowchart for (14).....	48

# 1 Introduction

Language is such a complex phenomenon that it is not enough to just have linguistics as a science and simply say that it deals with language, full stop. Over the course of its development and its path to become a fully recognized scientific field, linguistics felt the need to diverge into numerous different areas. There is sociolinguistics, psycholinguistics, historical linguistics, pragmatics, to name but a few. Each field focuses on a certain aspect of language, looking at it from its own unique perspective. And rightfully so. Different issues deserve, and require different analyses and more often than not, these very specific points of view manage to highlight what others cannot. These subfields connect linguistics with other relevant scientific fields: psychology, sociology, anthropology, philosophy etc. Together, they form a vast interconnected universe, all revolving around and connected by one thing - language.

In the past several decades, a new member appeared in this intricate network of disciplines: computational linguistics. It has brought many other connections to the playground: mathematics, statistics, artificial intelligence, computer science etc. Today, computational approaches have worked their way into other linguistic disciplines and become so widespread that now we even have computational psycholinguistics, computational syntax, computational sociolinguistics, computational semantics etc. Everything has become computational. It seems that computers represent another point of convergence in the linguistic world. This is not at all surprising, though. There are two ways, or rather two reasons why computers represent a point of contact. One is a simple fact of efficiency. When you have a lot of data to go through, it would be excellent if you had a computational model which could do the robust analysis for you. That is why we have parsers<sup>1</sup>, valency frames extraction and many other useful tools which help linguists with their analyses. This point of view puts computers in the service of linguists, so to say. Also, when creating such models, researchers are forced to think in a succinct and precise way, because otherwise, it would be very difficult to create an efficient computer model. This ensures that theories themselves are as simple as possible, and only as elaborate as they need to be. The other

---

<sup>1</sup> The term *parser* has slightly different meanings across different areas of linguistics. In psycholinguistics, it usually refers to the way humans analyze and comprehend sentence structures. In computational linguistics and natural language processing (NLP), a *parser* is an algorithm or a program whose job is to analyze a sentence or a phrase into its constituents.

perspective, however, does things the other way around. Linguists – their theories and models – serve the computers. It may sound strange, but it is true. Today, when there is a lot of demand for products that deal with natural language, it is the job of linguistics to provide good models and good solutions to natural language processing problems in order to create commercial tools and applications such as Google translate, Siri, spellcheck and auto-correct programs, and other products or services used by the general public. This thesis takes a kind of a hybrid path.

An interesting challenge to both linguists, from a theoretical point of view, and developers, from a practical point of view, is the phenomenon of structural ambiguity. Structural ambiguity in a sentence means that it has more than one possible interpretation. There are many instances of structural ambiguity, some as simple as determining who has the telescope in the sentence *He saw a man with the telescope*. This thesis focuses on the more peculiar case of the so-called garden-path sentences, where the analysis of structural ambiguity is not so easily recognized or resolved. These sentences are interesting not only because they are a fun word play, but also because they are a motherload of some very fascinating psycholinguistic phenomena: how humans process ambiguous sentences and why we do it in the way we do, as well as how we recover from misanalysis to get to the right interpretation. And while garden-path sentences are a brilliant example of the intricacies of *human* sentence processing, another interesting question is how *computers* process such sentences. This thesis explores the phenomenon of garden-path sentences within the wider context of structural ambiguity, comparing psycholinguistic and computational approaches to their analysis. It presents the existing psycholinguistic research regarding the topic and then analyzes those same select sentences using a computational approach. The computational perspective utilizes the two basic methods used in computational linguistics: rule-based and statistical approaches. The goal is to see how similar (or how different) the processes of parsing garden-path sentences are between humans and computers. There are other questions that need to be answered: how can we use psycholinguistic research to modify or create entirely new ways of processing natural language? Should the human mind and human language processing models serve as a template in the creation of more sophisticated NLP systems? Should artificial intelligence mimic our own? Is all that even possible, or do humans and computers simply “think” differently? These questions remain an open scientific challenge for many disciplines and the scope of this thesis can only begin to scratch the surface.



## 2 Theoretical framework

A garden-path sentence is a kind of an ambiguous sentence which people generally have trouble analyzing (i.e. understanding) the first time reading it, for example *Fat people eat accumulates*. At first sight, these sentences strike the reader as ungrammatical, but they are in fact perfectly correct. Marcus (1980) defines garden-path sentences as those sentences “which have perfectly acceptable syntactic structures, yet which many readers initially attempt to analyze as some sort of construction, i.e. sentences lead the reader ‘down the garden path’.” (202) The first analyses of garden-paths are accredited to Thomas Bever and his work in *The Cognitive Basis for Linguistic Structures* (1970).

The following sections of this chapter provide an overview of theoretical research done in the area of psycholinguistics and also present certain computational models which give insight into how the garden-path phenomenon is dealt with in the field of natural language processing. Given the fact that the analysis of structural ambiguity and the garden-path phenomenon is a very popular research topic among psycholinguists, only the most prominent theories and findings dealing with this topic will be covered. The same principle of selection will apply to theories of natural language processing described in this thesis.

## 2.1 Psycholinguistic approaches

In this section, some of the most prominent approaches to structural ambiguity and garden-path sentences in particular are described, covering both syntactically and lexically oriented theories: Right Association, Minimal Attachment and Late Closure, Lexical Preference, Parallel vs. Serial Processing and Competition and Reanalysis: Reparsing and Repair.

### 2.1.1 Right Association

The principle of Right Association states that elements are attached to the lowest non-terminal node currently present in the structure built during the parsing of the sentence (Kimball 1973). To illustrate the principle, let us analyze the following sentence (Gibson 1991):

(1) Bill thought John died yesterday.

The adverb *yesterday* can attach itself as a modifier to both verbs in the sentence, *thought* and *died*, producing two different meanings. Gibson (1991) states that the strongly preferred reading of this sentence is to attach *yesterday* to the verb *died*. The principle of Right Association can be seen in this preference as *yesterday* will be attached to the embedded verb *died* instead of *thought*, since it is the lowest node that the adverb can be attached to.

This section very briefly outlines this principle, as Kimball's (1973) principles of New Nodes and Right Association are predecessors to Frazier's principles of Minimal Attachment and Late Closure described in the next section, so this phenomenon is analyzed further as the Late Closure principle.

### 2.1.2 Minimal Attachment and Late Closure

Minimal Attachment and Late Closure principles were introduced by Frazier (1979) as explanations for the preferred readings of various sentences. The garden-path phenomenon arises precisely because these principles are followed by the reader while processing sentences where their grammaticality dictates the opposite of what the principles predict. Both principles follow the serial processing model (see 2.1.4 *Parallel vs. Serial Processing and Competition*), meaning that only one representation is present at a certain point during parsing. The principles come into play when there is a decision to be made which structure should be created in the current parse state.

The principle of Minimal Attachment states that the sentences and phrases with an ambiguous element are constructed in such a way that the result of the processing always has the smallest number of non-terminal nodes possible (Frazier 1979). Following this line of thinking, argument and specifier attachments are preferred over modifier attachments (Gibson 1991). On the other hand, the principle of Late Closure stipulates that the parser would attempt to attach the incoming element directly to the structure currently being processed, if possible (Frazier 1979). When the two principles assume different structures, the Minimal Attachment principle is given precedence (Gibson 1991).

The two principles can correctly predict the garden-path effect in many ambiguous sentences. The following analyses of ambiguous sentences will illustrate the inner workings of these principles. Let us start with the following sentence, perhaps one of the most commonly known garden-paths, first analyzed by Bever (1970):

(2) The horse raced past the barn fell.

The ambiguity of this sentence is present in the analysis of the element *raced*. The element can either be interpreted as a matrix verb or as a reduced relative clause (*The horse (that was) raced [...]*). The latter, of course, is the only one yielding a grammatical sentence as the NP *the horse raced past the barn* stands as the subject of the verb *fell*. The Minimal Attachment principle explains why the matrix verb reading is preferred, resulting in a garden-path effect. Consider the following representations of the ambiguous element *raced* (Gibson 1991):

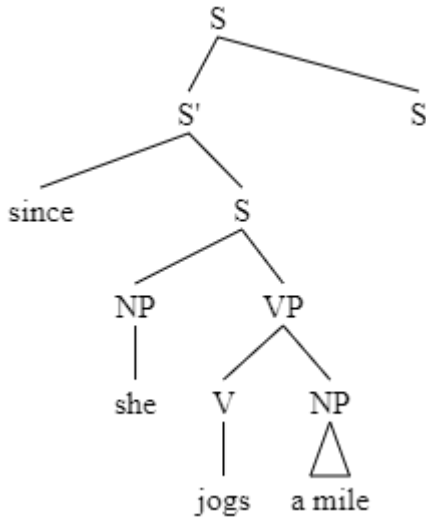
- a) Matrix verb: [S[NP the horse] [VP raced ]]
- b) Reduced relative clause: [S[NP the [N' [N' horse<sub>i</sub>] [S' [S[VP raced<sub>e</sub>]]]] ] ]

It is evident that the interpretation of *raced* as a matrix verb results in a significantly smaller number of non-terminal nodes, thus, readers will prefer the reading a) over the reading b). However, as the reading b) is the one which produces a grammatical sentence, readers encounter a garden-path effect.

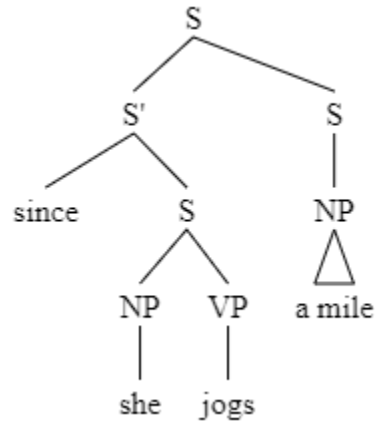
To illustrate the principle of Late Closure, let us analyze the preferred reading of the following example taken from Wanner (1980):

(3) Since she jogs a mile seems light work.

Like (1), this one is not a typical garden-path either, but structural ambiguity is evident. The element where syntactic ambiguity arises during the parse is the NP *a mile*. The NP can be interpreted as both the direct object of the verb *jogs*, or as a subject of the sentence *a mile seems light work*. Take a look at the tree representations of the two possible interpretations:



Tree Schematic 1: Since she [jogs a mile] as a direct object<sup>2</sup>



Tree Schematic 2: [Since she jogs] [a mile] as a subject of the second clause

Since the number of new non-terminal nodes after the analysis of *a mile* is the same, the Minimal Attachment principle cannot be applied, and thus cannot explain the occurrence of the garden-path effect. However, when we take into account the principle of Late Closure, the preferred attachment is indeed that of a direct object. Since the parser is processing the first clause and the first VP at the moment when the element *a mile* appears, it will attempt to attach it into the current structure to avoid creating a new one, thus resulting in an ungrammatical sentence.

It is, perhaps, interesting to notice that the lack of a comma between the two clauses also plays an important role in the creation of the ambiguity. If the sentence looked like this: *Since she jogs, a mile seems light work*; there would be no confusion as the comma would provide an

<sup>2</sup> <http://ironcreek.net/syntaxtree/>

intonation break which would not allow for the first interpretation and would clearly separate the two clauses, guiding the parser towards a correct interpretation. The same would be true in spoken language, where an intonational pause would be present, having the same disambiguating effect as a comma.

As previously mentioned, Minimal Attachment seems to be necessarily applied before Late Closure in order to correctly describe the preferred readings of sentences such as this one (Gibson 1991):

(4) John put the book on the table in the hall.

