

COMPUTER-AIDED ANALYSIS
OF ENGLISH PUNCTUATION
ON A PARSED CORPUS
THE SPECIAL CASE OF COMMA

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER
ENGINEERING AND INFORMATION SCIENCE
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Murat Bayraktar

September 1996

QA
76.9
.N28
B39
1996

COMPUTER-AIDED ANALYSIS
OF ENGLISH PUNCTUATION
ON A PARSED CORPUS
THE SPECIAL CASE OF COMMA

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER
ENGINEERING AND INFORMATION SCIENCE
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

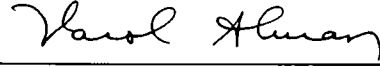
Murat Bayraktar

September, 1996

QA
46.9
.N38
B39
1986

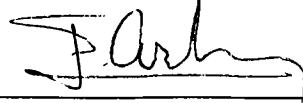
B 035250

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Prof. Varol Akman (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Prof. Erdal Arıkan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



Asst. Prof. Tuğrul Dayar

Approved for the Institute of Engineering and Science:



Prof. Mehmet Baray

Director of Institute of Engineering and Science

ABSTRACT

COMPUTER-AIDED ANALYSIS OF ENGLISH PUNCTUATION ON A PARSED CORPUS: *THE SPECIAL CASE OF COMMA*

Murat Bayraktar

M.S. in Computer Engineering and Information Science

Supervisor: Prof. Varol Akman

September, 1996

Punctuation, an orthographical component of language, has usually been ignored by most research in computational linguistics over the years. One reason for this is the overall difficulty of the subject, and another is the absence of a good theory. On the other hand, both 'conventional' and computational linguistics have increased their attention to punctuation in recent years because it has been realized that true understanding and processing of written language will be almost impossible if punctuation marks are not taken into account.

Except the lists of rules given in style manuals or usage books, we know little about punctuation. These books give us information about how we should punctuate, but they are generally silent about the actual punctuation practice. This thesis contains the details of a computer-aided experiment to investigate English punctuation practice, for the special case of comma (the most significant punctuation mark) in a parsed corpus. The experiment attempts to classify the various uses of comma according to the syntax-patterns in which comma occurs. The corpus (Penn Treebank) consists of syntactically annotated sentences with no part-of-speech tag information about individual words, and this ideally seems to be enough to classify 'structural' punctuation marks.

Keywords: Computational Linguistics, Natural Language Processing, Punctuation, English, Corpus-based Analysis, Comma.

ÖZET

İNGİLİZCE'DE NOKTALAMA İŞARETLERİNİN CÜMLE YAPISINA GÖRE NOTLANMIŞ BİR METİN VERİTABANINDA BİLGİSAYAR DESTEKLİ ANALİZİ: *VİRGÜLÜN ÖZEL DURUMU*

Murat Bayraktar

Bilgisayar ve Enformatik Mühendisliği, Yüksek Lisans

Tez Yöneticisi: Prof. Dr. Varol Akman

Eylül 1996

Dilin yazımsal bir ögesi olan noktalama, bilgisayarlı dilbilim alanındaki araştırmalarda yıllar boyu ihmal edilegelmiştir. Bunun bir nedeni konunun genel zorluğu, diğer bir nedeni de dayanak noktası olabilecek sağlam bir teorinin eksikliğidir. Öte yandan, son yıllarda gerek 'geleneksel' gerekse bilgisayarlı dilbilim alanlarının noktalamaya ilgisi giderek artmıştır; çünkü, noktalama işaretlerini dikkate almadan yazılı dili gerçekten anlayıp işlemenin neredeyse imkansız olduğu ortaya çıkmıştır.

Biçim kılavuzları ve genel dilbilgisi kitaplarında verilen kural listeleri dışında noktalama hakkında az bilgiye sahibiz. Bu tür kitaplar noktalama işaretlerinin nasıl kullanılacağına dair bilgiler verirken, bunların uygulamada nasıl kullanıldığı konusunda genelde sessiz kalmaktadırlar. Bu tez, İngilizce'de noktalama uygulamasının, virgölün (noktalama işaretlerinin en önemlisi) özel durumu için, cümle yapısına göre notlanmış bir metin veritabanında incelenmesi amacıyla yapılmış bilgisayar destekli bir deneyin ayrıntılarını içermektedir. Bu deneyde, virgölün değişik kullanımlarını cümlede ortaya çıktığı değişik sözdizimi şablonlarına göre sınıflandırmaya çalıştık. Kullanılan metin veritabanı (Penn Treebank) sadece sözdizimi yapısına göre notlanmış cümlelerden oluşup başka hiçbir bilgi içermemekte, bu ise yapısal noktalama işaretlerinin sınıflandırılması için ideal olarak yeterli görünmektedir.

Anahtar sözcükler: Bilgisayarlı Dilbilim, Doğal Dil İşleme, Noktalama, İngilizce, Metin-tabanlı Analiz, Virgöl.

I am very grateful to my supervisor, Prof. Varol Akman, for his guidance and motivating support during this study. It was a pleasure to work with him.

I would like to thank Prof. Erdal Arıkan and Asst. Prof. Tuğrul Dayar for reading and commenting about the thesis.

I owe special thanks to my colleague Bilge Say for her voluntary intellectual guidance and for informing me about the recent related work on punctuation.

I would like to thank my colleagues Dilek Z. Hakkani, Gökhan Tür, A. Kurtuluş Yorulmaz, Yücel Saygın and everybody who have in some way contributed to this study by giving intellectual support and especially by helping with \LaTeX .

I also thank all my friends and especially Yasemin İçel for lending me moral support whenever I was in need of it.

Finally, I would like to express my deep gratitude to my family whom I owe everything for my present position.

I dedicate this thesis to my mother.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	This Study	2
2	Punctuation	3
2.1	History of Punctuation	3
2.2	Modern English Punctuation	5
3	Related Work	8
3.1	Related Work in Linguistics	8
3.2	Related Work in Computational Linguistics	11
4	The Comma	16
4.1	Significance	16
4.2	Classification of Potential Uses of Comma	17
4.2.1	Elements in a Series	19
4.2.2	Sentence-initial Elements	20

4.2.3	Sentence-final Elements	21
4.2.4	Nonrestrictive Phrases or Clauses	21
4.2.5	Appositives	22
4.2.6	Interrupters	23
4.2.7	Quotations	23
5	The Corpus	25
5.1	The Penn Treebank	26
5.2	Structure of the Parsed Corpus	28
5.3	Constructs Related to Comma	31
5.3.1	Appositions	31
5.3.2	Coordinations	33
5.3.3	Gapping	34
5.3.4	Verbs of Saying	34
5.4	Problems with the Corpus	35
6	The Experiment	37
6.1	Implementation	37
6.1.1	Preprocessing the Corpus	38
6.1.2	Construction of the Syntax-pattern Database	41
6.2	Classification of the Syntax-patterns	42
6.3	Results of the Classification	45
6.4	Verification of the Classification	48

<i>CONTENTS</i>	viii
7 Conclusion	49
A Samples from the Corpus	57
A.1 Raw Format	57
A.2 Tagged Format	58
A.3 Parsed in LISP Format	59
A.4 Converted to Prolog Format	63
B Sorted List of Syntax-patterns	67
C Outputs of the Classification	74
C.1 Classified Syntax-patterns	74
C.2 Similar Syntax-patterns Brought Together	82
D Source Code of the AWK Program	89
E Source Code of the Prolog Program	92

List of Tables

4.1	Distribution of Punctuation Marks in Meyer's corpus	17
5.1	Contents of the Penn Treebank	27
5.2	Syntactic Tag-set of the Penn Treebank	30
6.1	Results of the Classification	46

Chapter 1

Introduction

1.1 Motivation

Until recently, punctuation had usually been neglected by most research both in ‘conventional’ (theoretical) and computational linguistics. This is not only due to the overall difficulty and complexity of the subject, but also due to the absence of a concise, theoretical and descriptive background for the abstract problem. However, once we remember that punctuation is an orthographical component of written language—that correct punctuation is almost as important as other essentials of written language such as correct spelling, good style and proper structure—we see that research on punctuation makes reasonable sense. Accordingly, interest in the subject rose within the recent years because it has been realized that fuller understanding and processing of written language is quite impossible without taking punctuation into account. Although punctuation was originally invented as a device for reflecting intonation in written text, it is now a linguistic “system on its own right” [32, p. 9] and “the only function of punctuation is making writing clearer for the reader” [11, p. iii]. We can logically infer that clearer reading means clearer understanding of language for the linguist, and—in the case of computational linguist—precise processing of natural language.

1.2 This Study

This thesis reports the details of a study in which it was attempted to analyze English punctuation practice in a computer-aided experiment. The material analyzed was a syntactically annotated (i.e., parsed) corpus, which was a part of the bracketed version of the Penn Treebank [27]. Due to its higher significance compared to other punctuation marks, only the comma was investigated. The purpose of the investigation was to classify various structural¹ uses of comma in the given corpus and observe their frequencies. The classification made by Ehrlich [11] was taken as a basis, although it was reorganized, and supported by other references concerning the subject. The corpus consists of syntactical analyses (parse trees) of sentences with no part-of-speech tag information about the individual words. For the classification, abbreviated syntax-patterns containing the comma as an immediate daughter were extracted and intuitively assigned to appropriate classes by looking at sample sentences containing these patterns. Observing this classification, frequencies of the individual uses of comma in the analyzed corpus were reported. A final experiment was done to verify the classification, based on the syntax-patterns. It turns out that a parsed corpus is sufficient for doing such a classification.

The remainder of this thesis is organized as follows. In Chapter 2, a short history of punctuation is given along with an appraisal of the current state of modern English punctuation. This is followed by Chapter 3, which is a brief survey of recent related work both in theoretical and computational linguistics. Chapter 4 starts with a discussion on the significance of comma and ends with the classification of its uses employed in this study. Information about the contents and the structure of the Penn Treebank is offered in Chapter 5, followed by a description of the problems experienced with this corpus. Chapter 6 contains the details of the implementation and the results of the experiments. The thesis is concluded with Chapter 7, where a discussion and suggestions for further work can be found.

¹Structural punctuation is elaborated in Section 2.2.

Chapter 2

Punctuation

The journey of punctuation through history is closely parallel to the development of written language, since punctuation is an essential element of it. Section 2.1 is the story of this journey until today. Section 2.2 surveys the current state of modern English punctuation.

2.1 History of Punctuation

The system of punctuation can be traced back to the systems employed in ancient Greece and Rome, where a written text was only used as a prepared speech. So, punctuation originally emerged from the need for a system that would show the orator when to stop and take a break during her speech [35].

The word *punctuation* comes from the Latin word *punctus*, meaning ‘a point’. Between the 15th and 18th centuries, the subject was known as *pointing*; the term *punctuation*, first recorded in the 16th century, meant the insertion of vowel points in Hebrew texts. These two words (i.e., *pointing* and *punctuation*) exchanged their meanings in the early 18th century [49].

Only three points were used by the ancient Greek grammarians, who placed them high, low or mid-line to indicate grammatical units and subunits. Hebrews used vowel signs and accents above or below the lines of holy text, the

Masorah. In the medieval times, English scribes usually employed a medial point [·], an inverted semicolon [∴] and a virgule [/] [52].

The invention of print in 1448 by Gutenberg was the starting point for the divergence of spoken and written language. Mass literacy improved rapidly and elocutionary punctuation became insufficient for the purposes of written language. Moreover, there were no standards and many inconsistencies within the punctuation system, which led to confusion and quarrels among printers and writers [35]. Manutius, who was a Venetian editor and printer, introduced a new system of punctuation in his *Orthographiae ratio* ('System of Orthography'), published in 1566 [33, 49]. He is considered to be the father of modern punctuation [56]. His work included the modern comma, semicolon, colon and period. Furthermore, Manutius voiced for the first time the view that clarification of grammatical structure (of the sentence) is the main function of punctuation. Following him, various punctuation marks received their now standard names, and new marks such as the exclamation mark, question mark, and the dash were added by the end of the 17th century [49].

Manutius had started the division of the theory and practice of punctuation into two main schools of thought. The *elocutionary* school, following the traditional practice, viewed punctuation marks as indications of pauses of various lengths observed by a reader who was reading aloud to an audience [49]. This view even reached to the point, where there were four separate lengths of pauses assigned to the comma, semicolon, colon and period (comma denoting the shortest pause and period the longest) [29]. The *syntactical* or *grammatical* school, winning the argument by the end of the 17th century, saw punctuation marks as indicators of the grammatical construction of sentences [35, 49]. Today, most writers agree that the main function of punctuation is to clarify the grammatical structure of a text. However, they also think that it has to take account of the speed and rhythm of actual speech: pauses in speech and breaks in syntax converge in many cases [49].

2.2 Modern English Punctuation

The modern dictionary [53] definition of punctuation goes as follows: “the practice, method, or skill of inserting points or marks in writing or printing, in order to aid the sense; division of text into sentences, clauses, etc. by means of such marks; the system used for this; such marks collectively. Also observance of appropriate pause in reading and speaking.” Other dictionaries [50, 51] state more or less the same, one [54] adding that “the marks of punctuation, originally conventionalized from normal speech patterns of pause, pitch and stress, no longer correspond with these in detail.” Although it seems that the last sentence of the former definition conflicts with the latter statement, this surely is not the case. The fact that punctuation is no more used as an intonation device does not imply that it hinders “the observance of appropriate pause.” In reading written material, the reader has to consider, among other things, punctuation in order to pronounce the meaning in the proper sense.

A popular practice today is to give the word punctuation a broader meaning [32, p. 17]: punctuation is “a set of non-alphanumeric characters that are used to provide information about structural relations among elements of a text, including commas, semicolons, colons, periods, parentheses, quotation marks and so forth. From the point of view of function, however, punctuation must be considered together with a variety of other graphical features of the text including font- and face-alternations, capitalization, indentation and spacing, all of which can be used to the same sorts of purposes.”

The last definition allows us to view modern punctuation as falling into roughly three categories:

- *Within word*: Hyphens, apostrophes, commas within numbers, periods within numbers and abbreviations, etc.
- *Between words*: What we traditionally think of as punctuation, e.g., commas, periods, colons, semicolons, exclamation marks, question marks, quotation marks, dashes and parentheses.

- *Higher-level graphical punctuation*: Paragraphing, indentation, underlining, font changes, general layout conventions, etc.

This categorization forces us to narrow our scope by defining the concept of *structural* punctuation marks [29, 30, 31]. These are the marks which we conventionally consider as punctuation marks, those that are to be found between the words of a sentence. These marks do not set off constructions larger than the sentence or smaller than the word. Unless otherwise indicated, the word *punctuation* will be used as a shorthand for the phrase *structural punctuation* in the rest of this thesis.

The focus of this study is the structural uses of the comma in English. These uses will be explained in detail in Chapter 4. The functions of the remaining structural punctuation marks in English can be summarized as follows [48]:

- The *period* [.] , which is also called *full stop*, is used at the end of a declarative or imperative sentence.
- The *question mark* [?] is put at the end of an interrogative sentence or phrase.
- The *exclamation mark* [!] is found at the end of an emphatic, loud or highly charged statement.
- The *colon* [:] is used to introduce an illustration, amplification or analysis, immediately after a main clause.
- The *semicolon* [;] is used in compound sentences to connect independent sentences, as a kind of conjunction. Another function of this mark is to separate the items of a series when commas are already present within the items.
- The *dash* [—] usually sets off items that seem to come after a break in the thought of the flowing text. This thought may be an interpolation, a kind of second thought, or a final statement.
- *Quotation marks*, double [“ ”] or single [‘ ’], are used to enclose directly quoted material, or to set off items that are brief allusive quotations or special kinds of vocabulary.

- *Parentheses* [()] enclose optional but still useful material that does not fit with close logic into the flow of the text.

As there are differences between American English and British English, there are also nuances between American and British punctuation practices. This fact is mentioned in a number of works [29, 30, 34, 35] concerning punctuation. Some of them [29, 34] include whole chapters on this subject. Nevertheless, while the two practices are usually viewed as being quite different, in fact, they are similar [29]. According to Clark [7, p. 211], the only important difference between the two systems is that “American punctuation tends to be more rigid than British, and more uniform, more systematic, and easier to teach and, once learned, easier to use.” Today, the widely accepted system of English punctuation is the American one. On the other hand, the corpus used in this study was not intentionally chosen for its American origin. After all, this should not be very significant, since the abstract problem of punctuation is universal.

