COMPUTATIONAL ASPECTS OF DISCOURSE ANNOTATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

BERFİN AKTAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF DEPARTMENT OF COGNITIVE SCIENCE

DECEMBER 2008

Approval of the Graduate School of Informatics.

Prof. Dr. Nazife Baykal
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Deniz Zeyrek
Head of Department

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

Assoc. Prof. Dr. Cem Bozşahin
Supervisor

Examining Committee Members

Prof. Dr. Deniz Zeyrek	(METU, FLE)
Assoc. Prof. Dr. Cem Bozşahin	(METU, CENG)
Prof. Dr. Wolf Konig	(METU, FLE)
Dr. Onur Tolga Şehitoğlu	(METU, CENG)
Dr. Ceyhan Temurcu	(METU, COGS)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name:	**Berfin Aktaş**
Signature :	____________

ABSTRACT

COMPUTATIONAL ASPECTS OF DISCOURSE ANNOTATION

Berfin Aktaş
M.Sc., Department of Cognitive Science
Supervisor: Assoc. Prof. Dr. Cem Bozşahin

December 2008, 39 pages

In this thesis, we aim to analyze the computational aspects of discourse annotation. Discourse is not only a concatenation of sentences; in fact the totality of discourse is more than the sum total of the sentences that constitute it. The property that differentiates discourse from a set of arbitrary sentences is defined as coherence. Coherence is established by the relations between the parts of discourse. In this study, discourse relations are considered to be set up by lexical items called discourse connectives. Systematic analysis of coherence requires an annotated corpus in which coherence relations are encoded. We developed an annotation environment to be used in an ongoing discourse level annotation project which aims to generate a theory-neutral source of coherence relations. We followed a data-driven methodology in design of the data structure employed in the annotation software. For this reason, we examined the predicate-argument structure of connectives. This analysis shows that stand-off annotation technique is more suitable than an inline method for such an annotation environment. This thesis also include a brief discussion on the formal implications of coherence relation constructions.

Keywords: Discourse Annotation, Discourse Connective, Coherence Relation, Turkish

ÖZ

SÖYLEM İŞARETLEMENİN BERİMSEL YÖNLERİ

Berfin Aktaş
Yüksek Lisans, Bilişsel Bilimler Bölümü
Tez Yöneticisi: Doç. Dr. Cem Bozşahin

Aralık 2008, 39 sayfa

Bu tezde, söylem işaretlemenin berimsel yönlerini analiz etmeyi amaçlıyoruz. Söylem sadece cümlelerin bir birleşimi değildir, aslında söylemin tümü bileşimindeki her bir parçacığın toplamından daha fazlasıdır. Söylemi herhangi bir cümleler kümesinden ayıran özellik "bağdaşıklık" olarak tanımlanır. Bağdaşıklık, söylemin parçaları arasındaki ilişkilerce sağlanır. Bu çalışmada, söylem için sözcüksel (lexicalized) bir yaklaşım kullanarak söylem ilişkilerinin söylemsel bağlaçlar denen sözcüksel öğeler ile kurulduğunu varsayıyoruz. Bağdaşıklığın sistematik analizi için bağdaşıklık ilişkilerinin işaretlenmiş olduğu bir külliyata ihtiyaç vardır. Amacı bağdaşıklık ilişkilerinin işaretlenmiş olduğu, kuram bağımsız bir veri kaynağı yaratmak olan bir söylem seviyesinde işaretleme projesinde kullanılmak üzere bir işaretleme yazılımı geliştirdik Bu işaretleme ortamında kullanılan veri yapılarının tasarımında veri yönelimli bir yöntem izledik. Bu amaçla, bağlaçların yüklem-özne yapısını inceledik. Bu analiz bize böyle bir işaretleme ortamı için "stand off" işaretleme tekniğinin "inline" yönteme göre daha uygun olduğunu gösterdi. Bu tez bağdaşıklık ilişki yapılarının biçimsel(formal) imaları üzerine kısa bir tartışma da içermektedir.

Anahtar Kelimeler: Söylem İşaretleme, Söylem Bağlacı, Bağdaşıklık İlişkisi, Türkçe

To My Family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my dear advisor Cem Bozşahin for his encouragement, guidance and willingness to share his insight and wisdom. I feel really lucky to know him as a teacher and a beloved friend. I owe special thanks to Deniz Zeyrek, the leader of the project I involved in, for her kindness, support and invaluable discussions. Many thanks to Ceyhan Temürcü for his valuable comments, to Işın Demirşahin and Ayışığı Başak Sevdik Çallı for their annotations and helpful discussions.

I especially thank to Müge Sevinç and Orkan Bayer for their support and encourgement at really difficult times. I am not sure whether I could have completed this thesis without them. I would like to thank to Özlem Sarıyıldız for the meals and love, to my dear friends Erkan Özmacun, Bade Altunel, Zafer Çeler, İdil Soyseçkin, Arman Arıkan, Muharrem Görkem, Gökhan Mansuroğlu, Arzu Yalçın, Yelda Yiğit, Yeliz Aygün for always saying me "hadiii" and being with me, to Tansel Okay for his precious interest to my research.

The closing word would go to my parents Güllü&Ali Haydar Aktaş and my brother, Ruşen Aktaş. I thank to them for their endless support and trust. I am grateful to them because my intellectual interests and personality are shaped by these extraordinary people. And Ruşencim, my dear, you can not know how privileged I feel to have you in my life. This thesis is dedicated to my great family with love and gratitude.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Coherence, as a discourse phenomenon, is one of the most discussed concepts in discourse area of linguistics. Systematic analysis of coherence could be realized if annotation data of coherence relations[1] do exist. This analysis will reveal how the sentences in a text are related with each other. A deep investigation of coherence phenomenon elicits major points of human communication in addition to the theoretical aspects of the language. Apart from these, a good understanding of coherence will enhance the computational applications of natural language such as information retrieval, question answering, text summarization, and machine translation systems.