This sentence may not be a typical garden-path sentence which produces severe difficulties in processing, but it does contain an interesting structural ambiguity. In this sentence, the principles of Minimal Attachment and Late Closure contradict one another. The former dictates that the PP *on the table* is an argument of the verb, since this is the analysis with the fewest number of non-terminal nodes, and the latter requires it to be a modifier of the NP *the book*, since it is the structure that is currently being processed. The Minimal Attachment principle correctly predicts the reading of this sentence, and the Late Closure principle does not. While this ranking may produce the desired results, Gibson (1991) states that “such a ranking is not adequately independently motivated” and is thus “theoretically undesirable.” (15)

### 2.1.3 Lexical Preference

The previous principles primarily focused on the syntactic reasons behind misanalysis. In contrast, Ford, Bresnan and Kaplan (1982) introduced the notion of lexical preference to describe the garden-path phenomenon. They state that the garden-path phenomenon occurs when a word has two or more possible forms, and each one of these forms has its saliency, or strength<sup>3</sup>. The stronger form is, naturally, preferred. However, when the weaker form is needed in order for a sentence to be grammatical, a garden-path effect occurs. In other words, the garden-path effect

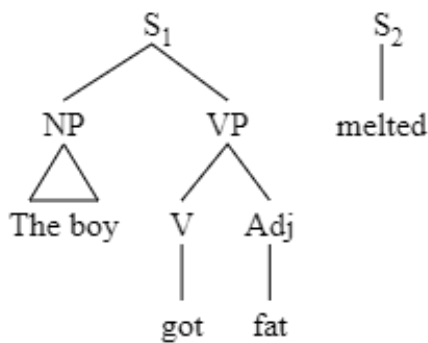
---

<sup>3</sup> In linguistic terms, saliency is a term describing the prominence of a linguistic item. Top-down saliency means that the item is cognitively preactivated based on some outside influence, for example if it had been mentioned earlier or if it is expected within a certain context. Bottom-up saliency, however, means that the item itself stands out because of its intrinsic characteristics. Strong forms are considered to be very salient, while weak forms are not.

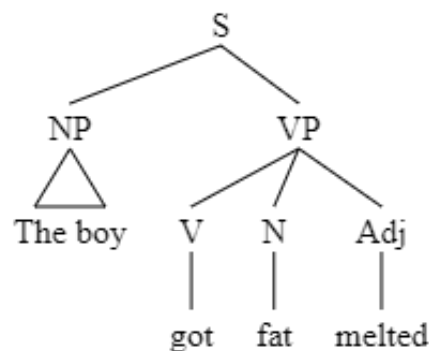
stems from lexical ambiguity and the need for a morphosyntactic or lexical reanalysis of an ambiguous item. The following sentence, taken from Pritchett (1988), illustrates this:

(5) The boy got fat melted.

The lexically ambiguous element that leads the reader down the garden-path is the word *fat*. According to Pritchett, the putative strong form of this word is as an adjective, rather than a noun<sup>4</sup>. To see the breakdown resulting from this analysis, let us look at the tree structure of this sentence with the strong form and the weak form:



Tree Schematic 3: [*The boy got*] *Fat* as an adjective



Tree Schematic 4: [*The boy got*] *Fat* as a noun

The garden-path effect is caused by the need to recategorize *fat* from an initial analysis as an adjective [*the boy became fat*] into a noun [*the boy caused the fat to be melted*], as the former is not compatible with the rest of the sentence.

Lexical Preference is not only concerned with words that can have different lexical categories, but also with verbs, which can have different subcategorization frames. Subcategorization frames dictate the possible environments in which a verb can occur. In other words, they represent certain grammatical constraints. As with different lexical frames, the theory

<sup>4</sup> More about why a certain word form is considered putative can be found in section 5 of this thesis.

is that different subcategorization frames have different saliency, and therefore, certain frames are preferred over others. Let us turn to the already analyzed sentence (2):

(2) The horse raced past the barn fell.

In the previous section, the garden-path effect in this sentence was described by the Minimal Attachment principle. The same sentence can be analyzed through the Lexical Preference principle as well in order to account for the garden-path effect. As noted above, the problematic element here is the word *raced*. As described by Ford, Bresnan and Kaplan (1982), the preferred reading of this word is as the simple past tense of the verb *to race*. However, when the parser reaches the verb *fell*, it becomes evident that the rest of the sentence does not comply with the subcategorization rules of the simple past tense structure, and a garden-path occurs. A reanalysis of *raced* is required, and a less salient structure of a passive participle is needed to resolve the problem.

However, Gibson (1991) outlines certain issues with this approach to garden-paths. He states that this model predicts the garden-path phenomenon when there is none and fails to account for the effect in sentences where it is clearly present. Let us again turn to an already presented example:

(6) Since she jogs a mile seems light work.

In this sentence, there is no morphosyntactic or lexical reanalysis, as no words have multiple lexical and/or subcategorization frames which could result in a garden-path. As the Lexical Preference model requires a morphosyntactic or lexical reanalysis to predict a garden-path, it fails to account for the effect in sentences such as these. However, despite these shortcomings, Gibson (1991) states that “nevertheless, the FB&K model is important since it was one of the first to point out some of the effects on processing of multiple lexical entries for the same word.” (38)

#### 2.1.4 Parallel vs. Serial Processing and Competition

Previous sections dealt with some models which tried to predict the garden-path effect, each in their own way. The assumptions covered in this section, however, can be applied alongside the described models to complete the picture about their *modus operandi*. The question is whether our parser deals with the ambiguities serially, or does it hold multiple possible analyses, which simultaneously compete for selection in a sort of a buffer, and then decide which one to keep. The serial models of sentence processing, of course, say no to this question. The opposing view, naturally, says yes.

Vosse and Kampen (2009) state that although these two strategies are different, they have much in common. According to them, both models presuppose that the final syntactic analysis must be compatible with the conceptual content. That is to say that a grammatical sentence must both be syntactically correct and semantically/pragmatically plausible. Also, both approaches assume incremental processing, word by word, as they are being input into the parser, without reading ahead or waiting for the rest of the sentence.

As an example of a serial model, we can look at Frazier's garden-path model (Frazier 1978; Frazier and Rayner 1982; Kimball 1973). It says that only a single analysis is being made based on various grammatical rules and structures by adding new words onto the structure already (being) processed. Vosse and Kampen (2009) describe these types of models as being syntax-first. This means that when an ambiguous input is encountered, one syntactic structure is constructed during the first processing stage by taking into account only the morphological and syntactic properties of the word. The models mentioned in the sections above describe the details of how these structures are built. Only after this initial processing does the parser submit the structure to, what the authors call, a conceptual processor, which then tries to assign certain meaning to the structure or a sentence. If the resulting structure is anomalous, the parser is said to have been led down the garden-path, and the ambiguous structure is revised. One of the mechanisms of reanalysis is covered in the next section.

Clifton and Staub (2008) describe numerous parallel models, some advocating direct inhibition, others indirect competition. Vosse and Kampen (2009) call these models constraint-based. For the purposes of this thesis, general conclusions regarding parallelism are selected to



illustrate the general principle. Clifton and Staub (2008) state that “the models basically claim that a single analysis must become dominant (perhaps to a very small degree) before a reader or listener can move on to begin processing the next word.” This means that if there are two or more possible analyses active in the parsing process, then competition ensues to determine which analysis will be dominant, however small the margin. Vosse and Kampen (2009) describe this process as an evaluation of “the extent to which [the analysis] satisfies relevant constraints of any type – pragmatic, discourse, semantic, morphological, syntactic, phonological etc.” (2) and the candidate which conforms best is the winner.

Competition in terms of lexical ambiguity, according to Clifton and Staub (2008), is quite straightforward. If a word’s two meanings are relatively equal in frequency and they appear in a context where both interpretations are plausible (at least at first, of course), then these meanings compete, and processing is slowed. If, however, one meaning is much more frequent and more common, and the context does not serve as much of a disambiguation tool, then competition has little effect on comprehension, as the parser makes an easy decision about the preferred analysis of the word. On the other hand, the authors question the competition model when it comes to syntactic ambiguity, and whether parallelism automatically entails competition.

To conclude, the question of parallel and serial processing, and whether there is any competition present when assuming the parallelism model is still an open one. Numerous authors have built upon initial theories of these strategies, providing new insights and relevant criticism, but an all-encompassing solution which would account for all the relevant empirical data regarding reading times, eye tracking and other experiments, to our knowledge, has not yet been found.

### **2.1.5 Reanalysis: Reparsing and Repair**

In the previous sections we described the phenomena which result in a garden-path effect. In order for a sentence to be properly understood, all of the mentioned phenomena presuppose some kind of reanalysis after the parser realizes that the structure which has been created is not grammatical and/or meaningful. This section focuses on this necessary process of reanalysis that comes after this realization. Naturally, there are several theories which attempt to describe the way in which our parser tries to repair the inconsistent structures and lead us back on the right track.

Lewis (1998) names four of them: strategies of backtracking, selection from parallel alternatives, refining commitments, and repair. The theories covered here are Reparsing and Limited Repair.

Reparsing is based on reanalysis by backtracking. Frazier and Rayner (1982) tested the reanalysis processes in readers by tracking their eye movements. They hypothesized three different methods of reanalysis: forward, backward and selective. Forward reanalysis entailed returning to the beginning of the sentence and parsing it again, looking for the place where a different parsing choice could be made. Backward reanalysis meant that the readers reread backward through the sentence to see if there are alternative parsing paths. In selective reanalysis, the reader tries to pinpoint and select the specific place where the analysis took a wrong turn and redo it. Frazier and Rayner concluded that selective reanalysis was the most plausible, even though readers also sometimes reread entire sentences from the beginning, giving some merit to the forward reanalysis strategy. Backward reanalysis was not supported by the data at all.

Winograd (1983) classified all of these hypotheses as a form of backtracking, as all three assumed returning to previous points during the parse and making a different choice regarding the ambiguous elements. Frazier and Rayner (1982) also concluded that eye movement is not necessarily present when backtracking and reanalysis occurs. Lewis (1998) states that “in fact, all backtracking reanalysis strategies can be realized covertly, without explicit eye movements and reparsing.” However, there is a difference. He says that overt backtracking deals with the actual linguistic input, while covert backtracking reparses the memory of that linguistic input. He goes on to emphasize that this distinction is not relevant in the field of Natural Language Processing, Computational Linguistics or AI, but it is an important difference for psycholinguistics because of memory issues.

As already mentioned, memory plays an important role in the theories of backtracking. For example, the Forward Reanalysis strategy does not require any memory input, as the parser simply resets and starts from the beginning. As Lewis (1998) points out, it only needs to remember not to choose the same interpretation that led to the garden-path effect in the first place. Other strategies, however, seem to require a lot of memory load, as the parser needs to retain the information of many prior states and the choices they provide during the process of reanalysis. Lewis (1998) criticizes some features of backtracking as a theoretical solution to reanalysis which, he believes,

are properly addressed in the Theory of Limited Repair. However, he states that backtracking, along with other mentioned theories, cannot be completely ruled out as a plausible solution to the problem.

Limited Repair parsers, according to Lewis (1998) have one major advantage over other theories, and this is reduced memory load. This type of parser needs only to maintain one parsing state at a certain point in time. Lewis (1998) enumerates several properties of the Limited Repair parser which make it highly suited for computational and psycholinguistic research:

- a) avoiding additional memory requirements, as opposed to backtracking
- b) avoiding the need for copy mechanisms
- c) the ability to incrementally produce complete syntactic structures
- d) the ability to distinguish between difficult-to-repair and easy-to-repair structures, thus differentiating between severe garden-path effects and those which one can easily recover from

However, there are, of course, downsides. To avoid backtracking, this parser needs new operators, such as DETACH and MOVE, which would undo and/or modify the problematic structures. Additionally, the operators would then have to be combined with certain constraints, making this parser relatively more complicated than those proposed by other theories of reanalysis.