Chapter 3

Related Work

If we look at related work focusing on punctuation, we detect two kinds of studies: linguistic work related to punctuation, and works within the framework of computational linguistics, which mostly attempt to take punctuation marks into account in Natural Language Processing (NLP).

3.1 Related Work in Linguistics

Humphreys [16, p. 199] states that “there are three sorts of books on punctuation. The first ... is selflessly dedicated to the task of bringing punctuation to the Peasantry ... The second sort is the Style Guide, written by editors and printers for the private pleasure of fellow professionals. The third, on the linguistics of punctuation system, is much the rarest of all.”

This section mainly focuses on the third sort of studies, since they [29, 32] make the largest contribution to the construction of a coherent theory of punctuation. But first, other sorts of studies are going to be mentioned briefly.

Introductory, intermediate and advanced composition handbooks and grammar books (such as [12]) are pedagogical approaches punctuation. These books usually contain lengthy passages for a better treatment of individual punctuation marks. Discussions of punctuation addressing a general audience are

style manuals such as the famous *Chicago Style Manual*, dictionaries such as the *Webster's II, New Riverside University Dictionary* [55], and full-length books [11, 17, 28, 34, 35] on the correct usage of punctuation. The common approach among these studies is that they employ a prescriptive treatment of punctuation (rarely, some of them contain descriptive discussions): long lists of rules for correct punctuation are given, but the actual practice is not considered.

Meyer's Ph.D. thesis [29] is the first example of a wholly descriptive study of punctuation as a system. Focusing on the American practice of using structural punctuation marks and working on 12 samples, each consisting of about 2,000 words, from the Brown Corpus [13] in fiction, journalistic and learned styles (i.e., $12 \times 2,000 \times 3 = 72,000$ words in total), he classifies and illustrates punctuation functions and how these functions are realized. An important observation he makes is that functions of marks and their realizations are distinct concepts; this is usually ignored within prescriptive work. According to Meyer, there may be three kinds of function of punctuation marks: to help the reader to understand the text easily, to emphasize a concept, to vary the rhythm of the text. The realization of those functions, on the other hand, fall into two main categories: marks that separate and marks that enclose. He proceeds with giving a detailed description of boundaries that are separated or enclosed by punctuation marks. Clauses, phrases and words are syntactic boundaries; questions, modifiers, etc. are examples for semantic boundaries; pauses, tone units, and changes in pitch and stress constitute the prosodic boundaries. A given punctuation mark in a sentence may determine more than one kind of boundary, but one of these is usually dominant.

Nunberg [32, p. 9] admits that Meyer's work is a "useful and thorough survey of the use of American punctuation." However, he also criticizes him for not viewing punctuation as a system on its own right, but only focusing on "the relation of punctuation to lexical structures."

Nunberg's *The Linguistics of Punctuation* [32] was the basic motivation for research in computational linguistics on the subject of punctuation in the 90's. In this important study, he attacks the general opinion that punctuation is prescriptive and only a device for reflecting intonation, and claims that this belief

is the major reason for the negligence of punctuation within the linguistics community. He admits that the origin of punctuation was the transcription of intonation, but also adds that after the divergence of written and spoken languages, punctuation has become a linguistic system on its own right. He proposes to use two separate grammars to analyze texts. A *lexical grammar* accounts for the text-categories (text-clauses, text adjuncts and text-phrases) occurring between the punctuation marks; and a *text-grammar* deals with the structure of punctuation, and the relation of punctuation marks to the text-categories they separate. He introduces the following rules for handling the interactions:

- *Point absorption*: For all of the point symbols (marks except bracket symbols, like parentheses or quotation marks), if two points are immediately adjacent, the stronger point absorbs the weaker one according to the following fixed hierarchy in ascending order, the comma being the weakest point: comma, dash, colon, semicolon and period.
- *Bracket absorption*: A point standing directly to the left of a closing quotation mark or parenthesis is removed. There may be some exceptions for the period.
- *Quote transposition*: Point symbols occurring directly to the right of a closing quotation mark are moved to the left of that mark. It is important to employ this rule carefully (in appropriate order with the bracket absorption rule), so that a character is not absorbed just after it is transposed from the right side to the left side of a quotation mark.
- *Graphic absorption*: Symbols that are orthographically same or similar to each other, but differ linguistically, are absorbed. For example, if a period marking an abbreviation and another one marking the end of a sentence occur adjacently, one of them is removed. Conversely, they stay in their places, if the second mark is a comma, a question mark or an exclamation mark.
- *Semicolon promotion*: If an item of a list itself contains a point symbol, the commas separating the items in the list may be promoted to semicolons to prevent ambiguity.

According to Jones [18, p. 4], the phenomena described by the rules given above and other phenomena discussed in Nunberg's book "will be fundamental to the implementation and treatment of punctuation in any framework," although in a recent paper [22, p. 363] he criticizes the book for being "a little too vague to be used as the basis of any implementation." Another comment is given by Humphreys [16, p. 201]: "Anyone considering a treatment of punctuation within natural language processing will want to read this book." Indeed, as can be observed in the next section, almost every study involving punctuation within the NLP framework is based on Nunberg's ground-breaking work.

3.2 Related Work in Computational Linguistics

One of the first studies accounting for punctuation in computational linguistics research is by Garside et al. [14], who undertook a research program during 1976–1986, to base NLP on the probabilistic analysis of a large corpus. During the tagging stage, they employ punctuation marks, which are tagged with themselves, to solve ambiguities. As another project within their research program, they describe a method called 'automatic intonation assignment', which tries to derive a prosodic transcription from written forms of spoken text under the guidance of punctuation.

Karlsson et al. [25] introduce a Constraint Grammar in their more recent NLP work involving punctuation marks. This grammar is a morphological and syntactic parsing scheme for language-independent, unrestricted text. When syntax-based methods fail, they employ optional heuristics. Their aim is to disambiguate parsing by discarding improper alternatives using several constraints. Typographical features such as punctuation or capitalization are some of their 24 simplifying tools. Punctuation marks are used in the detection of comma-delimited clause boundaries, or adjective or adverb lists with a limited variety of separating marks.

There are several works on the semantic information carried by punctuation

marks. Dale [9, 10] investigates the role of punctuation within discourse structure. Similarly, Say and Akman [40, 41, 42] examine the information-based aspects of punctuation by formulating their treatment in Discourse Representation Theory [24].

Following the publication of Nunberg's book, many works appeared, explicitly focusing on the involvement of punctuation marks within the NLP context. Srinivasan [43] investigates the possibility of using punctuation in lexicography and abstracting. According to him, it is important to derive information from real-word texts using punctuation. He makes the classification of the uses of punctuation marks into four groups: separating, delimiting, distinguishing and morphological. As an experiment, he constructs an expanded lexicon to be used in machine translation, employing information derived from the involvement of punctuation marks.

Jones [18, 19] starts his research on the potential role of punctuation within the NLP framework by asking the question, "Can punctuation help parsing?". Taking Nunberg's work as a basis, he investigates parsing with a feature-based tag grammar. On the other hand, instead of using a two-level grammar as suggested by Nunberg, he prefers an integrated grammar, which deals with words and punctuation marks simultaneously.¹ Jones takes an existing grammar for English and extends it by introducing the notion of *stoppedness* (of a category), to handle punctuation explicitly. A punctuational character following a category in the grammar is described by a *stop* feature. The rules, based on this notion, cover the optionality of certain marks and the absorption rules introduced by Nunberg. Jones tests this grammar on the Spoken English Corpus [45], which contains sentences of various lengths, which in turn are rich in punctuation. This feature of the corpus allows him to view the advantages or disadvantages of accounting for punctuation during the parsing process. At the end, he concludes that the involvement of punctuational phenomena within parsing reduces the number of parses of complex sentences, contributing to the solution of the ambiguity problem. As a final remark, he observes that the ambiguity of complex sentences may be related to the number of elements that

¹He criticizes the two-level grammar for making the process unnecessarily complex by causing extra interactions between the levels. Therefore, he does not see any advantage in using this approach.

occur between punctuation marks.

Another approach to parsing sentences with punctuation marks is by Briscoe and Carroll [3, 4]. As opposed to Jones, they follow Nunberg's path more 'loyally' and build a two-level grammar by tokenizing punctuation marks separately from words. As a lexical-grammar they use a unification based one and give the role of the text-grammar to a probabilistic LR parser. The former is a Definite Clause Grammar [36], which is integrated into another one for part-of-speech tagging. The last step is the integration of this grammar with the LR parser. The result is a more modular tool than Jones's, since text-categories and syntactic categories are treated as overlapping, and disjoint sets of features are dealt with in each grammar separately. They use two different corpora to test their grammar: the Spoken English Corpus [45] and the SUSANNE Corpus [37]. They interpret their results according to various performance factors, and propose, as a future work, to develop semantic rules for several text-unit and text-adjunct combinations.

White [47] considers Natural Language Generation (NLG), and tries to look at the problem from that perspective. He examines how to integrate Nunberg's approach to presenting punctuation (and other phenomena) into NLG systems. He investigates Nunberg's punctuation presentation rules and gives example cases where some rules work fine in parsing but overgenerate from a generation perspective. He then builds his implementation on a layered architecture, which has three components: syntactic, morphological and graphical. In order to overcome several shortcomings of Nunberg's analysis, White tries to put the rules in the generation process into action as early as possible.

The most closely related work to our study is Jones's very recent PhD thesis [23].² Jones, finding Nunberg's approach inappropriate to be used as the basis of any implementation, stresses the need for a new theory of punctuation which is suitable for computational implementation. He first carries out a study [20] displaying the variety of punctuation marks and their orthographic interactions. In this study, he points to the existence of a set of more unusual symbols—besides the set of symbols that we conventionally regard as punctuation, accounting for the majority of punctuation in written language—usually

²Our work was carried out independently from Jones's work.

with a higher semantic content, which are usually specific to the corpus in which they appear and therefore are less suitable for a standardized treatment. He also shows that the average number of punctuation symbols to be expected in a sentence of English is four, which proves the necessity for the inclusion of punctuation in language processing systems.

Jones further continues with another research [22] to examine the true syntactic function of punctuation marks in text. For him, there may be two possible approaches to this problem: an *observational* one and a *theoretical* one. He tries to adopt both of these approaches, in order to be able to compare the results, hoping to combine them in the future. For the observational part, he chooses the Dow Jones section of the Penn Treebank [27], which is a large (approx. 2 million words), parsed corpus and is therefore suitable for the investigation of grammatical punctuation usage. Jones collects each node that has a punctuation mark as its immediate daughter in the parse tree, and abbreviates its other daughters to their categories, as is shown in examples (1)-(3) [22, p. 364]:³

(1) [NP [NP the following] :] \implies [NP \rightarrow NP :]

(2) [S [PP In Edinburgh] , [S ...]] \implies [S \rightarrow PP , S]

(3) [NP [NP Bob] , [NP ...] ,] \implies [NP \rightarrow NP , NP ,]

He proceeds with grouping different syntax-patterns into different sets for each punctuation mark and derives rule-patterns representing the behavior of individual marks, using common properties among syntax-patterns within a set. As a result, he reduces the 12,700 unique syntax-patterns, found in the corpus, to just 137 rule-patterns for the colon, semicolon, dash, comma and period. He reduces this number further to 79, employing a pruning procedure, where he removes idiosyncratic, incorrect and exceptional rule-patterns. Using this reduced set of rule-patterns, he derives some generalized punctuation rules,

³See Table 5.2 on page 30 for the meanings of the abbreviations (such as NP) in the examples.

which he describes in [21] in detail and suggests the integration of these rules into a normal syntactic grammar to add punctuation capabilities. He gives, among other rules for other punctuation marks, rule (4) [22, p. 364] for potential syntax-patterns in which the comma may appear:

$$(4) \mathcal{C} \longrightarrow \mathcal{C} , * \qquad \mathcal{C} : \{NP, S, VP, PP, ADJP, ADVP\}$$

$$\mathcal{C} \longrightarrow * , \mathcal{C}$$

In his theoretical approach, Jones starts with introducing the following hypothesis, based on his observations [22, p. 364]: punctuation seems to come “immediately before or after a phrasal level lexical item (e.g., a noun phrase).” To verify this hypothesis he looks at several real-life examples and tries to fine-tune his generalization by restricting or relaxing his hypothesis, whichever the case in question demands. The adjustments lead him to the conclusion that punctuation could be described as being either adjunctive or conjunctive.

As a next stage of his ongoing research, Jones proposes to verify and compare the results of both approaches, and hopes to be able to combine them. Finally, he suggests, as a further work, an investigation of the semantic function of punctuation marks, which is done partly in the present thesis.

Chapter 4

The Comma

The etymological root of the word *comma* does not conflict with its meaning in current practice: the word *comma*, which is originally Latin, comes from the Greek word *komma*, related to *koptein*, ‘to cut’, means literally ‘a part cut off’. In the context of punctuation, the word *comma* means a clause, a phrase or a word, cut off from the rest of the sentence, or the sign that indicates this separation [34]. Section 4.1 explains the importance of this punctuation mark. Section 4.2 gives a classification for the different uses of comma.

4.1 Significance

The comma has been described as “the most ubiquitous, elusive and discretionary of all stops” [17, p. 10], since it is the most frequent and most versatile mark that can be observed in any given text taken from any domain of literacy. Meyer [29] gives some numbers that indicate the percentages of individual marks to the total number of all structural punctuation marks encountered in the corpus he worked on (Table 4.1).

It can be argued that, other marks on the side, the period may be at least as important as the comma, since their frequencies are almost the same. However, the comma beats the period with its versatility, which can best be illustrated by the interesting data obtained by Jones [22]. As it was already mentioned in

Mark	Percentage
comma	46%
period	45%
dash	2%
parenthesis	2%
semicolon	2%
question mark	1%
colon	1%
exclamation mark	1%

Table 4.1: Distribution of punctuation marks in Meyer’s corpus (adapted from [29, p. 18])

Section 3.2, Jones groups different syntax-patterns into different sets for each punctuation mark. At the end, he observes 12,700 unique syntax-patterns in total for all punctuation marks, where the cardinality of the set for the comma is 9,320, which makes about 73% of the patterns. It can be inferred that the high frequency of the occurrence of comma is the result of this versatility.

In the light of the above facts, the comma can easily be declared as the most significant structural punctuation mark. Therefore, it is fitting to dedicate a study to the comma. Furthermore, such a study could be a guide for the investigation of the remaining marks.

4.2 Classification of Potential Uses of Comma

The number of classes mentioned for the uses of comma differ from two to 10 or 20 in different studies done on punctuation, depending on the potential audience of the study in question. Those works in the linguistics camp [29, 32] prefer to be as general and theoretical as possible, whereas those on the teaching side (e.g., style guides or punctuation usage books) [11, 17, 34, 35] try to cover and illustrate all possible uses, for the benefit of the reader.

Nunberg [32], for example, recognizes only two main classes for the use of comma: the *delimiter* comma, which encloses certain elements either at both

ends (when the element is within the clause) or at one end (when the element is either at the beginning or at the end of the clause); and the *separator* comma, which is put between members of certain types of coordinate elements. He also mentions a probable third type of comma, the *disambiguator*, but notes that this can be seen as a *separator*, this time separating elements of different syntactic types, as in example (5) [32, p. 37]:

(5) Those students who can, contribute to the United Fund.

Meyer [29], also in the linguistics camp, prefers to use a two-level classification. In the first level, there are two main classes: marks that *enclose* and marks that *separate*. This maps directly to Nunberg's classification. In the second level, however, Meyer becomes more specific by reporting that the comma may *enclose* coordinate elements, adverbials or modifiers, and only *separate* coordinate elements.

For the purposes of our study, we need a more specific classification of potential uses of the comma in the corpus, in order to be able to group the syntax-patterns containing the comma later into these classes. At this point, it is more reasonable to refer to sources such as style guides or punctuation usage books. There are plenty of such books on the market [11, 17, 34, 35], each making a different classification. Since there is no consensus among these works, it would be wrong to say that one of them shows the 'correct' classification. Therefore, it is plausible to simply select one of them—preferably a popular one, one which affects actual punctuation practice more widely—and complement its shortcomings with the others.

The following classification, which will be used in the upcoming chapters, is mainly based on Ehrlich's book [11]. In cases where it came short for the needs of the corpus, we referred to other books [17, 34, 35]. At some points, the classification is reorganized by making some classes subclasses of other classes. Furthermore, cases for the use of unstructural comma (such as the comma in numbers, dates and addresses) are discarded, since they are out of the scope of this study. Every class is supported with examples to make its character more understandable. The examples are taken from the corpus, whenever possible.