Coherence is defined as the property that distinguishes discourse from being an arbitrary set of sentences. The similarities and differences of discourse theories can be revealed by referring to their descriptions of discourse and coherence (Webber, 2006). Structural accounts of discourse have the assumption that discourse has a hierarchical structure and coherence is achieved via structural relations (Mann & Thompson, 1988), (Polanyi, 1996), (Lascarides & Asher, 1993), (Lascarides & Asher, 2007). In contrast to structural frameworks, presuppositional accounts claim that the source of coherence is the non-structural cohesive links between discourse units (Halliday & Hasan, 1976). There are also hybrid accounts which assign a structure to discourse but also claim that there exist anaphoric relations as well as structural relations in a discourse (Webber, 2004).

The main aim of this thesis is to present computational aspects of discourse annotation on the basis of a lexically grounded approach to discourse relations. We implemented an annotation tool to be used in the annotations of ongoing discourse level annotation project[2] (TDAP). For English there exist large scale discourse level annotation resources like RST TreeBank (Carlson et al., 2003), Discourse GraphBank

[1] In this thesis, coherence relations refer to informational relations in discourse and we use the terms "coherence relation" and "discourse relation" interchangeably throughout the thesis.

[2] ODTÜ Metin Düzeyinde İşaretlenmiş Derlem: ODTÜ-MEDİD (Yönetici: Prof. Dr. Deniz Zeyrek)

(Wolf et al., 2003) and PDTB (Miltsakaki et al., 2004). TDAP is the first attempt for Turkish. The annotation scheme of PDTB is adopted in TDAP. The aim of TDAP is to generate a theory-neutral discourse level data source as a final product. In order to achieve this, no specific account of discourse is employed as a data gathering methodology. The only assumption is coming from the lexical approach of TDAP which is the assertion that discourse relations are set up by lexical items which are called discourse connectives. All discourse relations are annotated in the same way regardless of the grammatical classes of the connectives. The investigation of dependency structures of connectives shapes the data representation of annotation.

In syntax, dependency constructions determine the computational power required to capture the natural languages. The existence of unbounded cross-serial dependencies in natural language syntax necessitates more computational resources than context-free grammars have. Joshi (1985) argues that a formal grammar class which is slightly more powerful than context-free grammars can capture natural languages. These class of grammars are called Mildly Context Sensitive Grammars (MCSGs). In this thesis, we include a brief discussion on the implications of dependency constructions of discourse from the view point of formal theory.

The thesis is organized as follows:

In chapter 2, we present the core ideas of major discourse accounts.

Chapter 3 contains the examination of dependency structures in Turkish discourse.

In chapter 4, we discuss formal aspects of discourse relations. We introduce formal grammar accounts briefly and discuss the concept of "mildly context sensitivity" on the ground of dependencies.

In chapter 5, we introduce our data driven design of annotation structures. We discuss how the constructions presented in chapter 3 affect our data representation. In addition to that, we also propose the software requirements that a discourse annotation tool should come with.

Chapter 6 consists of the summary of our conclusions.

CHAPTER 2

COHERENCE AND DISCOURSE RELATIONS

2.1 Halliday and Hasan (1976)

Halliday & Hasan (1976) have a presuppositional approach to discourse relations. In this theory, discourse relations are formed by non-structural links between discourse units. H&H define text as a linguistic unit. It is a semantic unit rather than a grammatical one. The concept of 'being a text' is termed as texture. We use the terms texture and coherence interchangeably as in Carrell (1982). Texture is achieved via the cohesive links within the text.

Cohesion can be described as the dependency of the interpretation of one discourse element to that of another one. It is a linguistic phenomena contributing to texture. H&H define five types of cohesion: reference, substitution, ellipsis, lexical cohesion and conjunction. H&H state that the structure of discourse, if it exists, is different from the structure in sentence-level:

> Whatever relation there is among the parts of a text - the sentences, the paragraphs, or turns in a dialogue - it is not the same as structure in the usual sense, the relation which links the parts of a sentence or a clause (Halliday & Hasan, 1976, 6).

H&H use the term 'tie' to refer to a single instance of cohesion. A text can be characterized by the number and type of ties it possesses. H&H assert that many linguistic phenomena can be expounded by analyzing the cohesive links in texts. According to them, there are certain features that should be taken into account for analyzing these links to provide a comprehensive account of the cohesion. The notion of cohesion can best be characterized in terms of the properties of its instances. Since H&H call these instances as tie, it can safely be asserted that text segments can be characterized in terms of its cohesive properties via the concept of *tie* in this framework.

Tie is a directional concept. Since ties are presuppositional links, they can be either anaphoric or cataphoric. The relative positions of presupposing and presupposed elements determine the direction of a tie. The coding scheme of any tie should contain information on the direction of the tie. The distance between the presupposed and presupposing elements distinguishes ties into three classes: Immediate, mediated and remote. If the presupposed element is in the immediately preceding sentence, then the related tie is referred as an immediate tie. If the presupposed element is distant and it is also cohesive, a chain of cohesive presuppositions may have to be followed in order to reach the target item. This kind of tie is called as mediated tie. And the last type of tie which is the remote class is referred when the presupposed element is distant and there is no intermediate references to that element. This classification suggests that a tie can be both mediated and remote at the same time. Any linguistic analysis of ties should take this classification into account, therefore coding scheme of the notion of tie also involves the type of tie. Lastly, the presupposed element should also be marked.

2.2 Rhetorical Structure Theory (RST)

RST (Mann & Thompson, 1988) is established on the conception that text has an underlying structure which is formed by discourse relations. RST is a descriptive approach to text organization. The central concepts in RST are rhetorical relations. Text coherence is the notion that differentiates the text from a set of arbitrary sentences and it is established by rhetorical relations. Atomic units of text processing are clauses or larger units composed of clauses and there is a requirement that these units must have no overlapping parts. The aim of the RST analysis is to span the whole text and construct a unique tree which covers the structure of whole text. An RST tree doesn't have to be a binary tree; a relation between two or more discourse units is allowed. As in syntax, any discourse element is part of only one larger element.

In RST, there are two levels of "building blocks" that occur in texts. First level deals with "nuclearity", and the second level deals with schemas. Nuclearity is the measure of the importance of the related text unit. Important units are assigned as nucleus and the others are satellites. Each text unit is assigned a status which represents its nuclearity. Relations that occur between equally important elements are called as symmetrical relations. They are asymmetrical in other cases.