## 2.2 Computational approaches

While the previous sections covered the psycholinguistic research into structural ambiguity and garden-paths, this section focuses on prominent computational theories and approaches to the same problems: The Sausage Machine, Race-based Parsing, Augmented Transition Networks and Machine Learning and Frequency Based POS-tagging.

### 2.2.1 The Sausage Machine

The Sausage Machine is a two-stage parsing model proposed by Frazier and Fodor (1978). The first stage uses a window<sup>5</sup> to analyze the input string and assigns structures to it according to grammar rules. The authors claim that the size (or scope) of this window is between five and seven words, meaning that the parser can only see and process a few words at a time. However, since they do not give any specific definition or reason for these numbers, for the purposes of our analysis, following Wanner (1980), we will assume that the specific number is six. The second stage of the model receives the chunks from the first stage and puts them together, again, following the rules of grammar. The parser also operates under the assumptions of Minimal Attachment and Right Association/Late Closure. The model is based on a parallel model of sentence processing (see 2.1.4). Frazier and Fodor suggest that the way the incoming lexical item is attached onto the existing structure depends on a race, namely the various ways in which an item can be attached are explored in parallel, like participants in a race, and the first successful option is the winner, while others are discarded.

The purpose of the Sausage Machine was to explain the processing difficulties not only in typical garden-paths, but in other sentences which are hard to process as well, for example, sentences with center-embedded relative clauses. To demonstrate the logic behind the Sausage Machine, let us look at one such sentence:

(7) The woman the man the girl loved met died.

Frazier and Fodor claim that the first stage of the parser analyzes the first six words of this sentence as a single conjoined noun phrase which is missing a conjunction [the woman and the man and the

---

<sup>5</sup> The term *window* can be described as a sort of a limited working memory of the parser.

girl]. The following words are analyzed in a similar manner, as a conjoined verb phrase [loved and met and died]. When trying to join the two structures together, the second stage parser fails since the resulting structure is an unacceptable representation. Upon reanalysis, Frazier and Fodor claim, the parser creates three separate NPs and three separate VPs in the first stage, and joining these NPs to their respective VPs appears to be too big of a task for the second stage, implying that this is the source of the difficulty in processing.

However, Wanner (1980) criticized the Sausage Machine model basing his argumentation on the fact that there are doubly center-embedded sentences which consist of only six words, enough for the first-stage parser to analyze by itself, thus avoiding the difficulties arising in the second stage of the parsing process:

(8) Women men girls love meet die.

Based on Frazier and Fodor's explanation, this sentence should be easy to process, as the first-stage processor should be able to see all six words right away and assign them proper structures. However, it is evident that this sentence is just as difficult to process as (7).

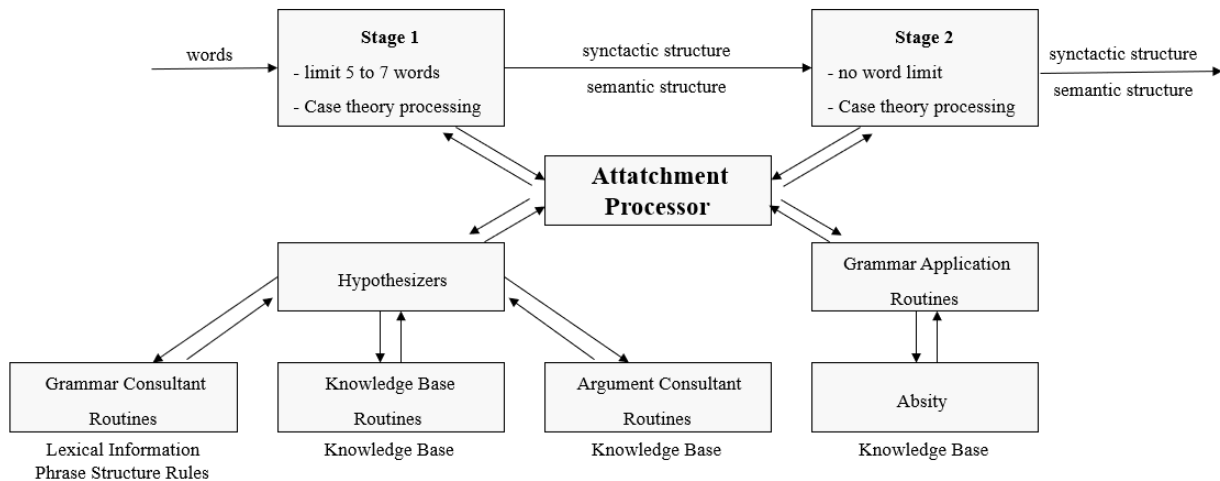
### **2.2.2 Race-based Parsing**

Using the Sausage Machine's parallel processing model as a steppingstone for their research, McRoy and Hirst (1990) developed Race-based Parsing, a computational model for sentence processing which is able to account for a far greater number of phenomena than the Sausage Machine. According to the authors, the parser "integrates some important psychological claims about the human sentence-parsing mechanism; namely, that processing is influenced by limitations on working memory and by various syntactic preferences." (313)

Because of the issue of memory limitations, the Race-based Parser calculates the time cost for every possible attachment, and only the fastest is kept for further processing, without maintaining other options. The cost is calculated based on several criteria, not only the number of grammar rules applied, but also factors such as priming, Minimal Attachment and Right Association, Lexical Frame Preferences, Semantic Preferences, and others. McRoy and Hirst (1990) conclude that these principles arise as a direct result of memory limitations since some

preferences simply make parsing easier or more efficient. Also, this means that sentences are broken down into smaller segments which are more manageable.

To understand this parser better, the authors created a simple system overview describing the stages in the parsing process within their model, all revolving around the main component called *Attachment Processor*:



Schematic 5: Race-based Parsing System Overview - adapted from McRoy & Hirst (1990)

The attachment processor is called upon in both stages to combine structures. It takes into consideration all possible attachments in a simulated parallelism and chooses the one with the lowest cost. The hypothesizers are independent processing modules which identify all possible attachments and calculate their time costs. They hold information about structural, thematic, and other expectations. The process goes like this: as a word or a phrase enters the parser, its lexical entry is retrieved. Then the attachment processor receives a suggestion on a suitable attachment point for this new item based on the input from hypothesizers – syntactic, semantic, pragmatic, and lexical procedures which suggest alternative attachment options and their time costs. The processor races the inputs to identify and execute the optimal attachment. Case theory is applied. If the package has reached a good breaking point and is nearing its capacity, it is closed and passed to the second stage. The second stage again calls the attachment processor to identify and evaluate the best attachment option for the next package into the current structure. Case theory is applied

again. Upon the completion of Stage 1 and Stage loops, tree formation routines carry out the final attachment with the lowest cost.

Kempen (1994) praises the model for its ability to account for psycholinguistic phenomena connected to syntactic recency and lexical frame preferences, the interaction between semantics and syntax, and effects such as priming. However, McRoy and Hirst (1990) themselves suggest that there are some problems with their model, mainly due to the lack of empirical data on timing of the hypothesizers, making their timing predictions *ad hoc*. They also believe that additional hypothesizers are needed to analyze more complex structures. Regardless of its shortcomings, the Race-based Parser represents a model which has the ability to combine relevant preferences and constraints within a single system.

### 2.2.3 Augmented Transition Networks

Augmented Transition Network, or ATN for short, is a structure within Graph Theory used to parse complex sentences in natural language. They are based on the general idea of finite state machines. According to Woods (1970), more efficient parsing is achieved when recursion is added to a finite state model, thus eliminating the need to build a finite state automaton (FSA)<sup>6</sup> for every particular sentence.

The parsing in an ATN is viewed as a set of transitions between states. Unlike the Sausage Machine, the ATN takes as input one word at a time. If the parser can go from the initial state to the final state via any number of states and transitions, the parsing is successful, and the input string is accepted as a viable natural language sentence. The final result represents a grammar for that sentence<sup>7</sup>.

The ATN also contains subnetworks for each of the major phrasal categories, S, NP, VP, RRC<sup>8</sup> and PP. These subnetworks are called upon when the parser has to analyze the phrase in

---

<sup>6</sup> FSA = Finite State Automaton – visual representations of a regular language.

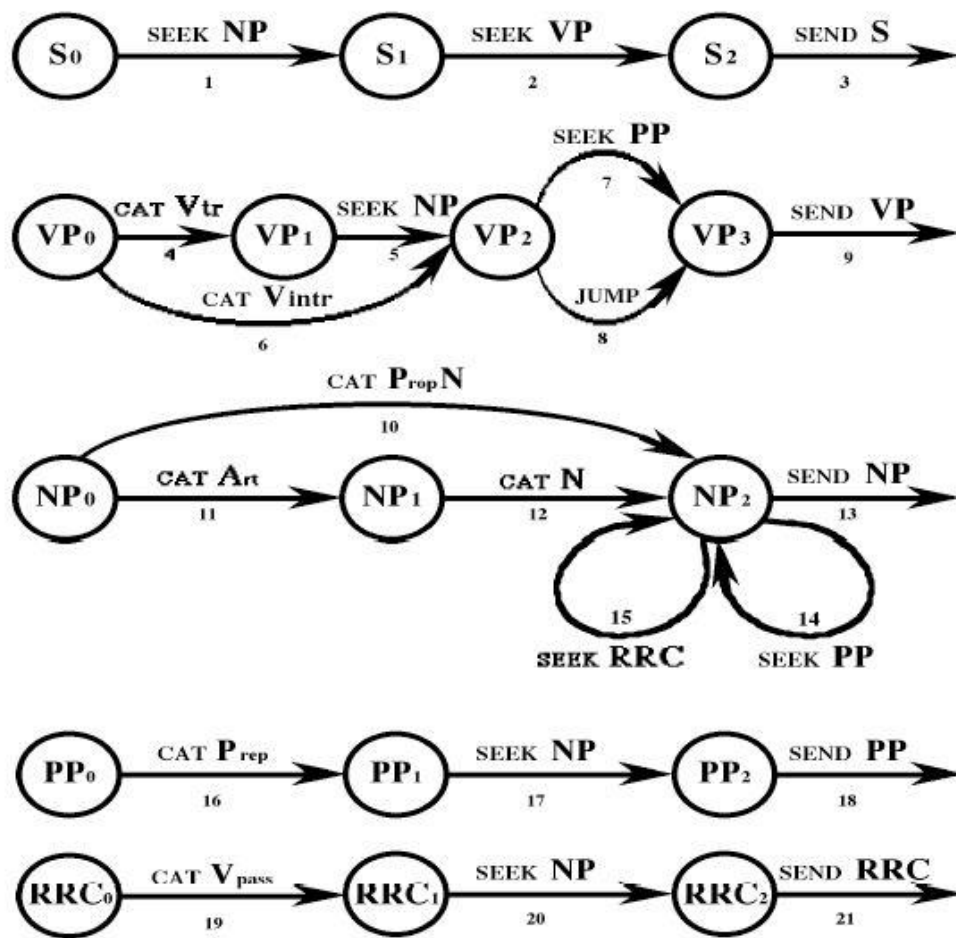
<sup>7</sup> By grammar is meant a set of permissible derivation rules.