4.2.1 Elements in a Series

One of the frequent uses of comma is the separation of three or more elements listed in a series. The elements may be words (6), phrases (7) or clauses (8) having the same syntactic type. The last element is usually separated by a conjunction such as *and* or *or*, and seldomly by another comma.

(6) Elsewhere, share prices closed higher in *Amsterdam, Brussels, Milan and Paris*. (from Penn Treebank (PT))

(7) We innovated *telephone redemptions, daily dividends, total elimination of share certificates and the constant \$1 per share pricing*, all of which were painfully thought out and not the result of some inadvertence on the part of the SEC. (from PT)

(8) *John went shopping, Mary cooked the meal and David washed the dishes.*

In some cases, the conjunction may be preceded by a comma, in order to prevent misreading. Ehrlich names this as the *bacon-and-eggs* problem and gives examples (9) and (10) [11, p. 17]:

(9) You may order anything you want at my dinner as long as you order *sausage and eggs, ham and eggs, or bacon and eggs*.

(10) The chef said he needed *sausage, ham, bacon, and eggs*.

Independent clauses joined by a coordinating conjunction, such as *and, or, but, etc.*, may be separated by a comma, if there is a risk of misreading, as in (11):

(11) The Red Cross doesn't track contributions raised by the disaster ads, but it has amassed \$46.6 million since it first launched its hurricane relief effort Sept. 23. (from PT)

Coordinate adjectives as in (12), which independently modify a noun, are separated by commas, if otherwise the meaning changes:

- (12) And some US army analysts worry that the proposed Soviet redefinition is aimed at blocking the US from developing *lighter, more transportable, high technology* tanks. (from PT)

4.2.2 Sentence-initial Elements

A comma may delimit long phrases or clauses, that appear sentence-initially as an introductory element, if there is a possibility of misleading the reader. This can be seen by looking at examples (13) and (14) for phrases and clauses respectively, and trying to read the sentences without the comma:

- (13) *Under two new features*, participants will be able to transfer money from the new funds to other investment funds or, if their jobs are terminated, receive cash from the funds. (from PT)
- (14) *Although the action removes one obstacle in the way of an overall settlement to the case*, it also means that Mr. Hunt could be stripped of virtually all of his assets if the Tax Court rules against him in a 1982 case heard earlier this year in Washington, D.C. (from PT)

Introductory modifiers, such as adjectives (15), adverbs (16) or participles (17), which usually consist of one word, are usually set off by a comma:

- (15) *Victorious*, the army withdrew a thousand meters and encamped for the night. (from [11, p. 25])
- (16) *Clearly*, the judge has had his share of accomplishments. (from PT)
- (17) *Running*, he went up the stairs.

An absolute phrase may appear sentence-initially, in which case it is always delimited by a comma, since it modifies the entire sentence and has no grammatical connection to any other element in the sentence, as in (18):

- (18) *The party over*, the couple began to wash a sinkful of dishes. (from [11, p. 37])

It is noted that absolute phrases differ from other phrases in their capability of expressing a full idea, but unlike clauses, they only consist of a subject and a modifier.

4.2.3 Sentence-final Elements

Like sentence-initial introductory elements, sentence-final complementary elements are delimited by a comma, if there is a need for disambiguation. The element may be a phrase (19), a subordinate clause (20) or an absolute phrase (21):¹

- (19) A bomb exploded at a leftist union hall in San Salvador, *killing at least eight people and injuring about 30 others, including two Americans*. authorities said. (from PT)
- (20) A face-to-face meeting with Mr. Gorbachev should damp such criticism, *though it will hardly eliminate it*. (from PT)
- (21) She ran faster, *her breath coming in deep gasps*. (from [35, p. 31])

4.2.4 Nonrestrictive Phrases or Clauses

Postmodifiers of nouns, which may be phrases or clauses, are enclosed by commas if they are nonrestrictive. Restrictive modifiers identify, define or limit

¹This class of usage was omitted by Ehrlich [11], except for the case of subordinate clauses and absolute phrases, which were shown as individual classes. In the corpus, I encountered sufficiently many examples involving sentence-final verbal phrases, so that it became mandatory to have this class.

the elements they modify, and thus are essential for the intended meaning of the sentence. A nonrestrictive modifier, on the other hand, may be removed without changing the intended meaning of the sentence, since it only adds information concerning an element already identified, defined or limited. Examples (22) and (23) show restrictive phrases and clauses, respectively, whereas (24) and (25) show nonrestrictive ones:

(22) The man *at the left* is taller.

(23) He was the only student *who answered all the questions in the exam*.

versus

(24) A Western Union spokesman, *citing adverse developments in the market for high-yield “junk” bonds*, declined to say what alternatives are under consideration. (from PT)

(25) At one point, almost all of the shares in the 20-stock Major Market Index, *which mimics the industrial average*, were sharply higher. (from PT)

4.2.5 Appositives

Appositives, also known as noun repeaters, identify or point out to the nouns they succeed. Only nonrestrictive appositives are delimited by commas, as in the case of modifying phrases or clauses, mentioned in Section 4.2.4. Example (26) illustrates a restrictive appositive, whereas (27) and (28) show nonrestrictive ones:

(26) Alexander *the Great* was a powerful emperor.

versus

- (27) With stocks having been battered lately because of the collapse of takeover offers for UAL, *the parent company of United Airlines*, and AMR, *the parent of American Airlines*, analysts viewed the proposal as a psychological lift for the market. (from PT)
- (28) The new company, called Stardent Computer Inc., also said it named John William Poduska, *former chairman and chief executive of Stellar*, to the posts of president and chief executive. (from PT)

4.2.6 Interrupters

Commas are also used to delimit interrupters, which occur sentence-internally as a complementary or parenthetical element. This may be a single word (29), a phrase (30) or an entire clause (31), which breaks the expected logical flow of the sentence:

- (29) The Brookings and Urban Institute authors caution, *however*, that most nursing home stays are of comparatively short duration, and reaching the Medicaid level is more likely with an unusually long stay or repeated stays. (from PT)
- (30) The new bacteria recipients of the genes began producing pertussis toxin which, *because of the mutant virulence gene*, was no longer toxic. (from PT)
- (31) Rebuilding that team, *Mr. Lee predicted*, will take another 10 years. (from PT)

4.2.7 Quotations

Direct quotations, indicating or repeating the exact words of the writer or the speaker, respectively, are set off by commas. Example (32) illustrates such a case:

- (32) “*The absurdity of the official rate should seem obvious to everyone,*” the afternoon newspaper *Izvestia* wrote in a brief commentary on the devaluation. (from PT)

It may be argued whether this is a structural use of comma, since its existence depends on another punctuation mark, the quotation mark, rather than on syntactical items like phrases or clauses. However, the existence of a direct quotation usually changes the grammatical structure by causing an inverted sentence. As a result, the comma here becomes an essential structural mark, along with the quotation marks.

Chapter 5

The Corpus

In computational linguistics, a *corpus* is defined as a set of carefully annotated, electronically available real-life texts, as opposed to a *collection*, which consists of raw (unprocessed) material. A corpus is to be produced in an actual context of language use; the texts included never contain artificial linguistic objects produced under laboratory conditions. Corpora are currently viewed as respectable sources of linguistic data [1]. Accordingly, corpus-based research [3, 4, 18, 19, 20, 21, 22] is becoming increasingly popular within computational linguistics, since it has been realized that valuable progress can be achieved in natural language understanding by investigating naturally occurring text and by automatically analyzing large¹ corpora. A large corpus has the advantage of covering a wide range of real language and minimizing the effect of any errors and inconsistencies introduced by editors, parsers or transcribers.

Design principles of corpora are determined by the research intentions. Some sort of annotation is added to most corpora, for example, to increase the information that a corpus contains. The annotations are usually syntactic in nature and are distinguished into two types: tagging and parsing. Tagging adds atomic

¹The average size of corpora has increased over the years and the definition of a 'large' corpus has changed drastically. Facilities to make texts machine-readable have greatly reduced the amount of money and effort to compile a corpus. The size of the 'legendary' Brown Corpus [13], for example, was a few million words, whereas the British National Corpus [5], which has been recently completed, contains about 100 million words.

and paradigmatic information to each word. while parsing includes the addition of structural information as well as information about larger units than the word form. Tagging minimally assigns the word to one of the major parts of speech (POS) (or word classes), although it may also add more specific information about the subclass to which the word belongs. Tagging a corpus can be done automatically in a short time and with a quite high degree of accuracy, although some manual post-editing is still necessary to reach 100% correctness. Therefore, there is no shortage of tagged corpora. Parsing, on the other hand, is a complex and laborious process, and requires in many cases manual work in terms of pre-editing, post-editing and intervention. Accordingly, the number of ‘fully parsed’ corpora is limited and their size is small, typically between 100,000 and 150,000 words. A feasible way of reducing the production time for a parsed corpus is to lessen the detail of the syntactic analysis. This technique is known as *skeleton parsing* and requires a significant amount of manual work [1].

Due to the nature of our study, there was also need for a corpus. A suitable source for the observation of structural uses of the comma in real-life texts would be a parsed corpus, since ‘structural’ commas set off syntactical boundaries and depend on the grammatical structure of the sentence. Therefore, we have chosen the parsed version of the Penn Treebank [27], which was produced using the skeleton parsing technique [1] mentioned above.

5.1 The Penn Treebank

The Penn Treebank, which is a 4.5 million word corpus of American English, was constructed by Marcus et al. [27] between 1989–1992. Their decision was to produce two types of corpora, which were differently annotated, to serve as a data source for different potential purposes of corpus-based studies. These two types were a POS-tagged corpus and a parsed corpus. Table 5.1 shows the output of the Penn Treebank project as of the end of 1992.² The part of the corpus used in our study is a 309,362 word (14,829 sentences) portion of the preliminary, parsed version (released in April '91) of the Dow Jones section

²The *Dow Jones Newswire stories* section was not parsed on the whole.

Description	POS-tagged (tokens)	Parsed (tokens)
Dept. of Energy abstracts	231,404	231,404
<i>Dow Jones Newswire stories</i>	3,065,776	1,061,166
Dept. of Agriculture bulletins	78,555	78,555
Library of America texts	105,652	105,652
MUC-3 messages	111,828	111,828
IBM Manual sentences	89,121	89,121
WBUR radio transcripts	11,589	11,589
ATIS sentences	19,832	19,832
Brown Corpus, retagged	1,172,041	1,172,041
Total	4,885,798	2,881,188

Table 5.1: Contents of the Penn Treebank (as of end of 1992, adapted from [27, p. 18])

(typeset in italics in Table 5.1), which consists of *Wall Street Journal* articles and is available as part of the first ACL/Data Collection Initiative CD-ROM.³ As it can be seen in Appendix A.1, the sentences included in this particular piece of corpus are usually long and complex, which in turn means that they are also rich in punctuation.

The tagged corpus consists of text with POS-tags attached to words with a slash [/], as in example (33) taken from the corpus. Punctuation marks are tagged with themselves. (See Appendix A.2 for more samples from the tagged corpus.)

(33)
 Scott/NNP C./NNP Smith/NNP ,/, formerly/RB vice/NN
 president/NN ,/, finance/NN ,/, and/CC chief/JJ financial/JJ
 officer/NN of/IN this/DT media/NNS concern/NN ,/, was/VBD
 named/VBN senior/JJ vice/NN president/NN ./ Mr./NNP
 Smith/NNP ,/, 39/CD ,/, retains/VBZ the/DT title/NN of/IN
 chief/JJ financial/JJ officer/NN ./.

The tagging was done in a two-stage process, in which first the POS-tags were automatically assigned by a stochastic algorithm called PARTS [6], and

³Available from Linguistic Data Consortium (<http://www.ldc.upenn.edu/>).

then manually corrected by human annotators. Since our main concern is with the parsed version of the corpus, details of the tagging process are not explained in this thesis.⁴

5.2 Structure of the Parsed Corpus

The parsed version of the Penn Treebank consists of parsed sentences, which show the skeletal structure of the text. The construction procedure of this version is completely parallel to the tagging process. A deterministic parser, called Fidditch [15], was employed to initially parse the material automatically using the tagged version as input. The output of this parser was first simplified, and then manually corrected by human annotators. The advantageous properties of Fidditch are listed by Marcus et al. [27] as follows:

- Fidditch produces exactly one parse for any given sentence, so that annotators are not confused with multiple analyses.
- In cases it is unsure of the role of certain grammatical structures, it outputs a string of trees, indicating the partial structure of the sentence.
- It has a reasonably good grammatical coverage, so that it usually produces quite accurate grammatical chunks.

Due to these advantages, the annotators only had to ‘glue’ together the syntactic trees produced by Fidditch, instead of rebracketing the sentences from scratch. All parsed materials were corrected once.

The final appearance of a parsed sentence is in a bracketed, LISP-like [44] structure as in example (34), which is the bracketed form of the second sentence of example (33). (See Appendix A.3 for more examples from the parsed corpus in LISP-like format.)

⁴See [27] for these details and [38] for POS-tagging guidelines for the Penn Treebank project.

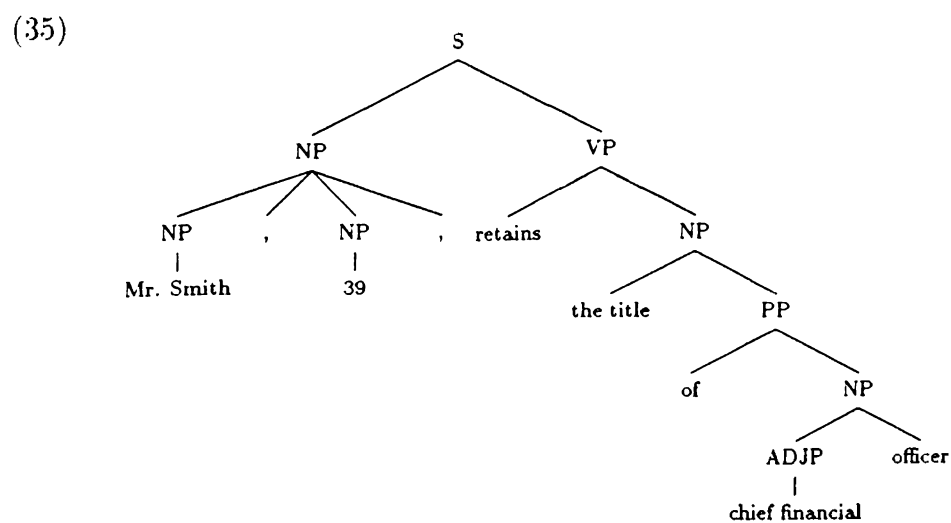
```

(34) ((S
      (NP (NP Mr. Smith)
           ,
           (NP 39)
           ,)
      (VP retains
           (NP the title
              (PP of
                 (NP
                    (ADJP chief financial)
                    officer))))))
      .)

```

Bracketing groups words into phrases and/or clauses, and represents the hierarchical relationship which exist among these constructs. Left brackets are labeled with the type of construct they enclose. The types of constructs available in the syntactic tag-set of the Penn Treebank are listed in Table 5.2.

An alternative representation for the bracketed sentence in example (34) may be a tree diagram:



In such a representation, internal nodes are nonterminal terms belonging to the syntactic tag-set, indicating the type of construct of the subtrees of which they

Tag	Description
1. ADJP	Adjective Phrase
2. ADVP	Adverb Phrase
3. NP	Noun Phrase
4. PP	Prepositional Phrase
5. S	Simple declarative clause
6. SBAR	Clause introduced by subordinating conjunction
7. SBARQ	Direct question introduced by <i>wh</i> -word or <i>wh</i> -phrase
8. SINV	Declarative sentence with subject-aux inversion
9. SQ	Subconstituent of SBARQ excluding <i>wh</i> -word or <i>wh</i> -phrase
10. VP	Verb phrase
11. WHADVP	<i>Wh</i> -adverb phrase
12. WHNP	<i>Wh</i> -noun phrase
13. WHPP	<i>Wh</i> -prepositional phrase
14. X	Constituent of unknown or uncertain category
Null Elements	
1. *	'Understood' subject of infinitive or imperative
2. 0	Zero variant of <i>that</i> in subordinate clauses
3. T	Trace—marks position where moved <i>wh</i> -constituent is interpreted
4. NIL	Marks position where preposition is interpreted in pied-piping contexts

Table 5.2: Syntactic Tag-set of the Penn Treebank (adapted from [27, p. 10])

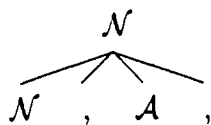
are parents. Terminal terms, the ordinary words of the sentence, are only to be found at external (or leaf) nodes. Terminals and nonterminals that belong together are the children of their common parent node. Clearly, tree diagrams and bracketed labels are equivalent ways of representing syntactic structure.