The second level elements of RST are schemas. RST schemas are context-free rules which define how the discourse structure is created from text units. One of the major constraints of RST schemas is that relations hold only between adjacent units in the text.

The RST Discourse Treebank (Carlson et al., 2003) utilizes RST as a data gathering strategy. 385 Wall Street Journal articles are annotated by the following the steps below:

- Text is segmented into its units. Units are non-overlapping text spans.
- The status of text units are labeled as nucleus or satellite.
- Instances of previously determined set of relations are determined.

2.3 Wolf and Gibson (2005)

Wolf & Gibson (2005) have an account similar to RST. Their difference lies in the representation of discourse structure. Wolf and Gibson claim that trees are not adequate data structures to describe the discourse structure (Wolf, 2005), (Wolf & Gibson, 2005). Instead of tree structures which are set up by the relations between adjacent text segments, Wolf and Gibson propose a directed chain graph representation which allows relations between non-adjacent segments as well. They justify this difference with the assertion that certain parts of discourse structure violate the tree structure. These parts involve crossing dependencies and nodes with multiple parents. Discourse relations are directional like those in the RST framework.

Discourse segments are non-overlapping text units. They constitute a segment group if there exist common attribution features or they share the same topic. In this account, coherence relations can be established either between discourse segments or a group of discourse segments. Unlike RST, the relations are not recursive; i.e. an established relation does not serve as an argument for another relation. Therefore, discourse structure is represented by a rather flat chain graph. Connectedness of the graph structure is the measure of the coherence of the text. An unconnected graph indicates a partially coherent text which contains unrelated discourse segments.

The Discourse Graphbank (Wolf et al., 2003) annotation project is developed upon the theoretical framework of Wolf & Gibson (2005). The following steps describe the annotation procedure of Discourse Graphbank:

- Text is segmented into its units.
- Segment groupings are constituted.
- Coherence relations are established between segments and/or group of segments.

2.4 Segmented Discourse Representation Theory (SDRT)

SDRT (Lascarides & Asher, 1993) is another structural theory of discourse and it is the enriched version of Discourse Representation Theory (DRT) with the notion of rhetorical structure. Lascarides & Asher (1993) argue that a full account of discourse can be captured by modeling the interaction between semantic content of texts and their global pragmatic structure. The structure of text is constructed by the coherence relations between discourse segments. In SRDT, the coherence relations refer to informational relations in the text.

DRT is based on the paradigm of dynamic semantics in which meaning of a discourse is a function from a discourse context to a discourse context. Meaning of a sentence is obtained from the meanings of those preceding it by making inferences not by compositional means. In discourse interpretation, the need for rhetorical relations emerges in pronoun resolution and analysis of temporal structure. SDRT models the semantics-pragmatics interface (Lascarides & Asher, 2007).

Lascarides & Asher (1993) propose some principles that governs the computation of coherence relations:

- Penguin Principle: A more specific rhetorical relation is preferred over a less specific one.
- Narration Principle: Events are described in their temporal structures.
- Push Causal Law: There exist a causal relation between two events only if the cause event is completely preceding the other one.
- Maximising Discourse Coherence: Lascarides & Asher (2007) observe that coherence quality of a text is a varying value. Therefore, in SDRT analysis, interpretations that maximize the discourse coherence are preferred. Discourse coherence value is affected by the number of rhetorical relations between two discourse items. In addition to this, the resolution of anaphoric expressions increases the discourse coherence as well.

SDRT does not allow crossing dependencies between discourse segments (Wolf, 2005). On the other hand, it does not constrain the number of parents that any node may have (Lascarides & Asher, 1991).

2.5 Discourse Lexicalized TAG (D-LTAG)

D-LTAG (Webber, 2004) is the extended version of LTAG for discourse processing purposes. D-LTAG is a lexically grounded theory which asserts that discourse relations are anchored by lexical elements. The lexical elements which signal the discourse relations are discourse connectives. Connectives are discourse level predicates and taking two abstract objects such as propositions, facts, or events (Asher, 1993) as arguments. They are lexical items belonging to the grammatical classes of coordinating conjunctions, subordinating conjunctions, subordinators, parallel constructions and discourse adverbials.

Webber et al. (2003) argue that discourse connectives can be classified into two different categories which differ in the connection types they set up. The first is structural category of connectives. The connectives belonging to the structural class take both their arguments syntactically. The other category consists of anaphoric connectives which take only the second argument syntactically and the first argument is resolved anaphorically. The difference lies in the obtainment of semantics; in the case of structural connectives, semantics is obtained compositionally while in the case of anaphoric connectives, making inference is necessary to get the semantics. In this account, the structural relations are represented by tree structure but an additional secondary structure is proposed to handle the anaphoric relations (Forbes-Riley et al., 2003).

2.5.1 Penn Discourse TreeBank (PDTB)

Penn Discourse TreeBank (Miltsakaki et al., 2004) is a large scale annotated corpus in which coherence relations are encoded. The theoretical framework upon which PDTB builds is D-LTAG. PDTB annotations include the markings of connectives and their argument spans. Abstract objects can be linked either by explicitly realized connectives or by implicit ones recognized by an inferential process. PDTB covers predicate argument structures of both implicit and explicit connectives. Except from this, semantics of the connectives in that context and attribution-related information on both connectives and arguments are also annotated.

The data source of PDTB is the Penn TreeBank (PTB) (Marcus et al., 1993). Annotated spans are linked with constituents in PTB trees. This alignment of different levels of annotation makes possible the comparison of linguistic information for different layers of structures.

2.6 METU Turkish Discourse Annotation Project (TDAP)

2.6.1 METU Turkish Corpus (MTC)

MTC (Say et al., 2002) is a written source of Turkish with approximately 2 million words. MTC contains samples of 2000 words and these samples are taken from 291 different sources published after 1990. Text sources belong to different genres including memoirs, novels, essays, interviews and news.

MTC samples are labeled with information on the author, publish date and genre of the source. In addition to these, paragraph boundaries are also marked. A small portion of MTC is annotated to create a data source which is called as METU-Sabancı TreeBank and it contains morphological and dependency features of 7262 sentences (Atalay et al., 2003).