<sup>8</sup> RRC refers to Reduced Relative Clauses which sometimes create ambiguities by being interpreted as main clauses.

question. The operations that take place when transitioning from one state to another are called arcs. Each arc has its name and its function:

- a) CAT arc - determines the grammatical category of the next input, such as Noun, Article, Transitive verb, etc.
- b) SEEK arc – calls upon the aforementioned subnetworks to analyze whole phrases
- c) SEND arc – if SEEK is successful, SEND arc terminates the current subnetwork and transfers control back to the previous subnetwork
- d) JUMP arc – indicates optionality of a structure

To clarify the structure of an ATN further, we provide the following graphs representing basic Augmented Transition Networks for analyzing sentences:



Schematic 6: ATN Overview – taken from Kempen (1994)



The first network is the main network which represents the derivation  $S \rightarrow NP VP$ . The SEEK operations call upon subnetworks needed to parse the possible phrases in the sentence. If the SEEK NP transition successfully analyzes an NP, the grammar transitions to the next state, where it then looks for a successfully parsed VP. Finally, it reaches the final state, and the sentence is accepted.  $VP_0$ ,  $NP_0$ ,  $PP_0$ , and  $RRC_0$  subnetworks shown here are representations of what is considered an acceptable structure of their respective phrases. For example, an acceptable NP which would be sent back to the S network can be described as this:

$NP \rightarrow \text{Art } NP_1 \mid \text{PropN} \mid \text{PropN RRC} \mid \text{PropN PP}$

$NP_1 \rightarrow N \mid N \text{ RRC} \mid N \text{ PP}$

Naturally, subnetworks can call upon other subnetworks. We can see this in the subnetwork designed for recognizing prepositional phrases, which shows that a PP consists of a preposition and an NP, and the NP is analyzed by its own subnetwork. If all transitions are successfully completed and the final state  $PP_2$  is reached, the prepositional phrase is sent back to wherever it was needed.

If, for any reason, the parser gets stuck somewhere and cannot reach the final state, the sentence is rejected. This is what is expected to happen when parsing garden-path sentences, and when it happens, intense backtracking needs to occur to see where the parser failed, and to correct the mistakes. A more detailed analysis of how ATNs deal with garden-path sentences and how they operate within some of the psycholinguistic frameworks described in section 2.1 is discussed in section 4.4 on Augmented Transition Networks and Minimal Attachment/ Right Association.

#### **2.2.4 Machine Learning and Frequency Based POS-tagging**

The approaches described in previous sections share a common denominator: they can all be described as rule-based. This section focuses on the other side of the coin in computational theories – statistical approaches. As J. Carbonell puts it in the foreword of Franz’s book *Automatic Ambiguity Resolution in Natural Language Processing*, “the beauty of statistical techniques for NLP is that in principle they only require training data, not manual reprogramming, to solve new or extended versions of the same problem.” (1996, V) Franz enumerates several different statistical

and corpus-based methods for resolving different types of ambiguity, however, for the purposes of this thesis, special focus is put on Machine Learning (ML) and frequency-based part-of-speech (POS) tagging, as an example of dealing with lexical-syntactic ambiguity.

Machine Learning is closely connected to the field of Artificial Intelligence, as its aim is to create algorithms through which computers can evolve certain patterns of behavior based on a large amount of empirical data. In terms of linguistics and NLP, this usually means training the computer's behavior and its pattern of decision making on a large corpus which has been annotated with a certain purpose in mind. This kind of approach is called *Supervised Machine Learning*. After the training phase, no further human input is required, as the machine is capable of generalizing based on what it has learned and can deal with novel data presented to it. Thus, this limits the need for human intuition which is a key part of rule-based systems that require manual algorithm revision. Of course, ML systems can receive new training data to improve or add new patterns of decision making. Some of the most notable supervised NLP Machine Learning systems are Bayesian Networks and Neural Networks, also known as Deep Learning. Unlike Supervised ML, *Unsupervised Machine Learning* is trained without annotated or pre-tagged data. This approach is generally used when there is need for just "basic" understanding of natural language, as it is only capable of extracting frequent patterns or clusters of data. Machine Learning can be applied to various NLP tasks, such as tokenization, Named Entity Recognition and Classification (NERC), POS-tagging and others.

One of the most notable POS-tagging systems which can be labeled as statistical as opposed to rule-based is *CLAWS*. *CLAWS*, or *Constituent-Likelihood Automatic Word-Tagging System*, was developed in the 1980s at Lancaster University's Centre for Computer Corpus Research on Language (UCREL). *CLAWS* was originally created to tag the Lancaster-Oslo-Bergen (LOB) Corpus, but the system has grown significantly since then. Garside (1987) states that around 35% of processed words in the LOB corpus are ambiguous, with two or more tags associated with it. In these situations, as a word cannot simultaneously be, for example, both a verb and a noun, it is necessary to somehow choose the correct POS tag. *CLAWS* has a tag-disambiguation program called CHAINPROBS, which determines which of the possible POS tags fits a word best. To do this, CHAINPROBS takes the data from the already tagged Brown corpus to determine the probabilities of a tag X occurring after a word tagged with Y. The multiple POS tags of an

ambiguous word are associated with figures representing the likelihood of their occurrence, and then reordered so that the most likely tag appears first. If the difference between these figures is very high, some possibilities are discarded right away. If not, the tags are left for manual post-editing phase. However, Garside (1987) states that in most cases, the first tag, one with the highest likelihood figure, proves to be the correct one.

As stated earlier, the system is continuously being upgraded. Currently, *CLAWS*'s accuracy is estimated to be as high as 96-97% and *CLAWS4* tagset was used to tag the British National Corpus. Later versions, from *CLAWS5* to *CLAWS7* entail changes to the number of tagsets and names of tags, and its latest version, *CLAWS8*, improved upon the distinctions between determiners and pronouns, and further improved the tagging of auxiliary verbs.

### 3 Methodology

As stated in the Introduction, the primary goal here is to compare psycholinguistic and computational approaches to structural ambiguity, with special focus on garden-path sentences. The analysis will explicitly and directly connect and compare the following concepts and approaches:

- a) Race-based Parsing and Minimal Attachment/Right Association
- b) Augmented Transition Networks and Minimal Attachment/Right Association
- c) Augmented Transition Networks and Lexical Preference
- d) Augmented Transition Networks and Machine Learning

In 2 *Theoretical Framework* several different concepts were elaborated upon, however, not all of these will be directly used in the comparison, and it is important to note why. Also, some concepts are so tightly interwoven that one automatically implies the other.

To explain these choices, let us take, for example, the Sausage Machine, which is missing from the list. The Sausage Machine as a computational approach, which is said to be able to account for concepts of Minimal Attachment and Right Association, served as a template for an improved version of its parser – Race-based Parsing. Hence, Race-based parsing is a preferred choice in this case since the analysis will also include all the basic notions of the Sausage Machine by extension. Furthermore, since Minimal Attachment and Late Closure evolved from the notion of Right Association, I believed it to be unnecessary to explicitly state and independently analyze all these principles, as they generally all operate together. Therefore, during the analysis, when Minimal Attachment and Right Association are mentioned, it implies the inclusion of the Late Closure principle as well. Moreover, reanalysis by backtracking is the default method used in computational approaches to recover from misanalysis, so there is no need for a separate section describing its interactions with some other approach, as it is present and elaborated to a certain extent in all of them. The same is true for parallel and serial modes of processing, since all computational approaches, with the exception of Race-based parsing, entail incremental serial processing. Additionally, frequency-based POS-tagging is not described as a separate unit in the comparison, as it is interlaced with the principles of Machine Learning and tightly connected to

the notion of Lexical Preference, and is therefore represented within the analysis of these two concepts.

The analysis is built upon four different sentences, one for each of the enumerated points of comparisons. Three of these are typical garden-path sentences, and one is a sentence which represents a similar attachment problem as (4). The sentences are presented through tree schematics to help clearly illustrate the different analyses. If applicable, the sentences are also shown as a derivational process or a flow-chart showing the decisions behind the analysis. In order to keep the sentence representations clean and simple, Context-Free Grammar, or CFG, is used to illustrate the processing breakdown, the backtracking that follows, and as a clarification of the workings of ATN networks.

As the analysis moves onto statistical approaches, the method of Chi-square is introduced to determine the likelihood of co-occurrence between words. The calculation is shown based on the following formula, where  $O_i$  represents observed values, and  $E_i$  expected values:

$$\chi^2 = \sum_i^n \frac{(O_i - E_i)^2}{E_i}$$

The figures needed for the calculation of Chi-square are taken from the British National Corpus, accessed via the SketchEngine tool<sup>9</sup>. It is important to note, however, that the tags within any corpus can be faulty, and that calculations such as these highly depend on the accuracy of the input data. As we can never be sure that the data is 100% correct without manually checking the tags of every single concordance we get, the results must be taken with a grain of salt. In addition, raw data regarding the frequency of a POS-tag for a certain word is also used to determine its strong and weak forms.

Finally, one sentence is shown as an example of synthesis, analyzed from scratch, combining CFG representation and statistical approaches to illustrate both the postulates of Minimal Attachment/Late Closure and Lexical Preference.

---

<sup>9</sup> <https://www.sketchengine.eu/>

## 4 Research and analysis

### 4.1 Race-based Parsing and Minimal Attachment/Right Association

To illustrate the inner workings of Race-based Parsing and how it handles the principles of Minimal Attachment and Right Association, we can go through the original sentence which McRoy and Hirst (1990) used as an example to explain their model:

(9) Dan read the letter to Chrysanne.

This sentence is not a typical garden-path, but it does contain ambiguity which needs resolving. The first part of the sentence is easy to process. When the word *Dan* is processed, after retrieving it from the lexicon and after finding no ambiguity, it is immediately given a corresponding syntactic structure. Since there are no other structures to connect it with at this point, it is left waiting for further input. Next, the word *read* is processed. This word contains some ambiguity as it can have two theta-grids. The lexicon suggests the options of  $V_{tr}$  and  $V_{pass}$ , either as a main verb satisfying the Case requirements of the preceding NP, or as a start of a reduced relative clause (RRC), respectively. The hypothesizers then calculate the costs of each of these options. It assigns the time cost of 3 to the first option, and 6 to the second option<sup>10</sup>. Based on this, the attachment processor attaches *read* as the main verb. This choice leads the parser to anticipate an object NP. Processing further input, *the letter* satisfies this requirement.

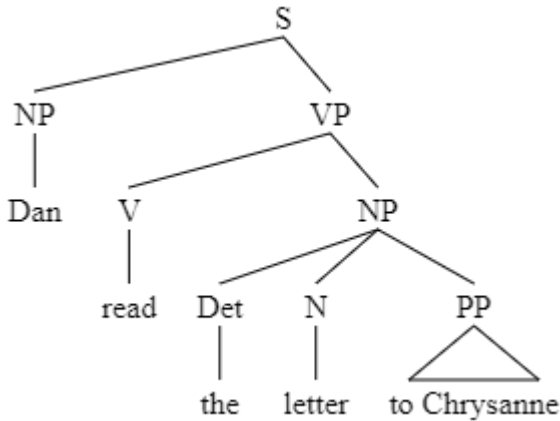
Next, however, the parser encounters a highly ambiguous *to*. Hypothesizers then report five possibilities:

- a) *to* as a start of a PP modifier to the noun *letter*, estimated cost: 4
- b) *to* as a start of a PP modifier to the verb *read*, estimated cost: 11
- c) *to* as start of a modifying IP clause to the verb, estimated cost: 11
- d) *to* as start of a PP denoting a BENEFICIARY to the verb *read*, estimated cost: 3
- e) *to* as start of a clause denoting a PURPOSE role, estimated cost: 5

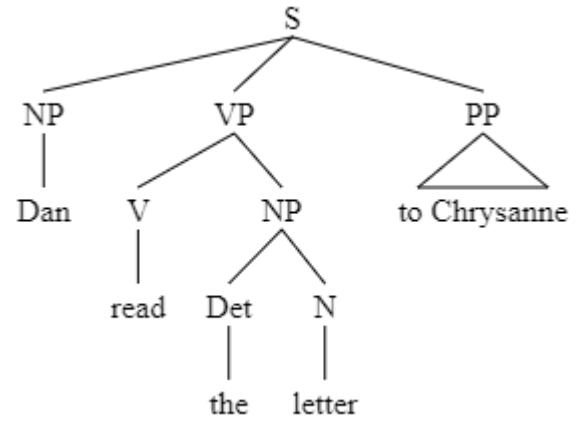
---

<sup>10</sup> It is important to again emphasize that the authors themselves admit that these estimated time costs are somewhat arbitrary.