Detailed guidelines for the bracketed version of the Penn Treebank are explained in [39], where a long list of problematic constructions and conventions (followed to represent them) are given. Section 5.3 gives the constructs related to comma, extracted from this list.

5.3 Constructs Related to Comma

5.3.1 Appositions

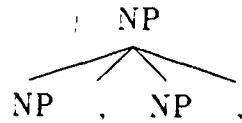
An apposition is defined to be a relation between a nucleus phrase and an appositive phrase, which modifies the nucleus phrase and is usually set off by commas. Here, the nucleus phrase and appositive phrase are rather general terms for every kind of phrase (e.g., NP, VP, PP, etc.) and clause (S, SBAR, etc.), as opposed to an appositive, which was only defined as a noun phrase modifier in Section 4.2.5. Constructs involving appositions are represented as adjunction structures. In other words, the nucleus phrase and the appositive phrase, labeled with their appropriate categories, are the children of the entire apposition structure, which is labeled with the same label as the nucleus phrase. The following is a general tree diagram for an apposition structure, where the syntactic categories of the nucleus phrase and the appositive phrase are labeled symbolically as \mathcal{N} and \mathcal{A} , respectively.



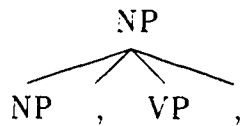
Commas are the siblings of the appositive phrase \mathcal{A} they enclose.

Examples (36)–(40), written in an abbreviated form, are cases for apposition structures with different nucleus and appositive phrases:

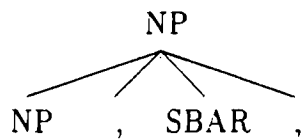
- (36) *Honey, a delicious and nutritious food, is produced by bees.* (from [11, p. 31])



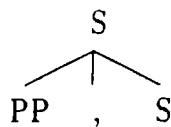
- (37) *The document, written in plain language, clearly applies to real situations.*



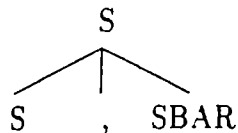
- (38) *Infectious diseases, which often spread because of poor sanitation, are not easily controlled in countries that do not have adequate medical facilities.* (from [11, p. 28])



- (39) *Finally, the family found a way to support all their relatives until economic recovery enabled them to stand on their own.* (from [11, p. 26])



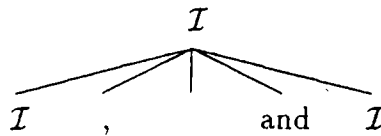
- (40) *Smoking is considered bad for the health, although pipe smoking is said to be less harmful than cigarette smoking.* (from [11, p. 33])



5.3.2 Coordinations

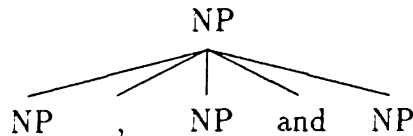
In a sentence, words, phrases or clauses may be coordinated with one another, constituting a list of items. In this case, the coordinated items are labeled with the appropriate tag of the entire list. Commas are the siblings of the items in the list.

The general form of such a coordination structure may be represented as follows, where the items in the list and the list itself are both labeled with \mathcal{I} .

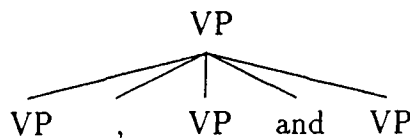


Abbreviated examples for typical coordination structures are given in (41)–(43) for different types of coordinated items:

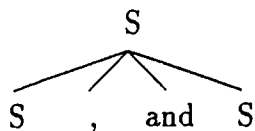
- (41) The size and effectiveness of your vocabulary affect your *writing, speaking and reading* throughout your life. (from [11, p. 15])



- (42) Daisy was able to *find her food, eat it and hide the empty dish*. (from [11, p. 16])



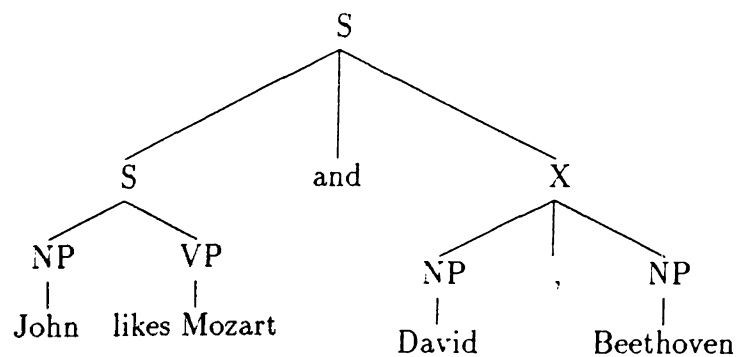
- (43) *With his remaining strength Larry fought the powerful sailor, and a military policeman who was passing by came finally to his aid*. (from [11, p. 18])



5.3.3 Gapping

Gapping is defined as a form of coordination, where the coordinated constructs after the first one are incomplete. In such a case, a comma is put in place of the missing word as in example (44):

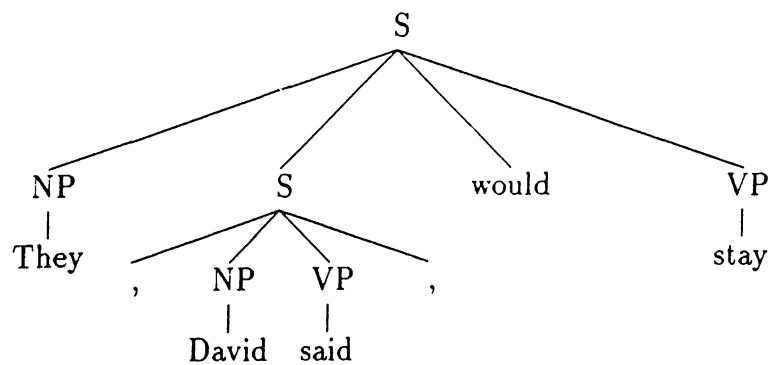
(44) John likes Mozart and David, Beethoven.



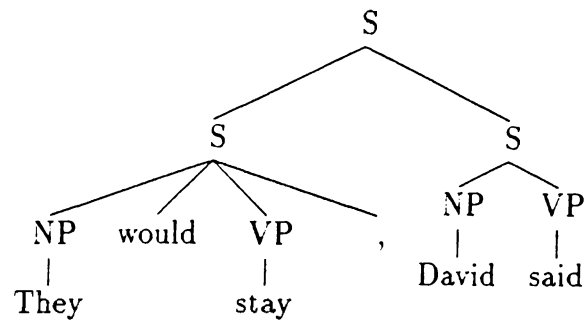
5.3.4 Verbs of Saying

Sentences with verbs of saying may require the use of comma if the order of their constituents diverges from the standard Subject-Verb-Object order as in examples (45)–(47):

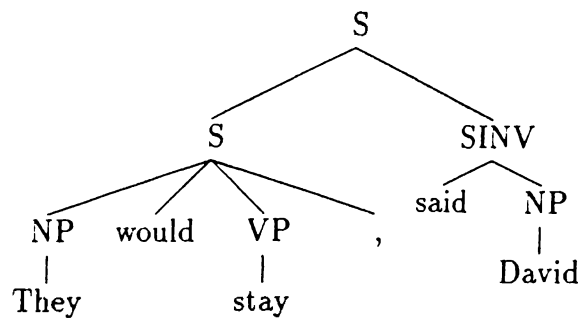
(45) They, *David said*, would stay.



(46) They would stay, *David said*.



(47) They would stay, *said David*.



5.4 Problems with the Corpus

While processing the corpus, many problems have been encountered. Since the corpus had been corrected by human annotators, it was not surprising that it contained errors, both syntactic and semantic, and inconsistencies. In order to obtain reasonable results in the analysis of the corpus, it was important for us to get rid of at least the syntactic errors and some of the semantic errors so that the corpus would become maintainable. This was the most laborious task of our study and it lasted a long time (a couple of months). The encountered problems and their employed or suggested solutions are listed below.

- The most surprising errors were the syntactical ones. There were many instances of sentences diverging from the declared syntax of the parsed

corpus. One of the most frequent syntactic errors was that of unbalanced (missing or spurious) brackets. The solution was to detect these errors using little programs written in the UNIX⁵ utility `awk` [2], and then manually correct them, obviously a time consuming process.

- The next problem was the existence of semantic errors. For example, at some places in the corpus, new syntactic categories (e.g., POSS, PNP, AUX, etc.), which were not listed in the declared syntactic tag-set (Table 5.2), were suddenly introduced. The only possible solution was to detect these automatically and then remove manually, if the error was of a sort that can influence the results of the analysis.
- The major semantic problem with the corpus was the lack of good theoretical rules for placing punctuation marks into appropriate places in the bracketings. This caused enormous inconsistencies within the corpus. Similar syntax-patterns were parsed differently and different patterns were parsed similarly, making the classification of the uses of comma according to these patterns much more difficult. Due to the size of the corpus, it was nearly impossible to detect and get rid of these inconsistencies. So, they affected the results to a noticeable extent, as we will see in Chapter 6.

Marcus et al. [27] admit the existence of a variety of inconsistencies in the annotation scheme they used within the Treebank. They promise to remove these in the future releases of the corpus. However, they are silent about the syntactic errors, which normally should not occur in a commercially distributed corpus.

⁵UNIX is a registered trademark of AT&T.

Chapter 6

The Experiment

In this chapter, the guidelines of the experiment to achieve a proper classification of the commas in the selected piece of Penn Treebank corpus are reported along with the results obtained. Details of the implementation are given in Section 6.1. This is followed by Sections 6.2 and 6.3, where the classification process and its outcome are explained and discussed in detail. Section 6.4 summarizes the results of an analysis which was performed to verify the validity of the classification.

6.1 Implementation

Every study based on an annotated corpus has to make use of the annotation scheme, since there is no other information. For the purposes of our study, a parsed corpus was selected, which consisted of sentences, each parsed into its grammatical constituents. The aim was to make use of these constituents, surrounding or appearing with the commas in the corpus, in order to classify the uses of comma. Since it is said that the major function of the structural comma is setting off syntactic boundaries, the information contained in the parse trees should be enough to make the classification. The first thing to do was to make the corpus easier to process for the language of implementation. The next step was the construction of a database of all syntax-patterns containing one or more

commas, by means of a Prolog [8] implementation. Finally, the classification could be made by assigning these syntax-patterns into appropriate classes.

6.1.1 Preprocessing the Corpus

As it was already stated in Section 5.2, the parsed version of the Penn Treebank had been constructed in a bracketed, LISP-like format, which may be easy to process in LISP, which in turn is not the case for Prolog.

The easiest way of reading input into Prolog is in the format of a so called *Prolog input term*,¹ which is defined as follows:

- Every Prolog input term is an *atom* or a *functor* followed by a period.
- An atom is a nonempty string consisting of alphanumeric characters or the underscore [`_`] character, but it may not have an uppercase letter or an underscore as its first character.
Examples: `atom`, `aTOM`, `a_tom`, etc.
- A functor consists of a *head* and a nonempty list of *arguments*, which are in the following format:

$$\textit{head}(\textit{arg}_1, \textit{arg}_2, \dots, \textit{arg}_n)$$

- The head of a functor has the same restrictions with an atom and the arguments may be of any data type, viz. *variables*, atoms or functors.
- A variable is a nonempty string consisting of alphanumeric characters or the underscore character, but it must have an uppercase letter or an underscore as its first character.
Examples: `Variable`, `VARIABLE`, `_variable`, etc.

A conventional way of representing trees in Prolog is in terms of nested functors: the parent (or root) node is assigned to the head of the functor, the children being its arguments. The format is as follows:

¹The alternative is to read character by character, but this is a rather laborious and inefficient way.

$$\text{parent}(\text{child}_1, \text{child}_2, \dots, \text{child}_n)$$

A child may be another functor if it is a nonterminal (another subtree), or an atom if it is a terminal. The result is the representation of the whole tree, as the one in (48), by a nested functor, as in (49):



(49) $\text{a}(\text{b}(\text{d}, \text{e}), \text{c}(\text{f}, \text{g}))$

In order to render the corpus by Prolog, it was convenient to convert it into the Prolog input format, following the rules and guidelines given above. This conversion was done by means of a program written in `awk` [2]. This program (see Appendix D for the source code) was run through the whole corpus, in order to transform it into Prolog format. The program simply applied the following rules:

1. Transfer all syntactic category labels from the right to the left of the round brackets.
2. Attach the letter `t` (meaning tag) in front of each category label in order to make it a valid functor head. (Heads, as atoms, cannot start with an uppercase letter.)
3. Enclose all terminals, i.e., words and punctuation marks, with double quotation marks ("`...`"), in order to make them Prolog strings.
4. Label all punctuation marks with `tPUNC()`, in order to make them distinguishable from the punctuation marks needed by the syntax of Prolog (for instance, `[,]`, `[()]` or `[.]`).
5. Put a comma at the end of each line, except the last line of the bracketed sentence, in order to make it a functor argument.

6. Put a period at the end of the last line of the bracketed sentence, to mark the end of the nested functor as a Prolog input term.

After this preprocessing stage, a sentence from the original corpus, such as example (50), is transformed into the form in (51). (See Appendix A.4 for more transformed examples.)

```
(50)
((S
  (NP (NP Bell)
    ,
    (VP based
      (PP in
        (NP Los Angeles)))
    ,)
  (VP (VP makes)
    and
    (VP distributes)
    (NP
      (ADJP electronic)
    ,
    computer and building products)))
.)
```

```
(51)
textUnit( tS(
  tNP( tNP( "Bell"),
    tPUNC(", "),
    tVP( "based",
      tPP( "in",
        tNP( "Los Angeles"))),
    tPUNC(", "),
  tVP( tVP( "makes"),
    "and",
```

```

tVP( "distributes"),
    tNP(
        tADJP( "electronic"),
        tPUNC(", "),
        "computer and building products"))),
tPUNC(".").

```

6.1.2 Construction of the Syntax-pattern Database

Having transformed the whole corpus into Prolog format, the next step was the construction of a database of all syntax-patterns containing one or more commas, by means of a Prolog program,² which analyzed all parse trees in the corpus and extracted each node with one or more commas as its immediate daughter(s), with the other daughters abbreviated to their syntactic category labels as in (52)–(56):

(52) (NP (NP My uncle) , (NP 39 years old) .)) \implies NP \rightarrow NP , NP ,

(53) (S (PP In London) , (S ...)) \implies S \rightarrow PP , S

(54) (NP (NP Amsterdam) , (NP Brussels) and (NP London))
 \implies NP \rightarrow NP , NP and NP

(55) (S (SBAR ...) , (S ...)) \implies S \rightarrow SBAR , S

(56) (S Ultimately , (S ...) , (S ...)) \implies S \rightarrow *** , S , S

The three consecutive asterisks (***) mean any number of successive terminal words, not labeled further with any syntactic tag.

Each entry in the constructed database was recorded with the following fields:

²See Appendix E for information about the source code.

- **Pattern** (primary key): The abbreviated syntax-pattern in question.
- **Count**: Number of occurrences of this syntax-pattern in the whole corpus.
- **SampleSentence**: The first sentence, in which the syntax-pattern occurred, recorded in raw text format.

The outcome of this process was a database consisting of 2,156 unique
(**Pattern**, **Count**, **SampleSentence**)

triples. A general description of the implementation used to accomplish this is given below:

- For every sentence in the corpus do...
 - For every comma in the current sentence do...
 - * Construct the abbreviated syntax-pattern for the comma in question.
 - * If this syntax-pattern is already in the database, then increment its **Count** by 1.
 - * If the pattern does not exist in the database, record
(**Pattern**, 1, **SampleSentence**)
as a new database entry, where **SampleSentence** is the current sentence, in raw format.