All the natural language examples in this thesis are taken from MTC, unless stated otherwise.

2.6.2 METU Turkish Discourse Annotation Project (TDAP)

TDAP aims to annotate MTC in order to obtain a discourse-level resource. The final product is expected to be a Turkish Discourse Relation Bank. In this project, the lexically grounded approach of PDTB is adopted. As in PDTB, discourse relations are considered to be set up by lexical items i.e. discourse connectives and these connectives are discourse level predicates. The annotation process can be described as the determination of the list of these connectives and labeling of arguments for each connective. Zeyrek & Webber (2008) present how Turkish connectives are determined and what these connectives are. The valency of connectives is exactly two for Turkish. Arguments are text spans which represent abstract objects. Abstract objects can be linked either by explicitly realized connectives or by implicit ones recognized by an inferential process. TDAP, primarily, aims to annotate explicit connectives; implicit connective annotation will start after all explicit connectives are annotated.

CHAPTER 3

DEPENDENCY ANALYSIS OF DISCOURSE RELATIONS

TDAP has a lexicalized approach to discourse which asserts that coherence relations are set up by lexical conjunctive items. These items are called as connectives and they are discourse level predicates which are taking two text units as their arguments. In this thesis, we follow the annotation convention used in Miltsakaki et al. (2004) and Zeyrek & Webber (2008): connectives, <u>Conn</u>, are underlined, the argument which contains the connective, **Arg2,** is in boldface and the other argument, *Arg1,* is in italics.

TDAP follows the minimality principle to limit the amount of marked text. Minimality principle enforces the labeling of the text spans that are necessary for the interpretation of the relation. In addition to the arguments of the connectives, TDAP also annotates the supplementary material which is relevant to the relation but not necessary for the interpretation. The supplementary material to Arg1 is labeled as Sup1 and the material to Arg2 is labeled as Sup2.

TDAP has no a priori assumption on the dependency structures of the coherence relations. Therefore, we need to examine these structures in order to design an annotation environment which can handle the marking of all kinds of coherence relations. In addition to this, the complexity of these dependency structures also have an impact on the formal properties of discourse. In this chapter, we examine the dependency types of coherence relations in Turkish.

3.1 Independent Relations

The predicate argument structure of the connectives are independent from each other. In other words, there is no overlap between the arguments of different connectives. These relation types are illustrated in Fig. 3.1 [1].

Here is an example of this case:

[1] In the following figures, we use the convention that $Arg1_{Conn1}$ represents the first argument of the connective <u>Conn1</u> and the other usages are straightforward.

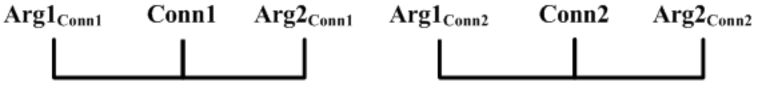


Figure 3.1. Independent Relations

(1) *Akıntıya kapılıp umulmadık bir geceyi bölüştü benimle* ve **bu kadarla kalsın istedi belki.** *Eda açısından olayın yorumu bu kadar yalın olmalı.* Ama eğer böyleyse **benim için yorumlanması olanaksız bir düşten başka kalan yok geriye şimdi.**

She was drifted with a current and shared an unexpected night with me and perhaps she wanted to keep it this much only. From the sight of Eda, the interpretation of the incident should be that simple. However, if this is the case, now there is nothing left behind for me but a dream impossible to interpret.

Conn	Arg1	Arg2
ve	Akıntıya ... benimle	bu kadarla ... belki
Ama	Eda ... olmalı	benim için ... şimdi

In (1), the relation set up by Ama is fully preceeded by the relation set up by ve. In other words, there is no overlap between the argument spans of the connectives ve and Ama.

3.2 Full Embedding

The text span of one connective with its arguments constitutes an argument of another connective.

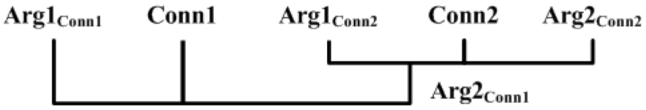


Figure 3.2. Full Embedding

We can exemplify the case of full embedding as follows:

(2)a. [..] madem **yanlış bir yerde olduğumuzu düşünüyoruz da doğru denen yere asla varamayacağımızı biliyoruz** , *senin gibi biri nasıl böyle bir soru sorar* ,[..]

b. [..] madem **yanlış bir yerde olduğumuzu düşünüyoruz** da *doğru denen yere asla varamayacağımızı biliyoruz* , senin gibi biri nasıl böyle bir soru sorar,[..]

[..] if we think that we are in a wrong place, and we know that we will never never reach the right place; how come a person like you ask such a question? [..]

Conn	Arg1	Arg2
madem	senin gibi ... sorar	yanlış ... biliyoruz
da	doğru ... biliyoruz	yanlış ... düşünüyoruz

In (2), the span of the relation headed by da constitutes the Arg2 of the connective madem.

3.3 Shared argument

The same argument is shared by two different connectives as illustrated in Figure 3.3.

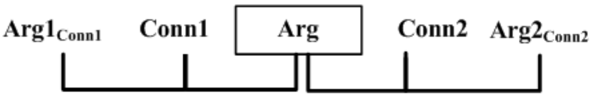


Figure 3.3. Shared Argument

The case of shared argument can be exemplified as in (3):

(3)a. *Bu sosyo - ekonomik ve sosyo - kültürel bir değişim ve dönüşümü yaşayan ve geleneksellikten modernizme geçiş sürecini henüz yaşamaya başlamış olan bir toplum için normal karşılanabilir .* Fakat **Alevi toplumu dayatan modernizm karşısında bu konumunu er geç terketmek zorunda olduğunu ve geçmiş ile sağlıklı bir hesaplaşmaya girip geleneksel değer yargılarını ve sosyo - kültürel yapısını köken taassubundan uzak bir şekilde analize tabi tutmak durumunda bulunduğunu görecektir.** Aksi halde kanaatimizce ikinci gruptaki problemleri çözmeye kolay kolay muvaffak olamayacaktır . Aynı tarihsel muhasebe ve eleştiri işlemi , Sünni kesim için de elzem ve eninde sonunda vazgeçilmez bir olgu olarak beklemektedir.