The two most plausible and least costly options are shown below:



Tree Schematic 7: option a) *to* as a start of a PP modifier to the noun *letter*



Tree Schematic 8: option d) *to* as start of a PP denoting a BENEFICIARY to the verb *read*

Based on the estimated time costs, the option d) is chosen, and *to* is attached to the verb and is expecting another NP after it. The next word fulfills this expectation and *Chrysanne*, an unambiguous element, is attached to it. The result is in tune with the preferred interpretation of the sentence.

The way in which the Race-based parsing model represents the Minimal Attachment principle is described by McRoy and Hirst (1990):

“We use a more general statement of MA: The attachment that can be computed most quickly (i.e., with lowest time cost) is the one that will be favored. [...] the stronger an expectation for attaching a syntactic object, the lower the time cost of identifying the attachment, and the smaller the amount of additional structure necessary for attaching an object, the lower the time cost of executing the attachment.” (323)

They also assume that the principle of Right Association can be boiled down to associating higher time costs with more distant connections, too. However, Kempen (1994) points out that the choice of attachment in (9) was not dictated by the original principle of Minimal Attachment – even

though the parser does opt for the minimal amount of new nodes – but rather by expectations imposed by lexical frame information. Also, Right Association/Late Closure is not employed here, as the PP is not attached to the NP, which is the closest and currently processed structure, but rather to the VP.

To conclude, the Race-based Parsing system, in principle, incorporates the Minimal Attachment and Right Association principles by assigning lower time costs to preferred structures, thus giving them an advantage in the “race” over the options not preferred by the two principles. However, McRoy and Hirst (1990) fail to clearly show how the principles operate based on their analysis of (9), as the time costs assigned to different interpretations seem to be highly arbitrary, and Right Association/Late Closure does not seem to be followed in their example at all.

#### **4.2 Augmented Transition Networks and Minimal Attachment/Right Association**

In his paper, *The ATN and the Sausage Machine: Which One is Baloney?*, Wanner (1980) criticized the Sausage Machine model, claiming that it does not represent the principles of Minimal Attachment and Right Association in a structured way. He also states that unlike the Sausage Machine, ATNs can. In order to account for this, he described two simple rules in terms of the arcs operating within an ATN:

- a) To model Minimal Attachment: all CAT arcs are scheduled before any SEEK arcs
- b) To model Right Association: all SEND and JUMP arcs are scheduled after SEEK and CAT arcs

Although he shows that these two strategies for ATNs can account for principles of Minimal Attachment and Right Association, he is not able to explain the necessity of applying these principles, as opposed to others. However, Kempen (1994) challenges the claim that the proposed strategies deal with Minimal Attachment successfully. He illustrates his point by testing the ATN on the following garden-path sentence:

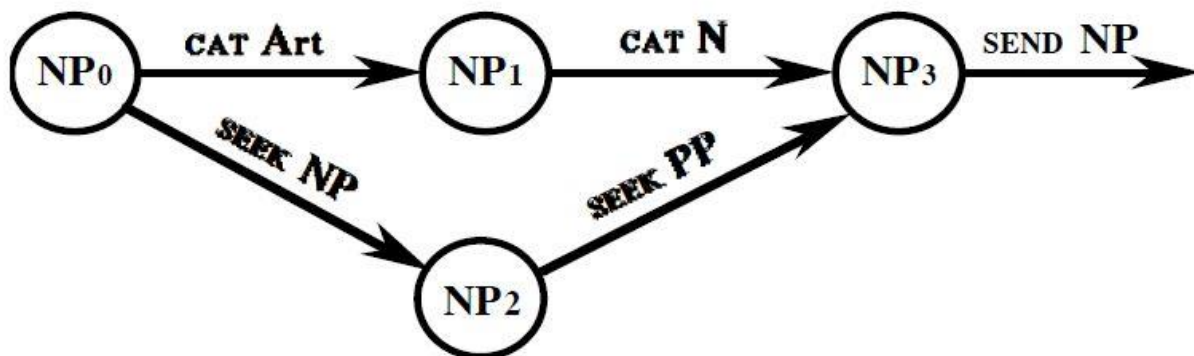
(10) The student read the letter to Chrysanne fainted.

This sentence is rather similar to (9), however, there are key differences in the final analysis. Unlike in (9), the correct analysis of the verb *read* in (10) is as a  $V_{\text{pass}}$ , a start of a reduced relative



clause. When we follow the ATNs in Schematic 6, we can see that the arcs have no problems until they hit the word *fainted*. Intense backtracking is needed to correct the analysis. Now, if we introduce the SEEK arcs before SEND arcs, as Wanner (1980) proposes, we eliminate the need for backtracking. However, in that case, a sentence without *fainted*, such as sentence (9), becomes a problem.

This is a general problem with Wanner's two strategies, however, Kempen (1994) specifically finds fault with the management of Minimal Attachment with these solutions. He states that by applying the solutions, Right Association is indeed shown, as *to Chrysanne* is attached to the NP, rather than the VP<sup>11</sup>. However, *fainted* remains a problem. He states that the problem is with the left-recursive rules, where the symbol on the left is the same as the first symbol on the right, such as the rewriting rule of  $NP \rightarrow NP + PP$ . Let us compare the sample grammar which has its NP<sub>0</sub> subnetwork<sup>12</sup> replaced by a different version:



Schematic 9: A new NP<sub>0</sub> replacing the default NP<sub>0</sub> – taken from Kempen (1994)

This new NP<sub>0</sub>, however, gets easily trapped in an infinite recursion. When a parser containing this modification encounters *fainted* in (10), the analysis fails, the parser backtracks and does not take the path of CAT arcs, but the only alternative, SEEK NP arcs. This again labels *the letter* as an NP, as it did the first time around as well. The parser is now caught between NP<sub>2</sub> states and NP<sub>3</sub> states since it cannot find a PP, given that *fainted* is not a preposition. It backtracks again, getting

<sup>11</sup> Interestingly, this is the opposite of the choice made by Race-based Parsing for (10).

<sup>12</sup> Shown in Schematic 6 alongside other subnetworks.

caught in the same trap. To clarify further, let us consider the sample grammar which Kempen (1994) associates with sentences (9) and (10):

1. $S \rightarrow NP VP$	6. $RRC \rightarrow V_{pass} NP$	11. $PP \rightarrow Prep NP$	16. $N \rightarrow letter$
2. $NP \rightarrow Art N$	7. $VP \rightarrow V_{intr}$	12. $V_{intr} \rightarrow fainted$	17. $N \rightarrow student$
3. $NP \rightarrow PropN$	8. $VP \rightarrow V_{intr} PP$	13. $V_{tr} \rightarrow read$	18. $PropN \rightarrow Chrysanne$
4. $NP \rightarrow NP PP$	9. $VP \rightarrow V_{tr} NP$	14. $V_{pass} \rightarrow read$	19. $Prep \rightarrow to$
5. $NP \rightarrow NP RRC$	10. $VP \rightarrow V_{tr} NP PP$	15. $Art \rightarrow the$	

Table 1: Sample Grammar – adapted from Kempen (1994)

The left-recursive rules such as 4 and 5 pose a problem, as described earlier. Hence, Kempen (1994) concludes that left recursion should be avoided in ATNs. However, the default  $NP_0$  shown in the Schematic 6 has its faults too. It does not contain left-recursive rules, but rather corresponds to non-left-recursive rules  $NP \rightarrow Art N PP^*$  and  $NP \rightarrow Art N RRC^*$ . If we were to switch the rules 4 and 5 in the Sample Grammar with their non-left-recursive versions, then, Kempen (1994) says, it is no longer predicted that the interpretation of *to Chrysanne* is as part of a  $VP^{13}$ , but rather as a modifier of  $NP$ , going against the Minimal Attachment principle. Based on this, he states that “[t]he conclusion must be that Wanner’s proposal cannot provide a viable account of Minimal Attachment preferences.” (13)

---

<sup>13</sup> To compare, this is exactly the analysis that Race-based Parsing deemed to be the least costly and in tune with the Minimal Attachment principle.

### 4.3 Augmented Transition Networks and Lexical Preference

To illustrate the importance of Lexical Preference and its impact on the processing breakdown of garden-path sentences, in this section we first analyze a common, unambiguous sentence using CFG, which is later modified in order to create a garden-path sentence with the same meaning, following Du & Yu (2012):

(11) The new singers record the song.

First, we define the sets of non-terminal (Vn) and terminal nodes (Vt) used in the analysis:

Vn = {Det, Adj, N, NP, V, VP, S}

Vt = {the, new, singers, record, song}

The set of production rules is as follows<sup>14</sup>:

1. S → NP VP	5. Det → {the}
2. NP → Det N	6. N → {singers, song}
3. NP → Det Adj N	7. Adj → {new}
4. VP → V NP	8. V → {record}

Table 2: Production rules for (11), adapted from Du & Yu (2012)

We can now show the complete parsing process for (11)<sup>15</sup>:

- a) Det new singers record the song (5)
- b) Det Adj singers record the song (7)
- c) Det Adj N record the song (6)
- d) NP record the song (3)
- e) NP V the song (8)

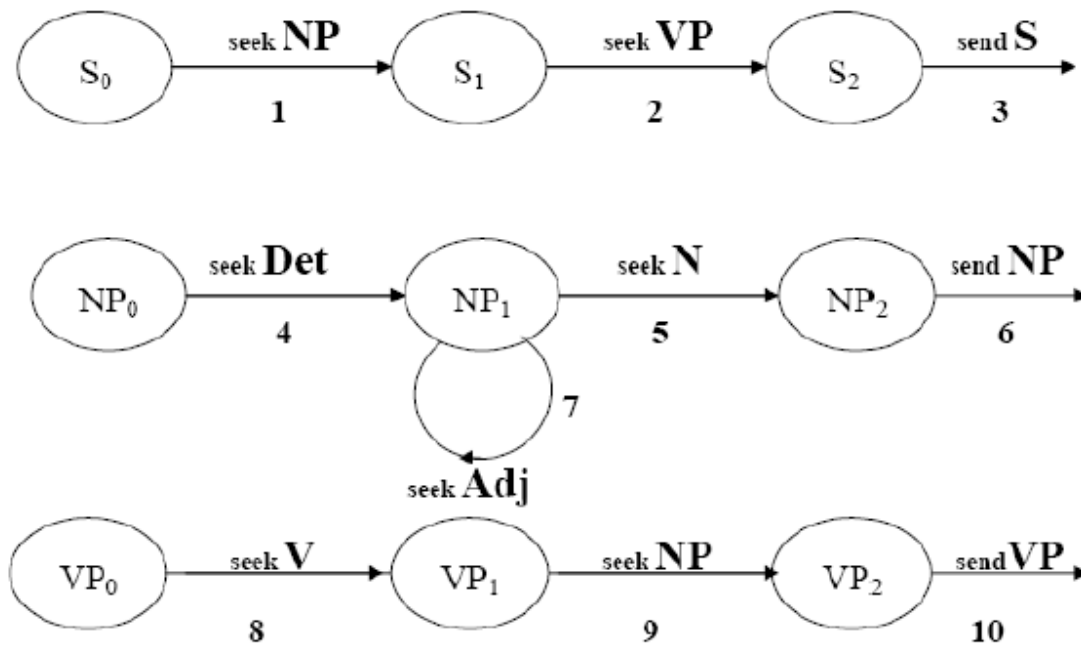
---

<sup>14</sup> Note the difference between the abbreviations for articles in Table 1 (Art) and Table 2 (Det). Also, the convention for denoting terminal symbols is different, where using curly brackets to differentiate terminals from non-terminals is now the norm, as shown in Table 2.