6.2 Classification of the Syntax-patterns

The aim of the construction of a database of syntax-patterns was to use them in the classification of the uses of the comma in the corpus. This could only be done manually and classifying all 2,156 unique syntax-patterns would be a tremendous task. So, we decided to classify only the most important patterns, such that, at the end, effectively 79% of all commas in the corpus would have been classified. This data would be sufficiently representative for the uses of the comma on the whole.

To determine the most important (i.e., the most frequent) syntax-patterns, the database was sorted according to the number of occurrences in the corpus, in descending order. (See Appendix B for the outcome of this sorting.) Starting from the top of this list, the number of occurrences were added cumulatively until the sum yielded 14,139, which is $\approx 79\%$ of the 17,911 commas in the corpus. The number of the syntax-patterns until this point was recorded as 241, which is only $\approx 11\%$ of the 2,156 unique syntax-patterns for the comma. In other words, it turned out that it was enough to classify the top 11% of the syntax-patterns in order to have effectively classified 79% of the commas in the whole corpus. This is a remarkable result.

The last task to be accomplished was to assign each of the top 241 syntax-patterns to one of the classes listed in Section 4.2. These assignments were done via a simple user interface displaying each time the syntax-pattern and the recorded sample sentence. We had to read the sentence, find the comma in question, and then intuitively select from the menu of classes the one that the use of the comma in question matches, which in turn is the one that the syntax-pattern has to be assigned to. In this way, all of the top 241 syntax-patterns were assigned to a class. Below is a list of all classes along with the top most syntax-pattern recorded for this class, its total number of occurrences, and the sample sentence (with the underlined comma(s)). (See Appendix C.1 for a complete list.)

1. Elements in a series:

Pattern: NP --> NP , NP and NP

Count: 223

SampleSentence:

Currently, the rules force executives, directors and other corporate insiders to report purchases and sales of their companies' shares within about a month after the transaction.

2. Sentence-initial elements:

Pattern: S --> PP , S

Count: 886

SampleSentence:

In an Oct. 19 review of "The Misanthrope" at Chicago's Goodman Theatre ("Revitalized Classics Take the Stage in Windy City," Leisure & Arts), the role of Celimene, played by Kim Cattrall, was mistakenly attributed to Christina Haag.

3. Sentence-final elements:

Pattern: S --> S , SBAR

Count: 101

SampleSentence:

Solo woodwind players have to be creative if they want to work a lot, because their repertoire and audience appeal are limited.

4. Nonrestrictive phrases or clauses:

Pattern: NP --> NP , SBAR

Count: 505

SampleSentence:

The changes were proposed in an effort to streamline federal bureaucracy and boost compliance by the executives "who are really calling the shots," said Brian Lane, special counsel at the SEC's office of disclosure policy, which proposed the changes.

5. Appositives:

Pattern: NP --> NP , NP ,

Count: 1812

SampleSentence:

Howard Mosher, president and chief executive officer, said he anticipates growth for the luxury auto maker in Britain and Europe, and in Far Eastern markets.

6. Interrupters:

Pattern: S --> NP , PP , VP

Count: 80

SampleSentence:

The U.S., along with Britain and Singapore, left the agency when its anti-Western ideology, financial corruption and top leadership got out of hand.

7. Quotations:

Pattern: S --> ‘ ‘ NP VP , ’ ’

Count: 176

SampleSentence:

In an Oct. 19 review of ‘ ‘The Misanthrope’ ’ at Chicago’s Goodman Theatre (‘ ‘Revitalized Classics Take the Stage in Windy City,’ ’ Leisure & Arts), the role of Celimene, played by Kim Cattrall, was mistakenly attributed to Christina Haag.

6.3 Results of the Classification

A summary of the results of the classification is given in Table 6.1. The first column contains the general class and the second column, the more specific subclasses of this general class. The next two columns display the number of occurrences of the class or subclass and the percentage of this number to the whole number of effectively classified commas (14,139 \approx 79% of 17,911), respectively. Column 5 shows the number of unique syntax-patterns assigned to the class or subclass, and column 6 includes the percentage of this number to the whole number of classified patterns (241 \approx 11% of 2,156). The last column contains the proportion of the number of commas to the number of patterns for the particular class or subclass, which we call the *stability* of that class or subclass. The stability of a class is calculated as follows:

$$stability = \frac{\text{number of commas}}{\text{number of patterns}}$$

Class	Subclass	#Commas	%Commas	#Patterns	%Patterns	Stability
Elements in a Series	Words in series	288	2.0%	5	2.1%	58
	Phrases in series	1113	7.9%	34	14.1%	33
	Clauses in series	114	0.8%	8	3.3%	14
	Coordinate clauses	1041	7.4%	13	5.4%	80
	Coordinate adjectives	143	1.0%	4	1.7%	36
	TOTAL	2699	19.1%	64	26.6%	42
Sentence- initial Elements	Introductory words	450	3.2%	5	2.1%	90
	Introductory phrases	1865	13.2%	22	9.1%	85
	Introductory clauses	617	4.4%	8	3.3%	77
	TOTAL	2932	20.7%	35	14.5%	84
Sentence- final Elements	Final phrases	455	3.2%	22	9.1%	21
	Final clauses	239	1.7%	9	3.7%	27
	Absolute phrases	41	0.3%	2	0.8%	20
	TOTAL	735	5.2%	33	13.7%	22
Nonrestrictive Phrases or Clauses	Nonr. phrases	945	6.7%	17	7.1%	56
	Nonr. clauses	1472	10.4%	14	5.8%	105
	TOTAL	2417	17.1%	31	12.9%	78
Appositives	TOTAL	3766	26.6%	22	9.1%	171
Interrupters	TOTAL	929	6.6%	39	16.2%	23
Quotations	TOTAL	661	4.7%	17	7.1%	39
TOTAL	TOTAL	14139	100%	241	100%	59

Table 6.1: Results of the Classification

According to Table 6.1, the most frequent use of comma in the corpus is the setting off of appositives, which is followed by sentence-initial elements and elements in a series, with sentence-final elements and quotations at the end. The most frequent elements listed in a series are phrases followed by coordinate clauses. Phrases are also dominantly set off by commas as sentence-initial and sentence-final elements. Finally, nonrestrictive clauses delimited by commas were approximately 50% more than nonrestrictive phrases.

The interpretation of the above statistics are left to experts in linguistics. However, we can at least guess that these results would be very different if we had analyzed a corpus from another source of origin than the *Wall Street Journal*.

The stability measure of a class or subclass, introduced above, requires an explanation. This number shows the average number of commas per syntax-pattern assigned to a class or subclass, which is also a sign of the variety of these patterns for the class in question. The more the number of commas per pattern means the less variety of patterns; i.e., the more stable is the class in question.

The most stable class of the use of the comma appeared to be the commas setting off appositives. This is followed by the commas delimiting sentence-initial elements, and nonrestrictive phrases or clauses. Conversely, the most versatile classes turned out to be the commas setting of interrupters and sentence-final elements, meaning that the syntax-patterns occurring in these classes are less standardized. The stability of a class shows also the capability of its individual syntax-patterns to be reduced to more general rule-patterns. On the other hand, the calculated stability of the whole corpus, approximately 59, may be viewed as an indicator of the precision and consistency of the parsing and correction procedure applied on that particular corpus. Since this experiment was done with a single corpus, we cannot compare this parameter with the stabilities of other corpora.

Appendix C.2 contains the list of the 241 classified syntax-patterns, this time sorted according to the patterns themselves, which brought similar patterns together within a class. This list may be another clue-giving source for the

stability of individual classes and the whole corpus.

6.4 Verification of the Classification

The uses of the commas in the corpus were intuitively classified by means of the syntax-patterns in which they occur, each time looking at exactly one sample sentence for each pattern. In other words, it was assumed that all commas appearing in the same syntax-pattern fall into the same class of use, regardless of the sentence in which they occur. This assumption, however, needs to be verified for its degree of validity. For example, two different uses of comma may have resulted in the same syntax-pattern during the parsing process.

A reasonable way of doing this verification was looking at randomly selected sentences in which the classified syntax-patterns occur. Accordingly, a random collection of 133 ($\approx 1\%$ of the whole corpus) sentences was extracted from the corpus and examined manually for erroneous classifications. It has been taken care that these sentences were not those that were used in the classification. The result was more than satisfactory, with 95% of the sentences selected showing a correct classification of their commas. The 5% error was completely due to the idiosyncrasies or the imprecision of the parsing or the correction procedure (see Section 5.4) applied to the corpus. For instance, the syntax-patterns of dates (such as *May 10, 1996*) or addresses (such as *Los Angeles, California*) both appeared to be

NP --> NP , NP

which was classified as 'Appositives'. This is obviously wrong since in both cases the constructs set off by commas are not appositives.

As a result, we can talk about nearly 100% validity of the assumption that a parsed corpus is a suitable input material to be used for the classification of the uses of the structural comma. In other words, syntax-patterns appear to be distinguishing measures for this purpose.

Chapter 7

Conclusion

The research explained in this thesis was guided by the hypothesis that abbreviated syntax-patterns derived from a parsed corpus should be capable of and sufficient for determining the rich variety of uses of the comma. To this end, the parsed version of the Penn Treebank corpus was analyzed. Especially the results of the experiment done for the verification of the classification based on the syntax-patterns show that the assumption seems to be valid. (Clearly, there is also a need for a theoretical validation.)

The corpus contained material from a single type of origin: the *Wall Street Journal*, which is a respected business paper published in the strictest journalistic style. Therefore, this study could be extended by investigating other corpora, containing material from other types of journals or other domains of literacy such as fiction or learned writing, in which the punctuation practice might display variety. It is not difficult to guess that, in such a case, the frequencies given in Section 6.3 for the various uses of comma would change and new classes of uses could appear. However, the above hypothesis would probably not lose its validity.

A disadvantage of Penn Treebank was the existence of syntactical and semantical errors and inconsistencies on the part of the parser (and the annotators who corrected it). In order to obtain better results, this experiment could be repeated with other corpora, such as the last release of the Penn

Trebank [26], which is announced to be more consistent and error free. The notion of *stability*, which was introduced in this thesis, could be a useful number to measure the consistency of a parsed corpus.

Although the most significant punctuation mark is the comma, other structural marks, especially the period, colon, semicolon and dash, also deserve investigation. The experience obtained during our work could profitably guide such a study.

In this work, effectively only the 79% of all commas in the corpus were classified according to their uses. This number could be extended to 90%, or even 100%. In this case, the percentage of the abbreviated syntax-patterns that need to be investigated would rise from 11% to 36% and 100%. As a solution, the unique syntax-patterns could be first reduced to more general rule-patterns, as Jones [22] did. These rule-patterns could then be easily assigned to appropriate classes of uses of the comma. Furthermore, the generality and coverage of such rule-patterns could be helpful in the determination of the class of a newly encountered syntax-pattern. In fact, with the development of parsers with nearly 100% coverage in the near future, it will be possible to have punctuation checkers—along with grammar and spell checkers—which will ascertain the correctness or the consistency of the punctuation practice in a given text, according to a specific style. Such a tool would make use of a database of rule-patterns derived from past experience.

Although our work focused only on English punctuation practice for the special case of comma, the problem is universal. Therefore, a similar study could be done on Turkish, besides other languages. Turkish [46] is an agglutinative and free word-order language as opposed to English. Accordingly, there are some differences in punctuation practice. For instance, a long and/or complex subject is always set off by a comma—something which is never done in English—in order to prevent ambiguity. On the other hand, there is less consensus on the punctuation rules in Turkish. After all, English has at least a 400 year tradition on punctuation as opposed to Turkish, in which modern punctuation marks began to be used with the introduction of the Latin alphabet only about 70 years ago.

Bibliography

- [1] Jan Aarts. Corpus analysis. In Jef Verschueren, Jan-Ola Östman, and Jan Blommaert, editors, *Handbook of Pragmatics*, pages 565–570. John Benjamin Publishing Company, Amsterdam, the Netherlands, 1995.
- [2] Alfred W. Aho, Brian W. Kernighan, and Peter J. Weinberger. *The AWK Programming Language*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1988.
- [3] Ted Briscoe. Parsing (with) punctuation. Rank Xerox Research Centre, Grenoble, France, 1994.
- [4] Ted Briscoe and John Carroll. Developing and evaluating a probabilistic LR parser of part-of-speech and punctuation labels. In *Proceedings of International Workshop on Parsing Technologies*, pages 48–58, Prague, Czech Republic, 1995.
- [5] G. Burnage and D. Dunlop. Encoding the British National Corpus. In J. Aarts, P. de Haan, and N. Oostdijk, editors, *English Language Corpora*, pages 79–95. Rodopi, Amsterdam, the Netherlands, 1993.
- [6] Kenneth W. Church. A stochastic PARTS program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing. 26th Annual Meeting of the Association for Computational Linguistics*, pages 136–143, Austin, Texas, 1988.
- [7] John W. Clark. A chapter on American practice. In Eric Partridge, editor, *You have a Point There: A Guide to Punctuation and its Allies*, pages 211–221, London, 1953. Hamish Hamilton Ltd. Reprinted by Routledge in 1993.

- [8] W. F. Clocksin and C. S. Mellish. *Programming in Prolog*. Springer-Verlag, New York, 3rd edition, 1987.
- [9] Robert Dale. Exploring the role of punctuation in the signalling of discourse structure. In *Proceedings of a Workshop on Text Representation and Domain Modelling: Ideas from Linguistics and AI*, pages 110–120. Technical University of Berlin, 1991.
- [10] Robert Dale. The role of punctuation in discourse structure. In *Working Notes for the AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation*, pages 13–14, Asilomar, California, 1991.
- [11] Eugene Ehrlich. *Theory and Problems of Punctuation. Capitalization, and Spelling*. Schaum's Outline Series. McGraw-Hill, Hong Kong, 2nd edition, 1992.
- [12] Claude W. Faulkner. *Writing Good Sentences: A Functional Approach to Sentence, Grammar and Punctuation*. Charles Scribner's Sons, New York, 1981.
- [13] W. N. Francis and H. Kučera. *Frequency Analysis of English Usage: Lexicon and Grammar*. Houghton Mifflin, Boston, Massachusetts, 1982.
- [14] Roger Garside, Geoffrey Leech, and Geoffrey Sampson, editors. *The Computational Analysis of English*. Longman, London, 1987.
- [15] Donald Hindle. *User Manual for Fidditch, a Deterministic Parser*. Naval Research Laboratory, Technical memorandum 7590-142, Washington D.C., 1983.
- [16] Lee Humphreys. Book review: The Linguistics of Punctuation. *Machine Translation*, 7:199–201, 1993.
- [17] Gordon Jarvie. *Chambers Punctuation Guide*. Chambers, Edinburgh, UK, 1992.
- [18] Bernard Jones. Can punctuation help parsing? Acquilex-II Working Paper 29, Cambridge University Computer Lab., 1994.

- [19] Bernard Jones. Exploring the role of punctuation in parsing natural language. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, pages 421–425, Kyoto, Japan, 1994.
- [20] Bernard Jones. Exploring the variety and use of punctuation. In *Proceedings of the 17th Annual Cognitive Science Conference*, pages 619–624, Pittsburgh, Pennsylvania, 1995.
- [21] Bernard Jones. Towards a syntactic account of punctuation. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages 604–609, Copenhagen, Denmark, 1996.
- [22] Bernard Jones. Towards testing the syntax of punctuation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 363–365, Santa Cruz, California, 1996.
- [23] Bernard Jones. *What's the point: A (computational) theory of punctuation*. PhD thesis, University of Edinburgh, 1996.
- [24] Hans Kamp and Uwe Reyle. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, Dordrecht, 1993.
- [25] Fred Karlsson, Atro Voutilainen, Juha Heikkilä, and Arto Antilla, editors. *Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text*. Mouton de Gruyter, Berlin, 1994.
- [26] M. Marcus, G. Kim, M.A. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the Human Language Technology Workshop*, pages 27–38, Morgan Kaufmann, San Francisco, California, March 1994.
- [27] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. University of Pennsylvania, 1992.
- [28] John McDermott. *Punctuation for Now*. The MacMillan Press, Hong Kong, 1990.