b. Bu sosyo - ekonomik ve sosyo - kültürel bir değişim ve dönüşümü yaşayan ve geleneksellikten modernizme geçiş sürecini henüz yaşamaya başlamış olan bir toplum için normal karşılanabilir . Fakat *Alevi toplumu dayatan modernizm karşısında bu konumunu er geç terketmek zorunda olduğunu ve geçmiş ile sağlıklı bir hesaplaşmaya girip geleneksel değer yargılarını ve sosyo - kültürel yapısını köken taassubundan uzak bir şekilde analize tabi tutmak durumunda bulunduğunu görecektir.* Aksi halde **kanaatimizce ikinci gruptaki problemleri çözmeye kolay kolay muvaffak olamayacaktır . Aynı tarihsel muhasebe ve eleştiri işlemi , Sünni kesim için de elzem ve eninde sonunda vazgeçilmez bir olgu olarak beklemektedir.**

This could be regarded as normal for a society living through a socio-economic and socio-cultural change and transformation which has just started the transition from traditional society to modernism. But, the Alavite society will sooner or later realize that it has to abandon its position against the imposing modernism and analyze its traditional value judgments and its socio-cultural structure by settling its accounts with the past in a manner away from fanaticism about origins. Otherwise, it will not easily succeed in solving the problems in the second group according to our opinion. The same process of accounting and criticism of history awaits the Sunni community as an essential and ultimately indispensable fact.

Conn	Arg1	Arg2
Fakat	bugünün ... değerlendirmektedir	Alevi toplumu... görecektir.
Aksi halde	Alevi toplumu ... görecektir.	kanaatimizce ... olamayacaktır

In (3), the Arg2 of Fakat is same with the Arg1 of Aksi halde. In other words, the connectives share the same text span as their arguments.

In some situations, different connectives can share both of their arguments as in the case of (4):

(4) Dedektif romanı içinden çıkılmaz gibi görünen esrarlı bir cinayetin çözümünü sunduğu için, *her şeyden önce mantığa güveni ve inancı dile getiren bir anlatı türüdür* ve bundan ötürü de **burjuva rasyonelliğinin edebiyattaki özü haline gelmiştir.**

Unraveling the solution to a seemingly intricate murder mystery, the detective novel is a narrative genre which primarily gives voice to the faith and trust in reason and being so, it has become the epitome of bourgeois rationality in literature.

Conn	Arg1	Arg2
ve	her şeyden önce ... anlatı türüdür	burjuva ... haline gelmiştir
bundan ötürü	her şeyden önce ... anlatı türüdür	burjuva ... haline gelmiştir

In (4), the relations set up by the connectives ve and bundan oturu share both of their arguments.

3.4 Properly contained argument

The argument span of one connective encapsulates the argument of another connective but they are not equal. This kind of dependency relation can be illustrated by the Figure 3.4.

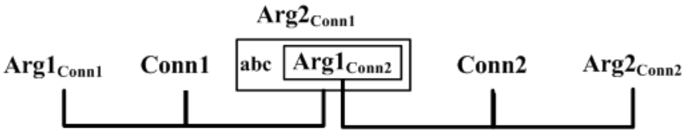


Figure 3.4. Properly Contained Argument

In (5), we exemplify the case of properly contained argument:

(5)a. *Biz yasalar karşısında evli sayılacak* , ama **gerçekte evli iki insan gibi değil de** (evlilikler sıradanlaşıyordu çünkü, tekdüze ve sıkıcıydı; biz farklı olacaktık), **aynı evi paylaşan iki öğrenci gibi yaşayacaktık.**

b. Biz yasalar karşısında evli sayılacak, ama *gerçekte evli iki insan gibi değil de* (**evlilikler sıradanlaşıyordu** çünkü, tekdüze ve sıkıcıydı; biz farklı olacaktık), aynı evi paylaşan iki öğrenci gibi yaşayacaktık.

We were to be married by law, but in reality we would live as two students sharing an apartment rather than as a really married couple (marriages were routine because they were monotonous and boring; we were to be different).

Conn	Arg1	Arg2
ama	Biz ... sayılacak	gerçekte ... değil de aynı evi ... yaşayacaktık
çünkü	gerçekte ... değil de	evlilikler sıradanlaşıyordu

In (5), the second argument of ama covers the first argument of çünkü and additional text span. Therefore the Arg2 of ama properly contains the Arg1 of çünkü.

An interesting example of this case is presented in (6). This example comes up with the question that whether the existence of attribution verbs like "dedi" as in (6) has an impact on such kind of constructions. Since this question is out of the scope of this thesis, we leave it as an open question further studies.

(6)a. *Kapıdan girdi* ve **söyler misin, hiç etkilenmedin mi yazdıklarından?, dedi.** Tersine, çok etkilendim.

b. Kapıdan girdi ve söyler misin, *hiç etkilenmedin mi yazdıklarından?,* dedi. Tersine, **çok etkilendim.**

S/he entered through the door and said “Tell me, are you not touched at all by what s/he wrote?”. On the contrary, I am very much affected.

Conn	Arg1	Arg2
ve	Kapıdan girdi	söyler misin ... dedi
Tersine	hiç ... yazdıklarından?	çok etkilendim

In (6), the Arg2 of ve properly contains the Arg1 of Tersine.

3.5 Properly Contained Relation

The argument span of one connective encapsulates the predicate argument structure of another connective but they are not equal. Encapsulating argument involves more text spans as illustrated in Figure 3.5.

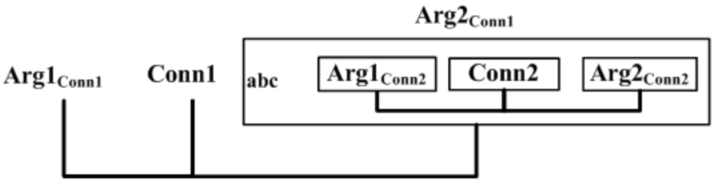


Figure 3.5. Properly Contained Relation

This kind of dependency relations can be exemplified by (7):

(7)a. *Burada bizce bir ifade bozukluğu veya çeviri yanlışı bahis konusu olabilir,* çünkü **elbiseler sanki giyildiği sürece ve yıpranmamışken yıkanamaz, fakat daha sonra yıkanabilirmiş gibi bir anlam taşımaktadır.**

b. Burada bizce bir ifade bozukluğu veya çeviri yanlışı bahis konusu olabilir, çünkü *elbiseler sanki giyildiği sürece ve yıpranmamışken yıkanamaz,* fakat **daha sonra yıkanabilirmiş** gibi bir anlam taşımaktadır.