<sup>15</sup> The numbers in round brackets represent the rule from Table 2 used in each step.

- f) NP V Det song (5)
- g) NP V Det N (6)
- h) NP V NP (2)
- i) NP VP (4)
- j) S (1)
- k) SUCCESS

The parsing goes smoothly, and the sentence is accepted both grammatically and semantically without encountering any problems during the processing. There is no backtracking involved nor are there any ambiguities. Du & Yu (2012) also provide an ATN which would be sufficient to analyze (11):



Schematic 10: ATN for (11) – taken from Du & Yu (2012)

Now let us continue to the garden-path sentence. This next sentence has the word *singers* removed creating ambiguity with the word *record*:

- (12) The new record the song.

Again, we define the sets of non-terminal and terminal nodes for this sentence:

$V_n = \{\text{Det, Adj, N, NP, V, VP, S}\}$

$V_t = \{\text{the, new, record, song}\}$

New production rules are needed as well:

1. $S \rightarrow NP VP$	6. $\text{Det} \rightarrow \{\text{the}\}$
2. $NP \rightarrow \text{Det Adj}$	7. $N \rightarrow \{\text{record, song}\}$
3. $NP \rightarrow \text{Det Adj N}$	8. $V \rightarrow \{\text{record}\}$
4. $NP \rightarrow \text{Det N}$	9. $\text{Adj} \rightarrow \{\text{new}\}$
5. $VP \rightarrow V NP$	

Table 3: Production rules for (12), adapted from Du & Yu (2012)

Note how the word *record* is represented by two possible non-terminals<sup>16</sup> in Table 3, as N and V.

The analysis is the following:

- a) Det new record the song (6)
- b) Det Adj record the song (9)
- c) Det Adj N the song (7)
- d) NP the song (3)
- e) NP Det song (6)
- f) NP Det N (7)
- g) NP NP (4)
- h) FAIL

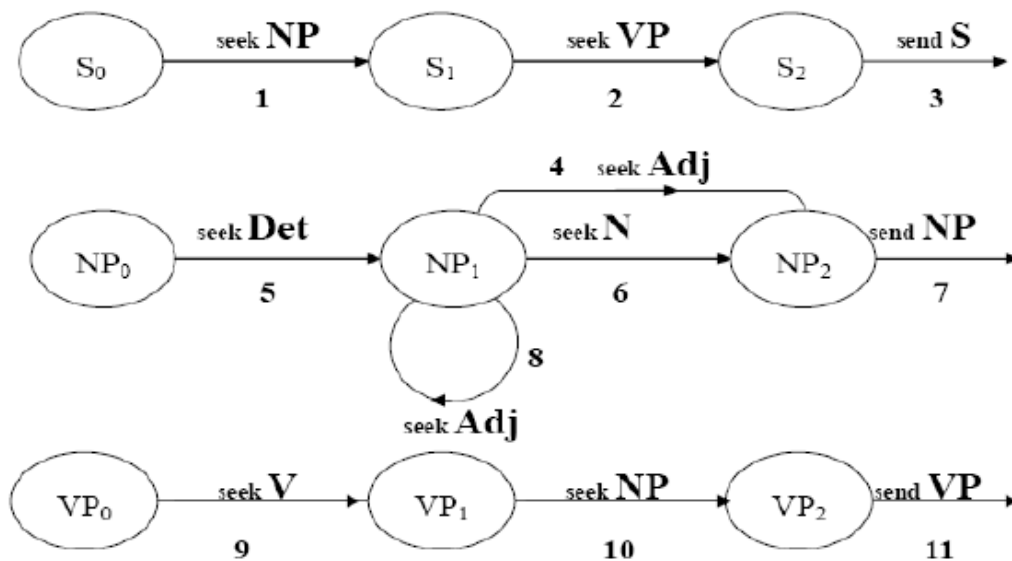
---

<sup>16</sup> Non-terminal symbols can be more simply called syntactic variables, too.

This processing path failed to create the final non-terminal node S, and the sentence is not accepted. Backtracking and reanalysis are needed to find and correct the breakdown in processing:

- i) Det Adj record the song (9)
- j) NP record the song (2)
- k) NP V the song (8)
- l) NP V Det song (6)
- m) NP V Det N (7)
- n) NP V NP (4)
- o) NP VP (5)
- p) S (1)
- q) SUCCESS

It may be noted how a different choice of a rewriting rule for *record* in j) results in a successfully parsed sentence, as opposed to b). Now, the choice between rules 7 and 8 can be made randomly, with a 50:50 chance that a sentence such as (12) can be parsed successfully. However, Du & Yu (2012) state that the NP *the new record* has a high probability of parsing and is therefore the preferred choice. Du & Yu (2012) also provide an ATN to show the parsing process:



Schematic 11: ATN for (12) – taken from Du & Yu (2012)

They emphasize that the NP subnet in this ATN is the reason behind the garden-path phenomenon in (12), as there are two options for analyzing an NP, either as NP → Det Adj, or NP → Det Adj N:

“Generally speaking, Adj is used to modify the Noun, the model of NP→Det Adj N is the prototype of parsing, and system interprets [12] by means of this programming rule rather than NP→Det Adj. After completing the NP subnet parsing of <the new record>, system returns to S network to seek VP. However, the left phrase <the song> has no VP factor according to the lexicon knowledge, and system stops, backtracks and transfers to another programming rule, i.e. NP→Det Adj. Cognitive breakdown happens.” (71)

From their analysis of several garden-path sentences and their non-ambiguous counterparts, including sentences (11) and (12), they conclude that the existence of homonyms creates ambiguities and results in the garden-path effect. However, this specific statement is problematic as words such as *record* are not homonyms, but rather homographs, since the diverging meanings of *record* as a verb and *record* as a noun are not accidental, but clearly cognitively connected. A better explanation would employ the notion of Lexical Preference, which could cover both homonymy and polysemy, in this case, and some other factors as well (see 2.1.3).

Furthermore, Du & Yu (2012) continue to analyze the reasons behind the garden-path phenomenon, emphasizing the role of syntactic structures. They show how The Stanford Parser, a parser trained to try to produce the most likely structures based on manually parsed sentences by using Probabilistic Context Free Grammar (PCFG) among other means, parses ambiguous sentences. They show that the parser also produces a failed NP NP structure when analyzing (12):

(ROOT  
(NP (I)  
  (NP (DT the) (JJ new) (NN record)) (II)  
  (NP (DT the) (NN song)) (II)  
(. .)))

Unlike the CFG/ATN rule-based approach, the Stanford parser does not employ backtracking and reanalysis, and simply gives the final result of the analysis in NPs. Du & Yu

(2012) conclude that ambiguous sentences can have multiple syntactic structures resulting from lexical ambiguity. They state that according to the principles of PCFG, the strongest probability parsing is the final parsing. We can back up this claim by accessing data on the frequency of the word *record* tagged as V and tagged as N in the British National Corpus:

<i>record</i> – verb (V)	9 511	84.66 per million
<i>record</i> – noun (N)	19 155	170.5 per million

Table 4: Different POS frequencies of *record*

It is more than clear, from both absolute and relative figures, that the more frequent form of *record* is that of a noun, which justifies the initial parsing choices, and implies a connection between putative/strong/preferred forms and their frequency in usage.

What can be concluded from the discussion above is that the notion of Lexical Preference is very neatly shown through ATNs and their descriptions using Context Free Grammar, although Lexical Preference as such is not explicitly mentioned in this way by Du & Yu in their paper.

#### 4.4 Augmented Transition Networks and Machine Learning

This section touches upon a similar interaction between rule-based and statistical approaches as in 4.3. In their short paper, Du, Yu & Li (2014) present an approach which attempts to integrate computational technologies, such as CFG and ATN to help machine learning systems in their natural language processing tasks dealing with garden-path sentences. They approach garden-path sentences with structural and statistical analyses, and algorithm processing. For the purposes of this thesis, only structural and statistical analyses are thoroughly analyzed, as they are sufficient to show how Machine Learning benefits from rule-based approaches, specifically Augmented Transition Networks and CFG, in this case. Du, Yu & Li (2014) use the following example to present their theory:

(13) The complex houses married and single students and their families.

Just as with (12), we start by showing the parsing and processing breakdown of this sentence using CFG:



$V_n = \{S, NP, VP, Det, AdjP, N, Conj, V, Adj\}$

$V_t = \{the, complex, houses, married, and, single, students, their, families\}$

1. $S \rightarrow NP VP$	6. $NP \rightarrow NP Conj NP$	11. $N \rightarrow \{complex, houses, students, families\}$
2. $NP \rightarrow Det Adj$	7. $NP \rightarrow Det N$	12. $V \rightarrow \{houses\}$
3. $NP \rightarrow Det NP$	8. $VP \rightarrow V$	13. $Adj \rightarrow \{complex, married, single\}$
4. $NP \rightarrow AdjP N$	9. $VP \rightarrow V NP$	14. $Conj \rightarrow \{and\}$
5. $AdjP \rightarrow Adj Conj Adj$	10. $Det \rightarrow \{the, their\}$	

Table 5: Production rules for (13), adapted from Du, Yu & Li (2014)

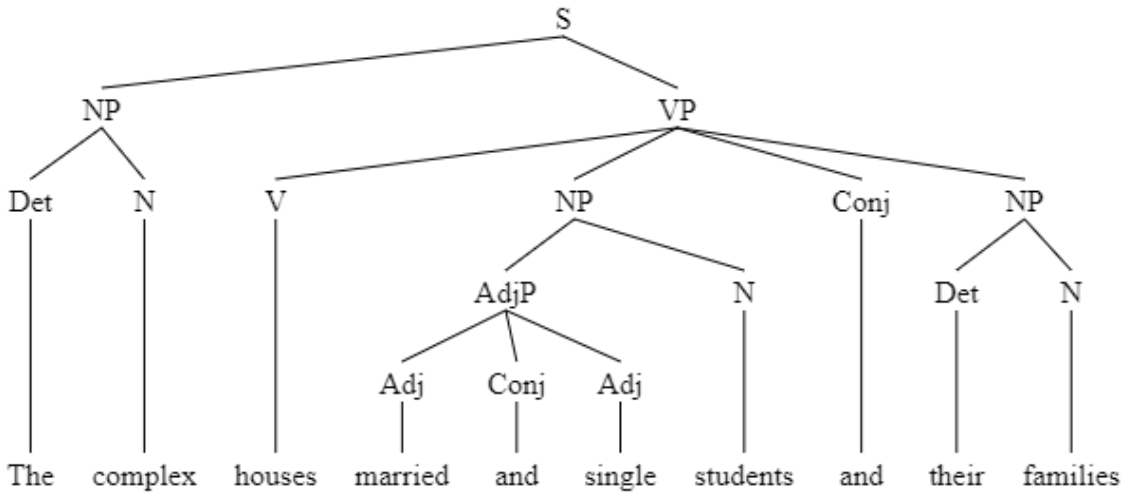
By applying these production rules (shown in brackets for each step), we get the following parsing process:

- a) Det complex houses married and single students and their families (10)
- b) Det Adj houses married and single students and their families (13)
- c) Det Adj N married and single students and their families (11)
- d) Det NP married and single students and their families (4)
- e) NP married and single students and their families (3)
- f) NP Adj and single students and their families (13)
- g) NP Adj Conj single students and their families (14)
- h) NP Adj Conj Adj students and their families (13)
- i) NP AdjP students and their families (5)
- j) NP AdjP N and their families (11)
- k) NP NP and their families (4)
- l) NP NP Conj their families (14)
- m) NP NP Conj Det families (10)
- n) NP NP Conj Det N (11)
- o) NP NP Conj NP (3)
- p) NP NP (6)
- q) FAIL

Since the NP NP construction cannot be the ultimate processing, based on the production rule 1, the parser experiences breakdown and is forced to backtrack to the point where it needs to employ a different rule to parse *houses* as a verb, instead of a noun in c). This step changes the parsing of subsequent items as well:

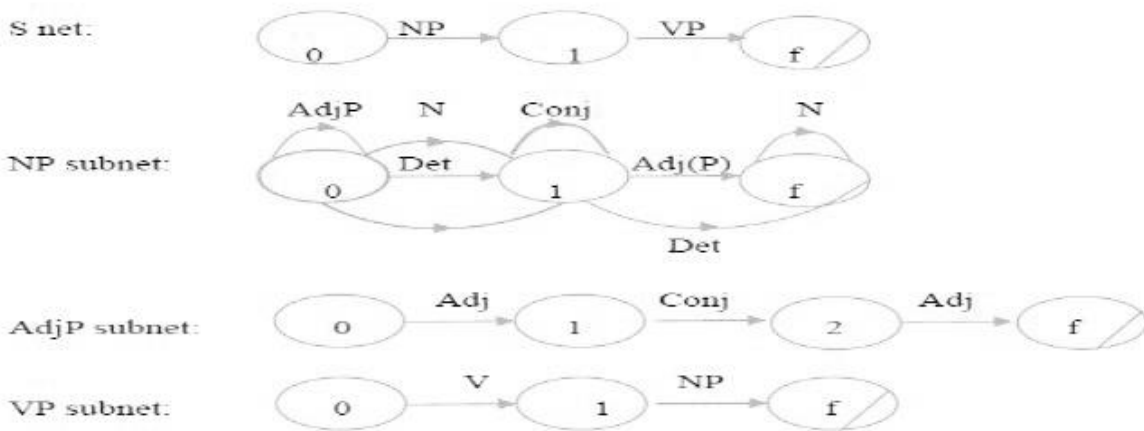
- a) Det Adj houses married and single students and their families (13)
- b) NP houses married and single students and their families (2)
- c) NP V married and single students and their families (12)
- d) NP V Adj and single students and their families (13)
- e) NP V Adj Conj single students and their families (14)
- f) NP V Adj Conj Adj students and their families (13)
- g) NP V AdjP students and their families (5)
- h) NP V AdjP N and their families (11)
- i) NP V NP and their families (4)
- j) NP V NP Conj their families (14)
- k) NP V NP Conj Det families (10)
- l) NP V NP Conj Det N (11)
- m) NP V NP Conj NP (7)
- n) NP V NP (6)
- o) NP VP (9)
- p) S (1)
- q) SUCCESS

The ultimate successful processing allows us to obtain a structurally sound tree diagram of the sentence:



Tree Schematic 12: *The complex houses married and single students and their families.*

We can also show the parsing process within an Augmented Transition Network designed to process this sentence:



Schematic 13: ATN for (13) – taken from Du, Yu & Li (2014)

Following this ATN, we can show just where the processing breaks down and induces backtracking:

- a) <S/0, The complex houses married and single students and their families. >
- b) <NP/0, The complex houses married and single students and their families, S/1 >

- c) <NP/1, complex houses married and single students and their families, S/1 >
- d) <NP/1, houses married and single students and their families, S/1 >
- e) <NP/f, married and single students and their families, S/1 >
- f) <VP/0, married and single students and their families, S/f >
- g) ?

The choice to process *houses* within an NP subnetwork led the parser down the garden-path, and backtracking is needed to reach a successful parsing of the full sentence, so instead of utilizing NP/1, *houses* is processed through VP/0, it successfully reaches VP/1, sending the correct interpretation back into the S net.

Based on this theoretical analysis, Du, Yu & Li (2014) create a statistical analysis using the Chi-square test to show the reasons behind the wrong choices that led to the processing breakdown of sentence (13). It is obvious that the problematic item is the word *houses*. The authors state that “the choice with much higher frequency is the preferred structure from the cognitive and statistical perspective.” (61) They search the BNC corpus and find that the word *houses* appears 215 times as a verb, and 4784 times as a noun. They present the observed frequency and the expected frequency, as well as the result of the Chi-square test calculation, and based on this they conclude that treating *houses* as a noun is preferred. Furthermore, they test not only *houses* as a single unit, but they calculate the likelihoods of structures for *complex houses* as [Adj + N] and [N + V].

Category	O	E	O-E	(O-E) <sup>2</sup>	(O-E) <sup>2</sup> /E
Houses, verb	215	2499.50	-2284.50	5218940.25	2087.99
Houses, noun	4784	2499.50	2284.50	5218940.25	2087.99
$\chi^2$					4175.99

Table 6: Chi-square test for *houses*, adapted from Du, Yu & Li (2014)

The numbers tell them that the [Adj + N] structure is the prototypical one. The fact that the parser automatically chooses prototypical structures instead of less frequent ones leads to incorrect parsing, and backtracking occurs.

After constructing a Machine Learning algorithm that employs not only the methods typically used for ML systems, but also includes the shown rule-based and statistical approaches, Du, Yu & Li (2014) state that garden-path sentences are an excellent way to test the accuracy of ML systems, as the structures present in them are very complex and require a ML system that is efficient enough to go through the peculiarities of backtracking to recover from a breakdown. To achieve this, the authors state that “a hybrid technique of computational linguistics, including theoretical and statistical analysis, is an effective approach to parse garden-path sentences for a ML system.” (62)

## 5 Synthesis

Following the main points made in 4.2, 4.3, and 4.4, we use (14) to show how employing CFG and statistical methods can illustrate the decision-making process behind parsing garden-path sentences, the breakdown that occurs and how the system recovers. While presenting the structural analysis of the sentence, the theoretical postulates of Minimal Attachment/Late Closure and/or Lexical Preference are emphasized as the main culprits behind the choices in the analysis:

(14) The old man the boat.

First, let us go through the CFG representation of the parsing process of (14). For a complete formal grammar to be built, we need to define a finite set of terminal and non-terminal symbols, production rules, and naturally, a distinguished sentence symbol  $S$ . Unlike in 4.3 and 4.4, we will use symbols  $N$  and  $\Sigma$  to denote non-terminal and terminal symbols, respectively, following Chomsky (1957)<sup>17</sup>:

$N = \{S, NP, VP, Det, Adj, N, V\}$

$\Sigma = \{\text{the, old, man, boat}\}$

Specific production rules are needed too:

1. $S \rightarrow NP VP$	6. $Det \rightarrow \{\text{the}\}$
2. $NP \rightarrow Det Adj$	7. $N \rightarrow \{\text{man, boat}\}$
3. $NP \rightarrow Det Adj N$	8. $V \rightarrow \{\text{man}\}$
4. $NP \rightarrow Det N$	9. $Adj \rightarrow \{\text{old}\}$
5. $VP \rightarrow V NP$	

Table 7: Production rules for (14)

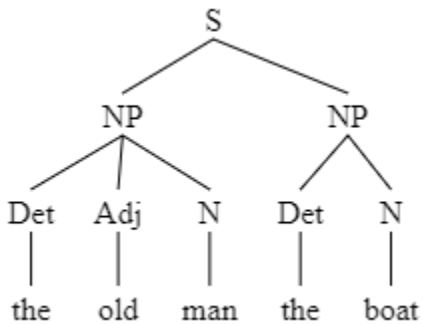
Based on this, we can begin the analysis:

---

<sup>17</sup> In sections 4.3 and 4.4, non-standard symbols were used to stay true to the authors who presented the analysis.

- a) Det old man the boat (6)
- b) Det Adj man the boat (9)
- c) Det Adj N the boat (7)
- d) NP the boat (3)
- e) NP Det boat (6)
- f) NP Det N (7)
- g) NP NP (4)
- h) FAIL

We can see that the choice in c) to utilize rule 7 to analyze *man* as a noun led us down the garden-path and produced a faulty analysis:

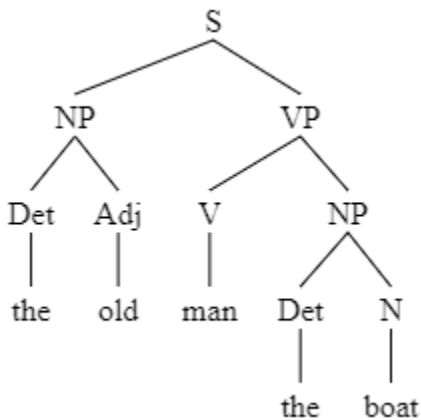


Tree Schematic 14: *The old man the boat*, with *man* as a noun

The parser has to backtrack and follow a different analysis:

- i) Det Adj man the boat (9)
- j) NP man the boat (2)
- k) NP V the boat (8)
- l) NP V Det boat (6)
- m) NP V Det N (7)
- n) NP V NP (4)
- o) NP VP (5)
- p) S (1)
- q) SUCCESS

Changing the choice of possible production rules for *man* from 7 to 8 allowed us to successfully complete the parsing and create a clean tree diagram:



Tree Schematic 15: *The old man the boat*, with *man* as a verb

The choice to parse *man* as a verb can be explained in two ways. The one we cover first is based on the ideas of Minimal Attachment/Late closure. In stage b), the parser had already processed Det and Adj. In stage c), it did not close the structure and start creating a new one, but rather remained in it to create a NP. As elaborated in 2.1.2, the Late Closure principle states that if possible, the new item will always be attached to the structure currently being processed, instead of jumping to create a new structure with the incoming item. As production rules presented in Table 6 allow for both options, the parser followed the principles of Late Closure. This creates a faulty interpretation, much like in (12), and the parser needs to backtrack and take the other available route.