- [29] Charles F. Meyer. *A Descriptive Study of American Punctuation*. PhD thesis, University of Wisconsin-Milwaukee, 1983.
- [30] Charles F. Meyer. Punctuation practice in the Brown Corpus. *ICAME Newsletter*, pages 80–95. 1986.
- [31] Charles F. Meyer. *A Linguistic Study of American Punctuation*. Peter Lang Publishing Co., 1987. Out of print.
- [32] Geoffrey Nunberg. *The Linguistics of Punctuation*. Number 18 in CSLI Lecture Notes. CSLI Publications, Stanford, California, 1990.
- [33] M. B. Parkes. *Pause and Effect: An Introduction to the History of Punctuation in the West*. University of California Press, Berkeley, California, 1992.
- [34] Eric Partridge. *You have a Point There: A Guide to Punctuation and its Allies*. Hamish Hamilton Ltd., London, 1953. Reprinted by Routledge in 1993.
- [35] William C. Paxson. *The Mentor Guide to Punctuation*. Mentor Books, New York, 1986.
- [36] F. Pereira and D. Warren. Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13(3):231–278, 1980.
- [37] Geoffrey Sampson. *English for the Computer: The SUSANNE Corpus and Analytic Scheme*. Oxford University Press, Oxford, UK, 1995.
- [38] Beatrice Santorini. Part-of-speech tagging guidelines for the Penn Treebank project. University of Pennsylvania, 1990. 3rd revision, 2nd printing.
- [39] Beatrice Santorini. Bracketing guidelines for the Penn Treebank project. University of Pennsylvania, 1991. Draft.
- [40] B. Say and V. Akman. Information-based aspects of punctuation. In *Proceedings of the First International Workshop on Punctuation in Computational Linguistics*, pages 49–56, SIGPARSE '96, Santa Cruz, California, 1996.

- [41] B. Say and V. Akman. An information-based treatment of punctuation. In *Abstracts of the Second Conference on Mathematical Linguistics*, pages 93–94, Tarragona, Spain, 1996.
- [42] Bilge Say. An information-based approach to punctuation. PhD proposal, Bilkent University, Ankara, Turkey, 1995.
- [43] V. Srinivasan. Punctuation and parsing of real-world texts. In *Proceedings of Sixth Twente Workshop on Language Technologies*, Twente, the Netherlands, 1991.
- [44] L. Guy Steele Jr. *Common Lisp: The Language*. Digital Press, Billerica, Massachusetts, 2nd edition, 1990.
- [45] L. J. Taylor and G. Knowles. Manual of information to accompany the SEC Corpus. University of Lancaster, UK, 1988.
- [46] Robert Underhill. *Turkish Grammar*. The MIT Press, Cambridge, Massachusetts, 1976.
- [47] Michael White. Presenting punctuation. In *Proceedings of the Fifth European Workshop on Natural Language Generation*, pages 107–125, Leiden, the Netherlands, 1995.
- [48] *The Encyclopedia Americana International Edition*, volume 23. Grolier Incorporated, Danbury, Connecticut, 1984.
- [49] *The New Encyclopedia Britannica*, volume 15. Encyclopedia Britannica, Inc., Chicago, Illinois, 1977.
- [50] *Collins Cobuild English Language Dictionary*. William Collins & Sons Co. Ltd., London, 1987.
- [51] *Longman Dictionary of Contemporary English*. Longman Group UK Limited, Suffolk, UK, 2nd edition, 1987.
- [52] *The Macmillan Family Encyclopedia*, volume 15. Grolier Incorporated, London, 1993.
- [53] *The New Shorter Oxford English Dictionary*, volume 2. Oxford University Press, New York, 1993.

- [54] *Webster's New Twentieth Century Dictionary of the English Language*. Prentice Hall, New York, 2nd edition, 1983.
- [55] *Webster's II, New Riverside University Dictionary*. Riverside Publishing Co., Boston, Massachusetts, 1994.
- [56] *The World Book Encyclopedia*, volume 15. World Book-Childcraft International, Inc., Chicago, Illinois, 1982.

Appendix A

Samples from the Corpus

A.1 Raw Format

Kemper Financial Services Inc., charging that program trading is ruining the stock market, cut off four big Wall Street firms from doing any of its stock-trading business. The move is the biggest salvo yet in the renewed outcry against program trading, with Kemper putting its money--the millions of dollars in commissions it generates each year where its mouth is. The Kemper Corp. unit and other critics complain that program trading causes wild swings in stock prices, such as on Tuesday and on Oct. 13 and 16, and has increased chances for market crashes. Over the past nine months, several firms, including discount broker Charles Schwab & Co. and Sears, Roebuck & Co.'s Dean Witter Reynolds Inc. unit, have attacked program trading as a major market evil. Several big securities firms backed off from program trading a few months after the 1987 crash. But most of them, led by Morgan Stanley & Co., moved back in earlier this year. The most volatile form of program trading is index arbitrage--the rapid-fire, computer-guided buying and selling of stocks offset with opposite trades in stock-index futures and options. The object is to capture profits from fleeting price discrepancies between the futures and options and the stocks themselves.

A.2 Tagged Format

Kemper/NNP Financial/NNP Services/NPS Inc./NNP ,/, charging/VBG that/DT
program/NN trading/NN is/VBZ ruining/VBG the/DT stock/NN market/NN ,/,
cut/VBD off/RP four/CD big/JJ Wall/NNP Street/NNP firms/NNS from/IN doing/VBG
any/DT of/IN its/PPN\$ stock-trading/NN business/NN ./.. The/DT move/NN is/VBZ
the/DT biggest/JJS salvo/NN yet/RB in/IN the/DT renewed/VBN outcry/NN against/IN
program/NN trading/NN ,/, with/IN Kemper/NNP putting/VBG its/PPN\$ money/NN
--/: the/DT millions/NNS of/IN dollars/NNS in/IN commissions/NNS it/PPN
generates/VBZ each/DT year/NN --/: where/WRB its/PPN\$ mouth/NN is/VBZ ./..
The/DT Kemper/NNP Corp./NNP unit/NN and/CC other/JJ critics/NNS complain/VBP
that/DT program/NN trading/NN causes/VBZ wild/JJ swings/NNS in/IN stock/NN
prices/NNS ,/, such/JJ as/IN on/IN Tuesday/NNP and/CC on/IN Oct./NNP 13/CD
and/CC 16/CD ,/, and/CC has/VBZ increased/VBN chances/NNS for/IN market/NN
crashes/NNS ./.. Over/IN the/DT past/JJ nine/CD months/NNS ,/, several/JJ
firms/NNS ,/, including/VBG discount/NN broker/NN Charles/NNP Schwab/NNP
&/CC Co./NNP and/CC Sears/NNP ,/, Roebuck/NNP &/CC Co./NNP 's/POS Dean/NNP
Witter/NNP Reynolds/NNP Inc./NNP unit/NN ,/, have/VBP attacked/VBN program/NN
trading/NN as/IN a/DT major/JJ market/NN evil/NN ./.. Several/JJ big/JJ
securities/NNS firms/NN backed/VBD off/RB from/IN program/NN trading/NN
a/DT few/JJ months/NNS after/IN the/DT 1987/CD crash/NN ./.. But/CC most/JJS
of/IN them/PPN ,/, led/VBN by/IN Morgan/NNP Stanley/NNP &/CC Co./NNP ,/,
moved/VBD back/RB in/RB earlier/RBR this/DT year/NN ./.. The/DT most/RBS
volatile/JJ form/NN of/IN program/NN trading/NN is/VBZ index/NN arbitrage/NN
--/: the/DT rapid-fire/JJ ,/, computer-guided/JJ buying/NN and/CC selling/NN
of/IN stocks/NNS offset/VBN with/IN opposite/JJ trades/NNS in/IN stock-index/NN
futures/NNS and/CC options/NNS ./.. The/DT object/NN is/VBZ to/TO capture/VB
profits/NNS from/IN fleeting/JJ price/NN discrepancies/NNS between/IN the/DT
futures/NNS and/CC options/NNS and/CC the/DT stocks/NNS themselves/PP ./..

A.3 Parsed in LISP Format

```

(
  (S (NP (NP Kemper Financial Services Inc.)
        ,
        (VP charging
          (SBAR that
            (S (NP program trading)
                is
                (VP ruining
                  (NP the stock market))))))
        ,)
    (VP cut
      (PRT off)
      (NP four big Wall Street firms)
      (PP from
        (S (NP *)
            (VP doing
              (NP any
                (PP of
                  (NP its stock-trading business)))))))
    .)
  ( (S
      (NP The move)
      (VP is
        (NP the
          biggest
          salvo
          yet
          (PP in
            (NP the renewed outcry
              (PP against
                (NP program trading))))))
        ,
        (PP with
          (NP (NP Kemper)
            (VP putting
              (NP (NP its money)
                --
                (SBAR (NP the millions
                  (PP of
                    (NP dollars))
                    (PP in
                      (NP commissions)))
                  (S
                    (NP it)
                    (VP generates
                      (NP T)
                      (NP each year))))
                --)
          (SBAR

```

```

(WHADVP where)
(S (NP its mouth)
  (VP is))))))
.)

((S
  (NP (NP The Kemper Corp. unit)
    and
    (NP other critics))
  (VP complain
    (SBAR that
      (S (NP program trading)
        (VP (VP causes
          (NP wild swings
            (PP in
              (NP stock prices))
            ,
            (ADJP such
              (PP as
                (PP (PP on
                  (NP Tuesday))
                and
                (PP on
                  (NP Oct. 13
                    and
                    16))))))
            ,))
          and
          (S has
            (VP increased
              (NP chances
                (PP for
                  (NP market crashes))))))))))
    .)

(
(S (PP Over
  (NP the
    (ADJP past
      nine)
    months))
  ,
  (S
    (NP (NP several firms)
      ,
      (VP including
        (NP (NP discount broker
          (NP Charles Schwab
            &
            Co.))
          and
          (NP
            (NP

```

```

        Sears
        ,
        Roebuck
        &
        Co.)
        's Dean Witter Reynolds Inc. unit)))
    ,)
have
(VP attacked
  (NP program trading)
  (PP as
    (NP a major market evil))))
.)

(
(S
  (NP
    Several big
    securities
    firms)
  (VP backed
    (PRT off)
    (PP from
      (NP program trading))
    (NP a few months
      (PP after
        (NP the 1987 crash))))))
.)

( (S But
  (S
    (NP (NP (ADJP most)
      (PP of
        (NP them)))
    ,
    (VP led
      (PP by
        (NP Morgan Stanley
          &
          Co.)))
    ,)
    (VP moved
      (ADVP back
        in)
      (ADVP earlier
        (NP this year))))))
.)

( (S
  (NP The
    (ADJP most volatile)
    form
    (PP of

```



```

      (NP program trading)))
(VP is
  (NP (NP index arbitrage)
    --
    (NP (NP the
      rapid-fire
      ,
      computer-guided
      buying and selling
      (PP of
        (NP stocks)))
      (VP offset
        (PP with
          (NP opposite trades
            (PP in
              (NP stock-index futures and options))))))))))
.)

( (S
  (NP The object)
  (VP is
    (S (NP *)
      to
      (VP capture
        (NP profits
          (PP from
            (NP fleeting price discrepancies
              (PP between
                (NP (NP the futures and options)
                  and
                  (NP (NP the stocks)
                    (NP themselves))))))))))
.)

```

A.4 Converted to Prolog Format

```

textUnit(
  tS( tNP( tNP( "Kemper Financial Services Inc."),
    tPUNC(", "),
    tVP( "charging",
      tSBAR( "that",
        tS( tNP( "program trading"),
          "is",
          tVP( "ruining",
            tNP( "the stock market"))))),
    tPUNC(", "),
    tVP( "cut",
      tPRT( "off"),
      tNP( "four big Wall Street firms"),
      tPP( "from",
        tS( tNP( "*"),
          tVP( "doing",
            tNP( "any",
              tPP( "of",
                tNP( "its stock-trading business"))))))),
    tPUNC(".")).

```

```

textUnit( tS(
  tNP( "The move"),
  tVP( "is",
    tNP( "the",
      "biggest",
      "salvo",
      "yet",
      tPP( "in",
        tNP( "the renewed outcry",
          tPP( "against",
            tNP( "program trading"))))),
    tPUNC(", "),
    tPP( "with",
      tNP( tNP( "Kemper"),
        tVP( "putting",
          tNP( tNP( "its money"),
            tPUNC("--"),
            tSBAR( tNP( "the millions",
              tPP( "of",
                tNP( "dollars")),
              tPP( "in",
                tNP( "commissions"))),
          tS(
            tNP( "it"),
            tVP( "generates",
              tNP( "T"),
              tNP( "each year"))),
            tPUNC("--"),
          tSBAR(

```

```

                                tWHADV( "where"),
                                tS( tNP( "its mouth"),
                                    tVP( "is")))))))
tPUNC(".").

textUnit( tS(
    tNP( tNP( "The Kemper Corp. unit"),
        "and",
        tNP( "other critics")),
    tVP( "complain",
        tSBAR( "that",
            tS( tNP( "program trading"),
                tVP( tVP( "causes",
                    tNP( "wild swings",
                        tPP( "in",
                            tNP( "stock prices")),
                        tPUNC(","),
                        tADJP( "such",
                            tPP( "as",
                                tPP( tPP( "on",
                                    tNP( "Tuesday")),
                                    "and",
                                    tPP( "on",
                                        tNP( "Oct. 13",
                                            "and",
                                            "16"))))),
                                tPUNC(",")))))
                    "and",
                    tS( "has",
                        tVP( "increased",
                            tNP( "chances",
                                tPP( "for",
                                    tNP( "market crashes")))))))
                tPUNC(",")))))
    tPUNC(".").

textUnit(
    tS( tPP( "Over",
        tNP( "the",
            tADJP( "past",
                "nine",
                "months")),
        tPUNC(","),
        tS(
            tNP( tNP( "several firms"),
                tPUNC(","),
                tVP( "including",
                    tNP( tNP( "discount broker",
                        tNP( "Charles Schwab",
                            "&",
                            "Co.")),
                        "and",
                        tNP(
                            tNP(

```

```

        "Sears",
        tPUNC(", "),
        "Roebuck",
        "&",
        "Co."),
        "'s Dean Witter Reynolds Inc. unit'))),
    tPUNC(", "),
    "have",
    tVP("attacked",
        tNP("program trading"),
        tPP("as",
            tNP("a major market evil")))),
    tPUNC(". ").

textUnit(
    tS(
        tNP(
            "Several big",
            "securities",
            "firms"),
        tVP("backed",
            tPRT("off"),
            tPP("from",
                tNP("program trading")),
            tNP("a few months",
                tPP("after",
                    tNP("the 1987 crash"))))),
        tPUNC(". ").

textUnit( tS( "But",
    tS(
        tNP( tNP( tADJP( "most"),
            tPP( "of",
                tNP( "them"))),
            tPUNC(", "),
            tVP( "led",
                tPP( "by",
                    tNP( "Morgan Stanley",
                        "&",
                        "Co.))),
            tPUNC(", "),
            tVP( "moved",
                tADVP( "back",
                    "in"),
                tADVP( "earlier",
                    tNP( "this year")))),
            tPUNC(". ").

textUnit( tS(
    tNP( "The",
        tADJP( "most volatile"),
        "form",
        tPP( "of",

```

```

        tNP( "program trading"))),
tVP( "is",
    tNP( tNP( "index arbitrage"),
        tPUNC("--"),
        tNP( tNP( "the",
            "rapid-fire",
            tPUNC(","),
            "computer-guided",
            "buying and selling",
            tPP( "of",
                tNP( "stocks"))),
            tVP( "offset",
                tPP( "with",
                    tNP( "opposite trades",
                        tPP( "in",
                            tNP( "stock-index futures and options")))))))),
tPUNC(".")).

textUnit( tS(
    tNP( "The object"),
    tVP( "is",
        tS( tNP( "*"),
            "to",
            tVP( "capture",
                tNP( "profits",
                    tPP( "from",
                        tNP( "fleeting price discrepancies",
                            tPP( "between",
                                tNP( tNP( "the futures and options"),
                                    "and",
                                    tNP( tNP( "the stocks"),
                                        tNP( "themselves")))))))))),
tPUNC(".")).