Here a mistake of expression or mistranslation might be the case, because the meaning is as if the clothes cannot be washed as long as they are used and not worn out, but can be washed later.

Conn	Arg1	Arg2
çünkü	Burada ... olabilir	elbiseler ... taşımaktadır
fakat	elbiseler ... yıkanamaz	daha ... yıkanabilirmiş

In (7), the second argument of çünkü covers the whole relation headed by fakat and, additionally, the span of the text "gibi bir anlam taşımaktadır". Hence, (7) involves an instance of a properly contained relation.

3.6 Nested Relations

A relation is placed between an argument and connective of another relation as illustrated in Figure 3.6.

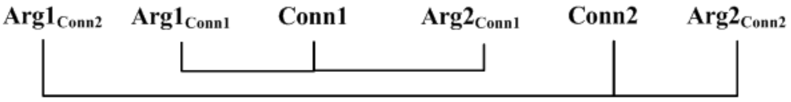

$\mathbf{Arg1}_{\mathbf{Conn2}}$
$\mathbf{Arg1}_{\mathbf{Conn1}}$
Conn1
$\mathbf{Arg2}_{\mathbf{Conn1}}$
Conn2
$\mathbf{Arg2}_{\mathbf{Conn2}}$

Figure 3.6. Nested Relations

The example (8) is presented as an instance of nested relations:

(8) *Büyük bir masada günlerce, gecelerce oturup konuşacağız - konuşmayı unuttum diyorum* da **gülüyorlar bana** - ve **biriniz kalkıp şiir okuyacak.**

We will sit and talk around a big table for days and nights - I say I have forgotten how to speak and they laugh at me - and one of you will stand up and recite poetry.

Conn	Arg1	Arg2
da	konuşmayı ... diyorum	gülüyorlar bana
ve	Büyük ... konuşacağız	biriniz ... okuyacak

In (8), the relation headed by da is properly nested between the connective ve and its first argument.

3.7 Pure Crossing

The dependency structure of a relation interleaved with the arguments or connective of another relation as shown in Fig. 3.7

(9) is an example for this dependency type:

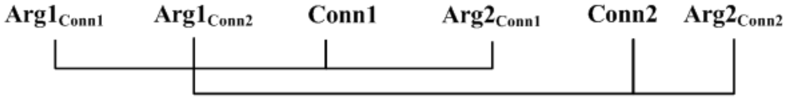


Figure 3.7. Pure Crossing

(9)a. (Constructed) *Kitabı okumaya başladım* : Okullar çoktan açılmıştı. Ardından **kapının çaldığını duydum ama yerimden kalkmadan okumaya devam ettim:** Ama bu okula henüz öğretmen atanmamıştı.

b. Kitabı okumaya başladım *Okullar çoktan açılmıştı.* Ardından kapının çaldığını duydum ama yerimden kalkmadan okumaya devam ettim: Ama **bu okula henüz öğretmen atanmamıştı.**

I started to read the book. The schools had long been opened. Then, I heard the door bell ring but I continued reading without getting up: But a teacher had not been appointed to this school yet.

Conn	Arg1	Arg2
Ardından	Kitabı okumaya başladım	kapının çaldığını ... devam ettim
Ama	Okullar çoktan açılmıştı	bu okula ... atanmamıştı

The dependencies of the example (9) are illustrated in Figure 3.8.

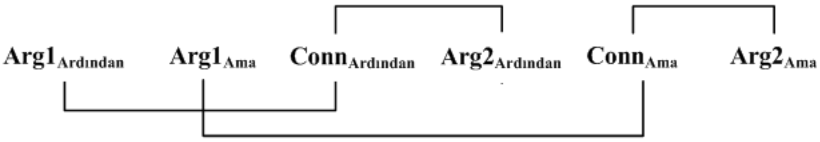


Figure 3.8. The dependencies between Ardından and its Arg1 and between Ama and its Arg1 are cross-serial.

It is possible to observe two different dependency constructions mentioned in this chapter in a single annotation. For instance, in the example (10), we observe both “Shared Argument” and “Pure Crossing” dependencies:

(10)a. *Olan biteni anlamaya, çözümlemeye çabalıyor, Saraybosna kuşatmasıyla ortaçağda kuşatılan kentler, özellikle de Simon de Montfort'un Fransa'da Katarlara karşı giriştiği kıyım arasında koşutluk kuruyordu.* **Bosna Müslümanları da Hıristiyanlık içinde batıni bir mezhep olan Bogomillerden geliyorlardı** çünkü. Dolayısıyla Papanın yüzyıllar önce Bogomiller ve Katarlar için söyledikleri onlar için de geçerliydi.

b. Olan biteni anlamaya, çözümlemeye çabalıyor, Saraybosna kuşatmasıyla ortaçağda kuşatılan kentler, özellikle de Simon de Montfort'un Fransa'da Katarlara karşı giriştiği kıyım arasında koşutluk kuruyordu. *Bosna Müslümanları da Hıristiyanlık içinde batıni bir mezhep olan Bogomillerden geliyorlardı* çünkü. Dolayısıyla **Papanın yüzyıllar önce Bogomiller ve Katarlar için söyledikleri onlar için de geçerliydi.**

S/he is trying to understand, analyze the events, seeing parallels between the Sarajevo siege and the cities under siege in the middle ages, especially the genocide of the Katars by Simon de Monfort in France. Because, Bosnian Muslims were also descendents of Bogomills, a mystic sect of Christianity. Thus, what the Pope had said of Katars and Bogomills centuries ago was also valid for them.