Another way to explain this choice is to focus on the intrinsic properties of the word *man*. It is a polysemous word with more than one possible meaning. The online Cambridge Dictionary presents us with numerous nuances, but for the sake of simplicity, we will focus on two basic meanings, belonging to two different lexical categories: *man* as a noun and *man* as a verb<sup>18</sup>:

- 1) an adult male human being (noun)

<sup>18</sup> <https://dictionary.cambridge.org/dictionary/english/man>



- 2) to man something such as a machine or vehicle is to be present in order to operate it  
(verb)

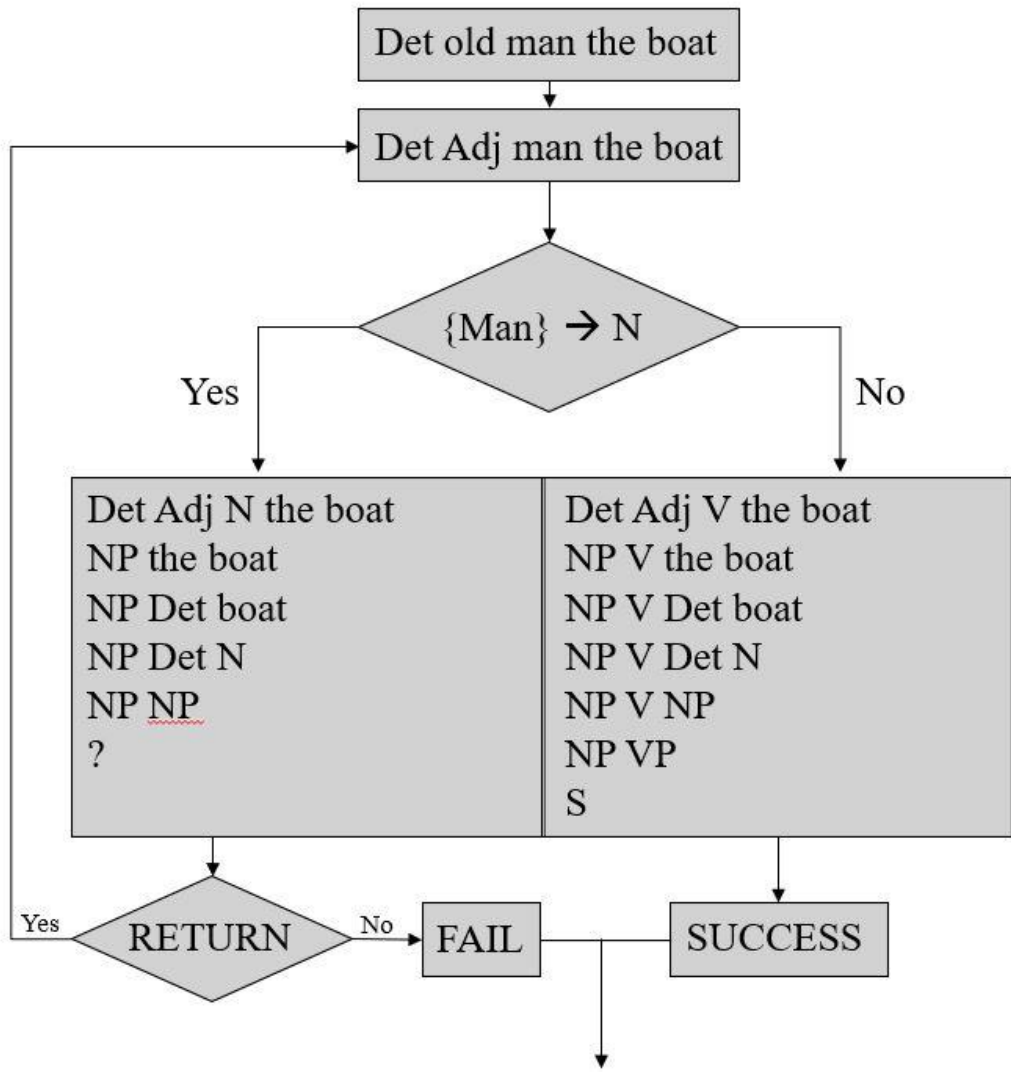
It is interesting to note that the first meaning, along the nuances connected to it pertaining to a noun, is presented first in the dictionary. This is not a coincidence. Let us look at the figures that the BNC can give us regarding the usage frequency of *man* as a noun and *man* as a verb:

<i>man</i> – noun (N)	94 496	841.12 per million
<i>man</i> – verb (V)	606	5.39 per million

Table 8: Different POS frequencies of *man*

It is clear that *man* is used as a noun far more frequently than as a verb. Based on the principles of Lexical Preference described in 2.1.3. the stronger, more salient form will always be a preferred choice for analysis. Since the word *man* is much more frequent as a noun than as a verb, it is to be expected that both human and computational parsers would opt to go with this analysis if possible. Naturally, in the case of garden-path sentences, this choice creates problems, and in order to achieve a successful interpretation, a less frequent, non-prototypical structure has to be maintained.

To conclude, whether the choices were guided by the principle of Late Closure or Lexical Preference, it is clear that the problem lies in taking the wrong turn when faced with multiple options. The problematic item in the sentence is the ambiguous *man*, having multiple lexical frames, that of a noun and that of a verb, which allows it to either be attached to a NP or serving as the head of a newly formed VP. In any case, regardless of the principle behind the choice, the whole process can be illustrated by the flowchart below. This schematic can be applied to various garden-path sentences to explain the two different paths taken by the parser, whether we are talking about a human analyzing the sentence, in which case this would be a concrete representation of an abstract thought process, or about a computer processing an ambiguous sentence input:



Schematic 16: Processing flowchart for (14)

## 6 Conclusion

The theoretical overview in this research presented some key psycholinguistic concepts involved in human sentence processing. By focusing on garden-path sentences as examples of structures difficult to process to show how these concepts operate, and by testing the hypotheses behind these psycholinguistics concepts through computational approaches, both rule-based and statistical, we have shown that a clear parallel can be drawn between the notions of Minimal Attachment/Late Closure and rule-based approaches, and the notion of Lexical Preference and statistical approaches. Each have their own pros and cons. As many things in language cannot be easily formalized, rule-based approaches fail where linguistic theory fails, too. Statistical approaches, however, make mistakes because they rely purely on numbers, failing in instances where rule-based methods would be able to discern the right interpretation from the wrong one provided they have a proper formal description of the problem.

Also, on a different note, regarding the applicability of theoretical concepts on practical NLP tasks, Du & Yu (2012) conclude that “the formal methods of computational linguistics, e.g. CFG, [...] and ATN, are useful for computational parsing,” (73) as they can guide the statistical methods from partly-parsed structures of garden-path sentences towards fully accepted sentences. Naturally, this is only applicable to a certain point. However, it is my personal belief, backed up by the research done in this thesis, that the most successful NLP models, which today mainly focus on implementing various strategies of supervised and unsupervised machine learning, can only progress so far without input from combined expertise of linguists and cognitive scientists. In my opinion, the future of technology lies in multidisciplinary, as the demand for more complex, more precise and more organic natural language processing will only grow in the coming decades. Naturally, in order to allow a machine to model human linguistic behavior, it needs to have access to the knowledge necessary to do that, meaning that linguists need to strive to improve their theoretical models as new empirical data comes to light, and to try to make those models as applicable within a multidisciplinary setting as possible. All of this does not only apply to language, but to other human behavior and capacities as well. Successfully modelling any kind of human thought processes, such as decision-making, is vital to numerous efforts in automation, from the automobile industry to the medical profession, to name but a few.

## 7 Appendix

### 7.1 List of analyzed sentences

(1)	Bill thought John died yesterday.....	8
(2)	The horse raced past the barn fell. ....	9
(3)	Since she jogs a mile seems light work. ....	10
(4)	John put the book on the table in the hall. ....	11
(5)	The boy got fat melted. ....	12
(6)	Since she jogs a mile seems light work. ....	13
(7)	The woman the man the girl loved met died.....	18
(8)	Women men girls love meet die. ....	19
(9)	Dan read the letter to Chrysanne.....	28
(10)	The student read the letter to Chrysanne fainted. ....	30
(11)	The new singers record the song.....	33
(12)	The new record the song.....	34
(13)	The complex houses married and single students and their families.....	38
(14)	The old man the boat.....	44

## 8 References

- Bever, T. (1970). "The Cognitive Basis for Linguistic Structures." In Hayes, R. (Eds.) *Cognition and the Development of Language*, 277-360, Wiley & Sons Inc, New York.
- Chomsky, N. (1957). *Syntactic Structures*. Mouton. The Hague.
- Clifton, C. & Staub A. (2008). "Parallelism and Competition in Syntactic Ambiguity Resolution." *Language and Linguistics Compass*, 2, 2, 234-250.
- Du, J. L. & Yu, P. F. (2012). "A computational linguistic approach to natural language processing with applications to garden path sentences analysis." *International Journal of Advanced Computer Science and Applications*, 3, 11, 61-75.
- Du, J. L., P. F. Yu & M. L. Li (2014). "Machine Learning from Garden Path Sentences: The Application of Computational Linguistics." *International Journal of Emerging Technologies in Learning*, 9, 6, 58-62.
- Elman, J. L. et al. (2005). "Cues, Constraints, and Competition in Sentence Processing." In M. Tomasello & D. I. Slobin (Eds.) *Beyond nature-nurture: Essays in honor of Elizabeth Bates*, 111–138, Lawrence Erlbaum Associates Publishers.
- Ford, M., J. Bresnan & R. Kaplan (1982). "A Competence-based Theory of Syntactic Closure." in J. Bresnan (Eds.) *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge.
- Franz, A. (1996). *Automatic Ambiguity Resolution in Natural Language Processing: An Empirical Approach*. Springer-Verlag. Berlin.
- Frazier, L. & Rayner, K. (1982). "Making and Correcting Errors During Sentence Comprehension: Eye Movements in the Analysis of Structurally Ambiguous Sentences." *Cognitive Psychology* 14, 2, 178-210.
- Frazier, L. & Rayner, K. (1987). "Resolution of Syntactic Category Ambiguities: Eye Movements in Parsing Lexically Ambiguous Sentences." *Journal of Memory and Language*, 26, 5, 505-526.

Frazier, L. (1979). *On Comprehending Sentences: Syntactic Parsing Strategies*. Ph.D. dissertation, University of Massachusetts.

Frazier, L., & Fodor, J. D. (1978). "The Sausage Machine: A New Two-stage Parsing Model." *Cognition*, 6, 4, 291-325.

Garside, R. (1987). "The CLAWS Word-tagging System," in: R. Garside, G. Leech and G. Sampson (Eds) *The Computational Analysis of English: A Corpus-based Approach*. Longman. London.

Gibson, E. (1991). *A Computational Theory of Human Linguistic Processing: Memory Limitations and Processing Breakdown*. Ph.D. dissertation, Carnegie Mellon University.

Kempen, G. (1994). "Computational models of syntactic processing in human language comprehension." in Dijkstra, T. & De Smedt, K. (Eds.) *Computational psycholinguistics: symbolic and subsymbolic models of language processing*, 192-220. Taylor & Francis. London.

Kimball, J. (1973). "Seven Principles of Surface Structure Parsing in Natural Language." *Cognition*, 2, 1, 15-47.

Lewis, R. (1998). "Reanalysis and Limited Repair Parsing: Leaping off the Garden Path." In Fodor J. D., Ferreira F. (Eds.) *Reanalysis in Sentence Processing, Studies in Theoretical Psycholinguistics*, 21, 247-285. Springer. Dordrecht.

Marcus, M. (1980). *A Theory of Syntactic Recognition for Natural Language*. Cambridge Press.

McRae, K. et al. (1998). "Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension." *Journal of Memory and Language*, 38, 3, 283-312.

McRoy, S. W. & G. Hirst (1990). Race-Based Parsing and syntactic disambiguation. *Cognitive Science*, 14, 3, 313-353.

Pritchett, B. (1988). "Garden Path Phenomena and the Grammatical Basis of Language Processing." *Language*, 64, 3, 539-576.

University Centre for Computer Corpus Research on Language. (n.d.). *CLAWS part-of-speech tagger for English*. Retrieved 15 May 2020 from <http://ucrel.lancs.ac.uk/claws/>.

Vosse T. & Kampen G. (2009). "In Defense of Competition During Syntactic Ambiguity Resolution." *Journal of Psycholinguistic Research* 38, 1, 1–9.

Wanner, E. (1980). "The ATN and the Sausage Machine: Which One is Baloney?" *Cognition*, 8, 2, 209-225.

Winograd, T. (1983). *Language as a Cognitive Process Vol. 1: Syntax*. Addison Wesley.

Woods, W. (1970). "Transition Network Grammars for Natural Language Analysis." *Communications of the ACM.*, 13, 11, 591–606.

## **9 Resources**

jsSyntaxTree (2003). <http://ironcreek.net/syntaxtree/>

SketchEngine (2004). <https://www.sketchengine.eu/>

The British National Corpus (2007). <http://www.natcorp.ox.ac.uk/>

The Cambridge Online Dictionary (1999). <https://dictionary.cambridge.org/>