```

Appendix B

Sorted List of Syntax-patterns

Count	Syntax-pattern
1812	NP --> NP , NP ,
1252	NP --> NP , NP
886	S --> PP , S
505	NP --> NP , SBAR
369	S --> PP , S .
350	NP --> NP , SBARQ ,
336	NP --> NP , VP ,
297	S --> SBAR , S
283	S --> *** , S
276	S --> S , S
223	NP --> NP , NP and NP
206	NP --> *** , ***
197	S --> S , and S
185	NP --> NP , VP
176	NP --> NP , SBAR ,
176	S --> ' ' NP VP , ' '
146	S --> S , *** S
140	NP --> NP , or NP ,
137	NP --> NP , or NP
134	NP --> NP , NP , NP and NP
129	S --> *** , S .
122	S --> ' ' NP *** VP , ' '
119	S --> SBAR , S .
112	SBAR --> *** S , S

106 ADJP --> *** , ***
 101 S --> S , SBAR
 98 S --> NP , S
 89 S --> S , S .
 84 NP --> NP , NP , and NP
 81 S --> VP , S
 80 S --> NP , PP , VP
 74 S --> S , *** S .
 74 NP --> *** , PP ,
 70 NP --> NP , ADVP
 67 VP --> *** NP , PP
 67 S --> ADVP , S
 66 S --> S , NP VP .
 66 NP --> *** , SBARQ ,
 65 VP --> VP , and VP
 63 NP --> NP , and NP
 61 NP --> *** , SBAR
 61 VP --> *** , S
 59 S --> NP , S .
 59 VP --> *** NP , VP
 56 S --> NP , S , VP
 56 S --> *** , S , S
 56 NP --> NP , PP ,
 55 S --> S , and S .
 55 VP --> *** NP , S
 54 NP --> ' ' *** , ' '
 54 NP --> NP , NP , NP
 52 NP --> NP , *** NP
 52 NP --> NP , ADJP ,
 52 S --> NP , PP , *** VP
 50 VP --> VP , *** VP
 50 NP --> NP , SBARQ
 50 S --> NP , *** , VP
 48 S --> PP , S , S
 48 NP --> NP , NP , NP ,
 47 NP --> NP , ADJP
 46 VP --> *** PP , ADVP
 46 S --> *** PP , S
 45 S --> NP VP , S
 43 S --> ' ' S , ' ' NP VP .
 42 NP --> NP , NP , PP
 41 SINV --> ' ' S , ' ' VP NP .

40 S --> PP , NP VP
 39 S --> S , VP .
 39 S --> VP , S .
 38 NP --> NP , S ,
 38 S --> *** , PP , S
 38 NP --> NP , NP , SBAR
 37 S --> S , SINV
 36 NP --> NP , *** NP ,
 36 NP --> NP , NP , NP , NP and NP
 36 S --> S , VP
 36 S --> NP , VP
 36 NP --> NP , NP , NP , and NP
 34 NP --> *** , SBAR ,
 34 S --> NP , *** VP
 34 S --> ' ' SBAR , S , ' '
 34 VP --> VP , VP and VP
 33 NP --> NP , S
 32 NP --> *** , *** ,
 32 S --> NP , S , *** VP
 31 VP --> *** NP PP , PP
 31 VP --> *** NP , ADVP
 30 S --> NP , *** , VP .
 30 S --> NP , PP , VP .
 30 VP --> *** NP PP , VP
 30 NP --> NP , and NP ,
 29 S --> ' ' NP VP , ' ' SINV
 29 VP --> *** PP , VP
 29 VP --> *** PP , PP
 28 NP --> NP , NP , NP , NP , NP and NP
 28 VP --> *** NP , SBAR
 28 NP --> *** , *** , ***
 28 S --> ' ' S , and S , ' '
 28 NP --> *** , ADJP ,
 28 VP --> *** PP , S
 28 S --> ' ' S , *** S , ' '
 27 NP --> *** , *** and ***
 26 NP --> *** , PP
 26 VP --> *** PP , SBAR
 26 S --> *** SBAR , S
 25 NP --> NP , NP , NP , NP , NP , NP and NP
 24 S --> NP , PP , *** VP .
 24 S --> NP . *** . *** VP

23 NP --> (NP , NP)
 23 NP --> *** PP , PP
 22 VP --> *** NP PP , S
 22 S --> PP , PP , S
 22 NP --> NP , PP
 21 VP --> *** , NP
 20 SINV --> S , VP NP .
 20 VP --> *** , PP , NP
 20 NP --> *** NP , NP , ***
 20 NP --> NP , X ,
 19 NP --> NP ADJP , VP
 19 SBAR --> WHADVP S , S
 19 S --> ' ' S , ' ' SINV .
 19 SBAR --> *** , S
 19 NP --> *** ' ' *** , ' '
 18 S --> S , SBAR .
 18 S --> PP , S , S .
 18 S --> ADVP , S .
 18 S --> ' ' PP , S , ' '
 18 NP --> NP , NP , NP , NP , NP , NP , and NP
 18 S --> PP , NP *** VP
 18 VP --> *** , VP
 18 NP --> NP , NP , NP , NP
 18 VP --> *** NP PP , ADVP
 18 S --> , PP , S
 18 PP --> PP , and PP
 17 S --> NP *** VP , S
 17 S --> ADJP , S
 16 S --> ' ' NP VP , ' ' S
 16 S --> PP , *** , S .
 16 S --> SBAR , S , S .
 16 S --> *** , S , S .
 16 S --> NP ' ' *** VP , ' '
 16 VP --> *** , *** , SBAR
 16 S --> S , S , and S
 16 S --> NP VP , VP
 15 S --> *** , NP VP
 15 S --> *** PP , S .
 15 SBAR --> SBAR , and SBAR
 15 VP --> *** , PP
 15 NP --> NP , ***
 15 VP --> VP , VP

15 NP --> ' ' *** PP , ' '

 14 NP --> NP , NP *** NP

 14 S --> NP , *** , *** VP .

 14 S --> ' ' *** , S , ' '

 14 VP --> VP , VP , and VP

 14 ADJP --> *** , *** , ***

 13 S --> ' ' NP *** VP , ' ' S

 13 S --> S , PP

 13 S --> S , NP

 13 S --> ' ' SBAR , S

 12 NP --> NP , NP , NP , NP , NP , NP and NP ,

 12 SINV --> ' ' S , ' ' VP .

 12 NP --> NP , NP , and NP ,

 12 ADJP --> ADJP , ADJP

 12 NP --> *** , *** , *** and ***

 12 S --> NP *** VP , VP

 12 S --> PP NP , S

 12 VP --> *** NP PP , SBAR

 12 NP --> *** PP , PP ,

 12 S --> NP , S , *** VP .

 12 VP --> *** , SBAR

 12 NP --> NP , NP , SBARQ ,

 12 S --> PP *** , S

 12 S --> ' ' PP , S

 12 NP --> NP , NP or NP

 11 S --> ' ' NP *** VP , ' ' SINV

 11 S --> ADVP , NP VP

 11 NP --> *** , *** PP

 11 S --> ' ' S , ' ' S .

 11 PP --> PP , PP

 11 S --> S , S and S

 11 X --> X , and X

 10 SBAR --> *** S , S , S

 10 NP --> NP , VP , SBAR

 10 NP --> NP , or NP , PP

 10 S --> S , S , *** S

 10 S --> NP , ADVP , VP

 10 VP --> VP , VP , VP and VP

 10 SBAR --> *** , PP , S

 10 VP --> *** NP , PP , PP

 10 NP --> NP (NP , NP)

 10 VP --> *** NP PP , NP

10 X --> VP , and X
 10 S --> NP , PP , S
 10 S --> *** NP , S
 10 S --> *** S , S
 10 ADJP --> ' ' *** , ' '
 10 VP --> *** PP , ADVP ,
 10 NP --> *** PP , SBAR
 10 VP --> *** ADJP , PP
 10 VP --> *** NP , *** , PP
 10 S --> SBAR , S , S
 10 S --> PP , *** , S
 10 S --> *** , SBAR , S
 10 NP --> *** , NP , ***
 9 NP --> NP , NP , SBAR ,
 9 S --> *** , SBAR
 9 NP --> *** , SBARQ
 9 PP --> PP , *** PP
 9 S --> S , PP .
 9 NP --> *** , NP
 9 S --> S , SINV .
 9 S --> NP VP , ' '
 9 VP --> *** NP , *** PP
 9 VP --> *** , *** PP
 9 SBAR --> SBAR , *** SBAR
 9 X --> X , *** X
 8 S --> ' ' S , S , ' '
 8 VP --> *** ADJP , VP
 8 S --> NP , SBAR , VP
 8 S --> , SBAR , S
 8 VP --> *** , PP , SBAR
 8 NP --> *** , NP ,
 8 VP --> *** , *** , NP
 8 NP --> NP , NP , or NP
 8 NP --> NP , ADVP ,
 8 VP --> *** ADJP , SBAR
 8 S --> NP , S , VP .
 8 S --> ' ' S , ' ' SINV
 8 NP --> NP , NP , NP , NP , and NP
 8 VP --> VP , VP , VP , VP , and VP
 8 NP --> (NP , NP , NP)
 8 NP --> NP NP , NP ,
 8 S --> NP , VP , VP

8 S --> *** ADVP , S
8 VP --> *** PP , ADVP , PP
8 S --> NP , S , S
8 S --> NP VP ,
8 NP --> NP , X
8 NP --> *** , ADJP
8 NP --> NP , NP , NP , NP , NP
7 NP --> NP , *** NP PP
7 S --> *** VP , S
7 S --> NP VP , *** S
7 VP --> *** S , PP

Appendix C

Outputs of the Classification

C.1 Classified Syntax-patterns

(1) Elements in a Series -----> 2699 = 18%

(1.1) Words in a Series -----> 288 = 2%

206 NP --> ** , **
28 NP --> ** , ** , **
27 NP --> ** , ** and **
15 NP --> NP , **
12 NP --> ** , ** , ** and **

(1.2) Phrases in a Series -----> 1113 = 7%

223 NP --> NP , NP and NP
134 NP --> NP , NP , NP and NP
84 NP --> NP , NP , and NP
65 VP --> VP , and VP
63 NP --> NP , and NP
54 NP --> NP , NP , NP
50 VP --> VP , ** VP
36 NP --> NP , NP , NP , and NP
36 NP --> NP , NP , NP , NP and NP
34 VP --> VP , VP and VP
30 NP --> NP , and NP ,
28 NP --> NP , NP , NP , NP , NP and NP
25 NP --> NP , NP , NP , NP , NP , NP and NP

23 NP --> (NP , NP)
 18 PP --> PP , and PP
 18 NP --> NP , NP , NP , NP , NP , NP , and NP
 18 NP --> NP , NP , NP , NP
 15 VP --> VP , VP
 14 VP --> VP , VP , and VP
 12 NP --> NP , NP or NP
 12 NP --> NP , NP , and NP ,
 12 NP --> NP , NP , NP , NP , NP , NP and NP ,
 11 X --> X , and X
 11 PP --> PP , PP
 10 VP --> VP , VP , VP and VP
 10 ADJP --> ' ' *** , ' '
 9 X --> X , *** X
 9 VP --> *** , *** PP
 9 PP --> PP , *** PP
 8 VP --> VP , VP , VP , VP , and VP
 8 NP --> NP , NP , or NP
 8 NP --> NP , NP , NP , NP , and NP
 8 NP --> NP , NP , NP , NP , NP
 8 NP --> (NP , NP , NP)

(1.3) Clauses in a Series -----> 114 = 1%

45 S --> NP VP , S
 15 SBAR --> SBAR , and SBAR
 11 S --> S , S and S
 10 X --> VP , and X
 10 S --> *** S , S
 9 S --> S , SINV .
 7 X --> X , and VP
 7 X --> X , *** VP

(1.4) Coordinate Clauses in Series -----> 1041 = 7%

276 S --> S , S
 197 S --> S , and S
 146 S --> S , *** S
 89 S --> S , S .
 74 S --> S , *** S .
 66 S --> S , NP VP .
 55 S --> S , and S .
 37 S --> S , SINV
 28 S --> ' ' S , and S , ' '

28 S --> ' ' S , *** S , ' '

20 SINV --> S , VP NP .

16 S --> S , S , and S

9 SBAR --> SBAR , *** SBAR

(1.5) Coordinate Adjectives -----> 143 = 1%

106 ADJP --> *** , ***

14 ADJP --> *** , *** , ***

12 ADJP --> ADJP , ADJP

11 NP --> *** , *** PP

(2) Sentence-initial Elements -----> 2932 = 20%

(2.1) Introductory Words -----> 450 = 3%

283 S --> *** , S

129 S --> *** , S .

15 S --> *** , NP VP

14 S --> ' ' *** , S , ' '

9 S --> *** , SBAR

(2.2) Introductory Phrases -----> 1865 = 13%

886 S --> PP , S

369 S --> PP , S .

98 S --> NP , S

81 S --> VP , S

67 S --> ADVP , S

59 S --> NP , S .

46 S --> *** PP , S

40 S --> PP , NP VP

39 S --> VP , S .

19 SBAR --> WHADVP S , S

18 S --> ' ' PP , S , ' '

18 S --> PP , NP *** VP

18 S --> ADVP , S .

17 S --> ADJP , S

15 S --> *** PP , S .

12 S --> ' ' PP , S

12 S --> PP NP , S

12 S --> PP *** , S

11 S --> ADVP , NP VP

10 S --> NP , PP , S

10 S --> *** NP , S
8 S --> *** ADVP , S

(2.3) Introductory Clauses -----> 617 = 4%

297 S --> SBAR , S
119 S --> SBAR , S .
112 SBAR --> *** S , S
34 S --> ' ' SBAR , S , ' '
26 S --> *** SBAR , S
13 S --> ' ' SBAR , S
8 S --> ' ' S , S , ' '
8 S --> , SBAR , S

(3) Sentence-final Elements -----> 735 = 5%

(3.1) Final Phrases -----> 455 = 3%

59 VP --> *** NP , VP
46 VP --> *** PP , ADVP
39 S --> S , VP .
31 VP --> *** NP PP , PP
31 VP --> *** NP , ADVP
30 VP --> *** NP PP , VP
29 VP --> *** PP , VP
29 VP --> *** PP , PP
18 VP --> *** NP PP , ADVP
18 VP --> *** , VP
16 S --> NP VP , VP
15 VP --> *** , PP
13 S --> S , PP
12 S --> NP *** VP , VP
10 VP --> *** PP , ADVP ,
10 VP --> *** ADJP , PP
9 VP --> *** NP , *** PP
9 S --> S , PP .
8 VP --> *** PP , ADVP , PP
8 VP --> *** ADJP , VP
8 S --> NP VP ,
7 VP --> *** S , PP

(3.2) Final Clauses -----> 239 = 2%

101 S --> S , SBAR

28 VP --> *** NP , SBAR
 26 VP --> *** PP , SBAR
 22 VP --> *** NP PP , S
 18 S --> S , SBAR .
 17 S --> NP *** VP , S
 12 VP --> *** NP PP , SBAR
 8 VP --> *** ADJP , SBAR
 7 VP --> *** PP PP , SBAR

(3.3) Absolute Phrases -----> 41 = 0%

28 VP --> *** PP , S
 13 S --> S , NP

(4) Nonrestrictive Phrases or Clauses -----> 2417 = 16%

(4.1) Nonrestrictive Phrases -----> 945 = 6%

336 NP --> NP , VP ,
 185 NP --> NP , VP
 67 VP --> *** NP , PP
 56 NP --> NP , PP ,
 52 NP --> NP , ADJP ,
 47 NP --> NP , ADJP
 36 S --> S , VP
 28 NP --> *** , ADJP ,
 26 NP --> *** , PP
 23 NP --> *** PP , PP
 22 NP --> NP , PP
 19 NP --> NP ADJP , VP
 12 NP --> *** PP , PP ,
 10 VP --> *** NP , PP , PP
 10 NP --> NP , VP , SBAR
 8 NP --> NP , ADVP ,
 8 NP --> *** , ADJP

(4.2) Nonrestrictive Clauses -----> 1472 = 10%

505 NP --> NP , SBAR
 350 NP --> NP , SBARQ ,
 176 NP --> NP , SBAR ,
 70 NP --> NP , ADVP
 66 NP --> *** , SBARQ ,
 61 NP --> *** , SBAR

50 NP --> NP , SBARQ
 38 NP --> NP , S ,
 36 S --> NP , VP
 34 S --> NP , *** VP
 34 NP --> *** , SBAR ,
 33 NP --> NP , S
 10 NP --> *** PP , SBAR
 9 NP --> *** , SBARQ