Conn	Arg1	Arg2
çünkü	Olan biteni ... kuruyordu	Bosna ... geliyorlardı
Dolayısıyla	Bosna ... geliyorlardı	Papanın ... geçerliydi

The dependencies of this example are illustrated in Fig. 3.9. Since çünkü is coming after its Arg2, a sort of crossing dependency also exists in this annotation.

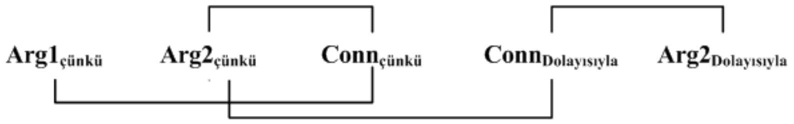


Figure 3.9. The dependencies between çünkü and Arg1 and between Dolayısıyla and Arg1 are cross-serial (The Arg2 of çünkü and Arg1 of Dolayısıyla cover the same text span).

CHAPTER 4

FORMAL ASPECTS OF DISCOURSE

Dependency analysis of discourse elements is a good starting point for investigating how much computational power is required to describe the structure of discourse.

We utilize the conceptual apparatus provided by formal language theory. We use the sub-categorization described in Chomsky (1956) in order to classify formal languages; which is known as the Chomsky hierarchy (Table 4.1).

Table 4.1. Chomsky Hierarchy

Type	Language	Automaton	Grammar
0	recursively enumerable	Turing machine	unrestricted
1	context-sensitive	non-deterministic Turing machine	context-sensitive
2	context-free	non-deterministic push-down automata	context-free
3	regular	finite state automata	regular

All language classifications in Table 4.1 is a proper superset of another class which is at a hierarchically lower position. In this manner, a Type 0 language is a proper superset and a Type 3 language is a proper subset of other classes.

While trying to formalize the discourse structure, as a principle, we follow Occam's razor. In other words, we try to describe it with the least adequate formal power. Before making a discussion on discourse structure, we mention the properties and generative capabilities of formal classes of languages.

Since we have a lexicalized approach to discourse structure, we consider discourse as a system of symbols. These symbols are strings of connectives and their arguments. While trying to formalize this system, we benefit from the findings and discussions of a well-studied environment of natural language syntax. We can use the implications of the research on natural language formalization. Therefore, the next section is devoted to the formal descriptions of two natural language grammars which would be helpful in further discoussions of this chapter.

4.1 Review of Formal Grammars

4.1.1 Head Grammars (HG)

HG can be considered as generalized context-free grammar to which *wrapping* operation is added. Wrapping operation allows capturing discontinuous constituents in a language. The notion of *head* is introduced in HG. HGs are string manipulation systems in which each string is associated with a *head.* In a formal way, HG can be described as a 4-tuple G such that $G = (V_N, V_T, S, P)$ where

- V_N denotes the non-terminal alphabet,
- V_T denotes the terminal alphabet,
- S denotes the sentence symbol in V_N,
- P is the finite set of production rules of the form either $A \rightarrow f(\alpha_1, ..., \alpha_n)$ or $A \rightarrow \alpha_1$ where $A \epsilon V_N$, α_i is a non-terminal or a string with a head and f is the function of concatenation or wrapping.

4.1.2 Tree Adjoining Grammars (TAG)

A TAG is a 5- tuple G such that $G = (V_N, V_T, S, I, A)$ where

- V_N denotes the non-terminal alphabet,
- V_T denotes the terminal alphabet,
- S denotes the sentence symbol in V_N,
- I is the finite set of initial trees,
- A is the finite set of auxiliary trees.

The internal nodes of initial trees are non-terminals from V_N, leaf nodes either are terminals from V_T or empty string ε. The root of these trees is the start symbol S. Auxiliary trees have non-terminals in their internal-nodes and root. One of the leaf nodes are labelled with a non-terminal which is the same as the root; the other leaves contain either a terminal or the empty string.

The description of a TAG shows that TAG's derivation rules generate trees. Tree generation is realized by the application of adjoining rules on tree structures called elementary trees. Formally, elementary trees can be described as $I \cup A$.

4.2 Weak Equivalence

A grammar is said to be *weakly* adequate for representing a language if it captures all and only strings of the language. Correspondingly, two grammar formalisms are said to be weakly equivalent if they capture the strings of same languages. The grammar formalisms we mentioned above (HG and TAG) are said to be weakly equivalent. The equivalence of these formalisms is proved by Joshi et al. (1991),Weir (1988) and Vijay-Shanker & Weir (1994). The equivalency class of these formalisms can be named as Mildly Context Sensitive Grammars (MCSG) (Joshi, 1985). Joshi argues that an MCSG has necessary and sufficient formal power to describe natural languages. The assertion that an MCSG can capture the syntax of natural languages bases on three characteristic properties of this class of grammars:

i. Mildly context sensitive languages are parsable in polynomial time,

ii. they have a constant growth property, and

iii. only certain kind of dependencies can be captured by MSCG. These dependency types are those that observed in natural languages. We mentioned these dependency constructions in section 4.4.

4.3 Strong Equivalence

A grammar is said to be *strongly* adequate, if it describes the semantic structure of captured strings (Steedman, 2000). The relationship among the structural descriptions of the formalisms mentioned in previous sections is another research area in computational linguistics. Deep investigation of derivation processes of natural language formalisms, such as HG, TAG and CCG (Steedman, 2000), shows that these processes are realized context freely. The functions defined over these formalisms (manipulation of strings or trees) share certain properties:

- These functions are size-preserving; they do not *erase* or *copy* the structures they manipulate.

- The function operations can be applied in a derivation regardless of the context.

These grammars can be classified as Linear Context-Free Rewriting Systems with respect to the structural description of their derivation processes. This classification provides a theoretical base for investigating how the languages generated by these grammars have constant-growth property and how these languages can be recognized in polynomial time.

4.4 Language as a Formal System

Descriptive language studies show that context-free grammars are not adequate to capture certain natural language aspects. The most powerful arguments are coming from Dutch (Bresnan et al., 1982) and Swiss-German (Shieber, 1985). These languages involve certain kind of crossing-dependencies. Before making a discussion on these types of crossing-dependencies, we make a review of dependencies in general.