(5) Appositives -----> 3766 = 25%

1812 NP --> NP , NP ,
 1252 NP --> NP , NP
 140 NP --> NP , or NP ,
 137 NP --> NP , or NP
 54 NP --> ' ' *** , ' '
 52 NP --> NP , *** NP
 48 NP --> NP , NP , NP ,
 42 NP --> NP , NP , PP
 38 NP --> NP , NP , SBAR
 36 NP --> NP , *** NP ,
 32 NP --> *** , *** ,
 20 NP --> *** NP , NP , ***
 15 NP --> ' ' *** PP , ' '
 14 NP --> NP , NP *** NP
 12 NP --> NP , NP , SBARQ ,
 10 VP --> *** NP PP , NP
 10 NP --> NP , or NP , PP
 9 NP --> NP , NP , SBAR ,
 9 NP --> *** , NP
 8 NP --> NP NP , NP ,
 8 NP --> NP , X
 8 NP --> *** , NP ,

(6) Interrupters -----> 929 = 6%

80 S --> NP , PP , VP
 74 NP --> *** , PP ,
 56 S --> NP , S , VP
 56 S --> *** , S , S

52 S --> NP , PP , *** VP
 50 S --> NP , *** , VP
 48 S --> PP , S , S
 38 S --> *** , PP , S
 32 S --> NP , S , *** VP
 30 S --> NP , PP , VP .
 30 S --> NP , *** , VP .
 24 S --> NP , PP , *** VP .
 24 S --> NP , *** , *** VP
 22 S --> PP , PP , S
 20 VP --> *** , PP , NP
 20 NP --> NP , X ,
 19 SBAR --> *** , S
 18 S --> PP , S , S .
 18 S --> , PP , S
 16 VP --> *** , *** , SBAR
 16 S --> SBAR , S , S .
 16 S --> PP , *** , S .
 16 S --> *** , S , S .
 14 S --> NP , *** , *** VP .
 12 S --> NP , S , *** VP .
 10 VP --> *** NP , *** , PP
 10 SBAR --> *** S , S , S
 10 SBAR --> *** , PP , S
 10 S --> SBAR , S , S
 10 S --> S , S , *** S
 10 S --> PP , *** , S
 10 S --> NP , ADVP , VP
 10 S --> *** , SBAR , S
 8 VP --> *** , PP , SBAR
 8 VP --> *** , *** , NP
 8 S --> NP , VP , VP
 8 S --> NP , SBAR , VP
 8 S --> NP , S , VP .
 8 S --> NP , S , S

(7) Quotations -----> 661 = 4%

176 S --> ' ' NP VP , ' '
 122 S --> ' ' NP *** VP , ' '
 61 VP --> *** , S

55 VP --> *** NP , S
43 S --> ' ' S , ' ' NP VP .
41 SINV --> ' ' S , ' ' VP NP .
29 S --> ' ' NP VP , ' ' SINV
19 S --> ' ' S , ' ' SINV .
19 NP --> *** ' ' *** , ' '
16 S --> ' ' NP VP , ' ' S
16 S --> NP ' ' *** VP , ' '
13 S --> ' ' NP *** VP , ' ' S
12 SINV --> ' ' S , ' ' VP .
11 S --> ' ' S , ' ' S .
11 S --> ' ' NP *** VP , ' ' SINV
9 S --> NP VP , ' '
8 S --> ' ' S , ' ' SINV

C.2 Similar Syntax-patterns Brought Together

(1) Elements in a Series

(1.1) Words in a Series

NP --> *** , ***
 NP --> *** , *** , ***
 NP --> *** , *** , *** and ***
 NP --> *** , *** and ***
 NP --> NP , ***

(1.2) Phrases in a Series

ADJP --> ' ' *** , ' '
 NP --> (NP , NP)
 NP --> (NP , NP , NP)
 NP --> NP , NP , NP
 NP --> NP , NP , NP , NP
 NP --> NP , NP , NP , NP , NP , NP
 NP --> NP , NP , NP , NP , NP , NP , and NP
 NP --> NP , NP , NP , NP , NP , NP and NP
 NP --> NP , NP , NP , NP , NP , NP and NP ,
 NP --> NP , NP , NP , NP , NP and NP
 NP --> NP , NP , NP , NP , and NP
 NP --> NP , NP , NP , NP and NP
 NP --> NP , NP , NP , and NP
 NP --> NP , NP , NP and NP
 NP --> NP , NP , and NP
 NP --> NP , NP , and NP ,
 NP --> NP , NP , or NP
 NP --> NP , NP and NP
 NP --> NP , NP or NP
 NP --> NP , and NP
 NP --> NP , and NP ,
 PP --> PP , *** PP
 PP --> PP , PP
 PP --> PP , and PP
 VP --> *** , *** PP
 VP --> VP , *** VP
 VP --> VP , VP
 VP --> VP , VP , VP , VP , and VP
 VP --> VP , VP , VP and VP

VP --> VP , VP , and VP
 VP --> VP , VP and VP
 VP --> VP , and VP
 X --> X , *** X
 X --> X , and X

(1.3) Clauses in a Series

S --> *** S , S
 S --> NP VP , S
 S --> S , S and S
 S --> S , SINV .
 SBAR --> SBAR , and SBAR
 X --> VP , and X
 X --> X , *** VP
 X --> X , and VP

(1.4) Coordinate Clauses in Series

S --> S , *** S
 S --> S , *** S .
 S --> S , NP VP .
 S --> S , S
 S --> S , S , and S
 S --> S , S .
 S --> S , SINV
 S --> S , and S
 S --> S , and S .
 S --> ' S , *** S , '
 S --> ' S , and S , '
 SBAR --> SBAR , *** SBAR
 SINV --> S , VP NP .

(1.5) Coordinate Adjectives

ADJP --> *** , ***
 ADJP --> *** , *** , ***
 ADJP --> ADJP , ADJP
 NP --> *** , *** PP

(2) Sentence-initial Elements

(2.1) Introductory Words

S --> *** , NP VP

S --> *** , S
 S --> *** , S .
 S --> *** , SBAR
 S --> " *** , S , "

(2.2) Introductory Phrases

S --> *** ADVP , S
 S --> *** NP , S
 S --> *** PP , S
 S --> *** PP , S .
 S --> ADJP , S
 S --> ADVP , NP VP
 S --> ADVP , S
 S --> ADVP , S .
 S --> NP , PP , S
 S --> NP , S
 S --> NP , S .
 S --> PP *** , S
 S --> PP , NP *** VP
 S --> PP , NP VP
 S --> PP , S
 S --> PP , S .
 S --> PP NP , S
 S --> VP , S
 S --> VP , S .
 S --> " PP , S
 S --> " PP , S , "
 SBAR --> WHADVP S , S

(2.3) Introductory Clauses

S --> *** SBAR , S
 S --> , SBAR , S
 S --> SBAR , S
 S --> SBAR , S .
 S --> " S , S , "
 S --> " SBAR , S
 S --> " SBAR , S , "
 SBAR --> *** S , S

(3) Sentence-final Elements

(3.1) Final Phrases

S --> NP *** VP , VP
 S --> NP VP ,
 S --> NP VP , VP
 S --> S , PP
 S --> S , PP .
 S --> S , VP .
 VP --> *** , PP
 VP --> *** , VP
 VP --> *** ADJP , PP
 VP --> *** ADJP , VP
 VP --> *** NP , *** PP
 VP --> *** NP , ADVP
 VP --> *** NP , VP
 VP --> *** NP PP , ADVP
 VP --> *** NP PP , PP
 VP --> *** NP PP , VP
 VP --> *** PP , ADVP
 VP --> *** PP , ADVP ,
 VP --> *** PP , ADVP , PP
 VP --> *** PP , PP
 VP --> *** PP , VP
 VP --> *** S , PP

(3.2) Final Clauses

S --> NP *** VP , S
 S --> S , SBAR
 S --> S , SBAR .
 VP --> *** ADJP , SBAR
 VP --> *** NP , SBAR
 VP --> *** NP PP , S
 VP --> *** NP PP , SBAR
 VP --> *** PP , SBAR
 VP --> *** PP PP , SBAR

(3.3) Absolute Phrases

S --> S , NP
 VP --> *** PP , S

(4) Nonrestrictive Phrases or Clauses

(4.1) Nonrestrictive Phrases

NP --> *** , ADJP
 NP --> *** , ADJP ,
 NP --> *** , PP
 NP --> *** PP , PP
 NP --> *** PP , PP ,
 NP --> NP , ADJP
 NP --> NP , ADJP ,
 NP --> NP , ADVP ,
 NP --> NP , PP
 NP --> NP , PP ,
 NP --> NP , VP
 NP --> NP , VP ,
 NP --> NP , VP , SBAR
 NP --> NP ADJP , VP
 S --> S , VP
 VP --> *** NP , PP
 VP --> *** NP , PP , PP

(4.2) Nonrestrictive Clauses

NP --> *** , SBAR
 NP --> *** , SBAR ,
 NP --> *** , SBARQ
 NP --> *** , SBARQ ,
 NP --> *** PP , SBAR
 NP --> NP , ADVP
 NP --> NP , S
 NP --> NP , S ,
 NP --> NP , SBAR
 NP --> NP , SBAR ,
 NP --> NP , SBARQ
 NP --> NP , SBARQ ,
 S --> NP , *** VP
 S --> NP , VP

(5) Appositives

NP --> *** , *** ,
 NP --> *** , NP
 NP --> *** , NP ,
 NP --> *** NP , NP , ***

NP --> NP , *** NP
 NP --> NP , *** NP ,
 NP --> NP , NP
 NP --> NP , NP *** NP
 NP --> NP , NP ,
 NP --> NP , NP , NP ,
 NP --> NP , NP , PP
 NP --> NP , NP , SBAR
 NP --> NP , NP , SBAR ,
 NP --> NP , NP , SBARQ ,
 NP --> NP , X
 NP --> NP , or NP
 NP --> NP , or NP ,
 NP --> NP , or NP , PP
 NP --> NP NP , NP ,
 NP --> ' ' *** , ' '
 NP --> ' ' *** PP , ' '
 VP --> *** NP PP , NP

(6) Interrupters

NP --> *** , PP ,
 NP --> NP , X ,
 S --> *** , PP , S
 S --> *** , S , S
 S --> *** , S , S .
 S --> *** , SBAR , S
 S --> , PP , S
 S --> NP , *** , *** VP
 S --> NP , *** , *** VP .
 S --> NP , *** , VP
 S --> NP , *** , VP .
 S --> NP , ADVP , VP
 S --> NP , PP , *** VP
 S --> NP , PP , *** VP .
 S --> NP , PP , VP
 S --> NP , PP , VP .
 S --> NP , S , *** VP
 S --> NP , S , *** VP .
 S --> NP , S , S
 S --> NP , S , VP

S --> NP , S , VP .
 S --> NP , SBAR , VP
 S --> NP , VP , VP
 S --> PP , *** , S
 S --> PP , *** , S .
 S --> PP , PP , S
 S --> PP , S , S
 S --> PP , S , S .
 S --> S , S , *** S
 S --> SBAR , S , S
 S --> SBAR , S , S .
 SBAR --> *** , PP , S
 SBAR --> *** , S
 SBAR --> *** S , S , S
 VP --> *** , *** , NP
 VP --> *** , *** , SBAR
 VP --> *** , PP , NP
 VP --> *** , PP , SBAR
 VP --> *** NP , *** , PP

(7) Quotations

NP --> *** ' ' *** , ' '
 S --> NP VP , ' '
 S --> NP ' ' *** VP , ' '
 S --> ' ' NP *** VP , ' '
 S --> ' ' NP *** VP , ' ' S
 S --> ' ' NP *** VP , ' ' SINV
 S --> ' ' NP VP , ' '
 S --> ' ' NP VP , ' ' S
 S --> ' ' NP VP , ' ' SINV
 S --> ' ' S , ' ' NP VP .
 S --> ' ' S , ' ' S .
 S --> ' ' S , ' ' SINV
 S --> ' ' S , ' ' SINV .
 SINV --> ' ' S , ' ' VP .
 SINV --> ' ' S , ' ' VP NP .
 VP --> *** , S
 VP --> *** NP , S

Appendix D

Source Code of the AWK Program

```
# Author: Murat Bayraktar
# Date of Start: 11 Jan 1996
# This AWK program converts the Penn Treebank ParseTag Corpus from
# LISP format to Prolog format

BEGIN { fileStart = 0
        startFile = "../corpus/dj/10/W1000.PAR" }
FNR == 1 { if(fileStart == 0)
            if(FILENAME == startFile)
                fileStart = 1
            else
                next
            print "fileName('" FILENAME "')" }
fileStart == 0 { next }
$0 ~ /^\*/ && $0 ~ /\*/ { next }
NF == 0 { next }
$0 ~ /END_OF_TEXT_UNIT/ || $0 ~ /END_OF_TEXT_UNIT/ { if($0 ~ /\(.+\)\( )*\$/)
    next
    else if($0 ~ /\)\+(\ )*\$/)
    {
        gsub(/END_OF_TEXT_UNIT/, "", $NF)
        gsub(/END_OF_TEXT_UNIT/, "", $NF)
    }
    else
        next }
$0 ~ /\( )*\(\(/ { gsub(/\(\(/, "((", $1)
    ind = index($1, "(")
    $1 = substr($1, 1, ind) "t" substr($1, ind+2) "(" }
$0 ~ /\( )*\(/ || $0 ~ /\( )*\(/ { $1 = "textUnit" $1 }
*$0 ~ /\( )*\(\(S / || $0 ~ /\( )*\(\(S$/ { $1 = "textUnit( tS(" }
```

```

{ start = 0
  for(i=1; i<=NF; i++)
    if($i ~ /^textUnit/)
      continue
    else
      if($i ~ /\^[A-Z]+(-[1-9]+)?$/)
        # if parse tag
        {
          $i = $i "("
          $i = "t" substr($i, 2)
          gsub(/-/ , "_", $i)
        }
      else # if punctuation alone
        if($i ~ /^[.,;:?!]|(')|(')|(--)|(\*[A-Z]+\*)|(\.\.\.)|(')|(')$/)
          $i = "tPUNC(\"" $i "\"")
      else # if punc. attached to ")"
        if($i ~ /^[.,;:?!]|(')|(')|(--)|(\*[A-Z]+\*)|(\.\.\.)|(')|(')\)/)
        {
          if($i ~ /(\.\.\.)/)
            $i = "tPUNC(\"" substr($i, 1, 3) "\"") substr($i, 4)
          else
            if($i ~ /^[.,;:?!]/)
              $i = "tPUNC(\"" substr($i, 1, 1) "\"") substr($i, 2)
            else
              if($i ~ /((')|('))/)
                $i = "tPUNC(\"" substr($i, 1, 2) "\"") substr($i, 3)
            else
              if($i ~ /((')|('))/)
                $i = "tPUNC(\"" substr($i, 1, 1) "\"") substr($i, 2)
            else
              if($i ~ /\*[A-Z]+\*/)
                $i = "tPUNC(\"" substr($i, 1, 5) "\"") substr($i, 6)
          } # punctuation attached to ")"
      else
        if($i ~ /(')$/ && NF == i)
          $i = "tPUNC(\"" $i "\"")
      else # if ordinary word
        if(start == 0)
          {
            start = i
            if($i ~ /\(/)
              gsub(/^(+/, "\&", $i)
            else
              $i = "\"" $i
          }

if(start > 0)
  if($NF ~ /\^)+$/)
  {
    if(!gsub(/\^)+$/, "\&", $(NF-1))
      $(NF-1) = $(NF-1) "\"
  }
else

```

```

        if(!gsub(/\)+$/, "\"&", $NF))
            $NF = $NF "\"

        oldnumtab = numtab
        numtab += gsub(/\(/, "&")
        numtab -= gsub(/\)/, "&")

        gsub(/\*LRB\*/, "(")
        gsub(/\*RRB\*/, ")")
        gsub(/\*LCB\*/, "{")
        gsub(/\*RCB\*/, "}")
        gsub(/\*LSB\*/, "[")
        gsub(/\*RSB\*/, "]")

        if(numtab == 0)
            print tabs $0 ".\n"
        else
            if($0 ~ /\($/)
                print tabs $0
            else
                print tabs $0 ","

        for(i=1;i<=NF;i++)
            if($i ~ /\($/)
                tabarray[oldnumtab++] = length($i)+1
        tabs = ""
        for(i=0;i<numtab;i++)
            for(j=0;j<tabarray[i];j++)
                tabs = tabs " "
    }

```

Appendix E

Source Code of the Prolog Program

The source code of the Prolog program was too long to be included here (about 1,500 lines). However, it is available in

<http://www.cs.bilkent.edu.tr/~bayrak/punc.pl>.