In a formal system dependencies can be nested as in Figure 4.1.

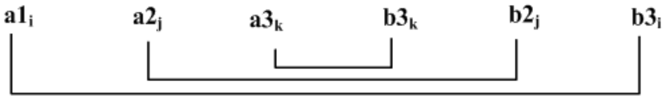


Figure 4.1. Nested Dependencies between 2 units

Nested dependencies can be characterized by context-free grammars.

Another dependency type is cross-serial dependency which is illustrated in Figure 4.2.

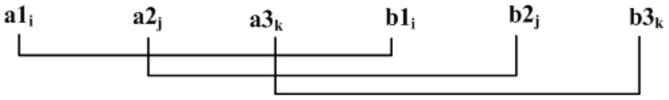


Figure 4.2. Crossing Dependencies between 2 units

The language L_1 having that kind of dependency can simply be defined as

(11) $L_1 = \{a^n b^n | n \geq 1\}$

This language is context-free but context-free grammars are not strongly adequate to describe cross-serial structures.

We just examined the cross-serial constructions whose dependency sets contain only two elements, but in some cases more elements can be dependent as in Figure 4.3.

These dependency types are involved in such a language

(12) $L_1 = \{a^n b^n c^n | n \geq 1\}$

These languages can not be generated by context-free grammars, they belong to context-sensitive class in Chomsky hierarchy.

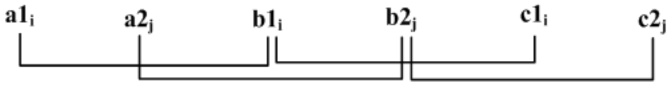


Figure 4.3. Crossing Dependencies between 3 units.

If we return to the assertion that MCSGs can describe natural languages formally (section 4.2), then it is espected that MCSGs can capture the dependency types natural languages possess. MCSGs can capture the dependencies illustrated in Figure 4.1 and Figure 4.2, but not the ones the MIX language (Bach, 1974) possess which are exemplified in Figure 4.3. In MIX there are types of strings which include a collection of letters each having an equal number of occurrences in any order. MIX is a context sensitive language and can not be captured by MCSG because of the dependency types it involves.

Since natural language syntax displays the types of dependencies illustrated in Figure 4.1 and Figure 4.2, any linguistic formalism should capture such dependencies. The following example is from Turkish showing that natural languages have nested dependencies in syntax:

(13) Ruşen$_i$ Ayşe'nin$_j$ geldiğini$_j$ sanmış$_i$. [1] (constructed)

Ruşen thought that Ayşe had come.

Dutch and Swiss-German grammars have crossing-serial dependencies. As mentioned above, this property of syntax constitutes the argument that context-free grammars are not adequate formalisms for describing natural languages. The following example is taken from Bresnan et al. (1982) to demonstrate the crossing-dependency in Dutch.

(14)

a. Jan Piet Marie zag helpen zwemmen.

b. Jan$_i$ Piet$_j$ Marie$_k$ saw$_i$ help$_j$ swim$_k$

Both the nested construction in (13) and cross-serial dependency construction in (14) are unbounded because there is no theoretical limit on the number of embeddings in each case. In section 4.5, we make a brief discussion on discourse structure.

[1] The strings with the same subscript symbol constitue a dependency set.

4.5 Formal Properties of Discourse

Lee et al. (2006) state that there exist cross-serial dependency constructions between discourse connectives and their arguments in English. They do not have an assertion such that these constructions are unbounded. Therefore, they do not argue that these constructions have an impact on the computational resources required to parse discourse structure. They investigate whether these structures can be considered as the possible departures of discourse structure from tree representation. This study shows that the structural vs. anaphoric distinction (Webber et al., 2003) of the connectives simplifies the structural description of discourse. Because, the investigation of the crossing dependencies displays that the anaphoric resolution of the first argument of adverbials causes crossing dependencies. Since the cross-serial dependencies in English discourse are not structural, these kind of dependencies can be factored out in the description of syntactic structure of discourse.

Our examination of dependency constructions in Turkish discourse shows that crossing dependency structures exist in Turkish discourse as well (section 3.7). We represent this case by a manually constructed example (9). In that example, there is a kind of bounded dependency, hence it can be captured even with a finite language. The questions “do these kind of dependencies have structural base? and if that is the case, does Turkish discourse include unbounded crossing dependencies” are still open and should be investigated in further studies.

CHAPTER 5

DATA REPRESENTATION AND SPECIFICATION OF ANNOTATION ENVIRONMENT

An important aspect of discourse understanding is figuring out the coherence relations it involves. Discourse annotation projects in general, and TDAP in specific, aim to generate a data source which can be used in the studies of the investigation of the nature of coherence relations. Since the final product of TDAP is intended to be as theory-neutral as possible, it is necessary to encode all the relations in the same way, regardless of the features of the relation elements.

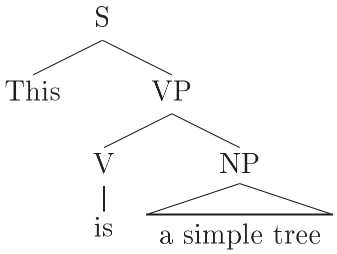
TDAP also aims to provide a large scale annotated corpora for a variety of applications operating on different fields of natural language processing. In order to achieve this, it is necessary to enrich the annotations. Therefore, the features selected as necessary for the annotation process have great importance. Since there is no theory independent definition of the relation notion, a completely theory-neutral annotation scheme is not possible. Nevertheless, the ultimate goal of TDAP is to produce annotated data which is to a large extent theory-neutral. An annotation tool which is designed by taking into account well-considered cases can handle all the constructions encountered in such an annotation process. In order to gather the requirements for such an annotation tool, it is necessary to investigate the internal structure of discourse and the relations between the connectives and their arguments.

5.1 Data Representation

We implemented a stand-off markup technique rather than an inline method in data structure design. Inline annotation can be described as an embedded annotation technique in which annotations are put in the same file with the original data source. At the beginning, we were considering to implement an XML-based inline annotation environment. However, we changed our mind after the deep investigation of the data structures we encountered. This section briefly introduces the reasons that prevent us from using an inline method.

The term XML stands for eXtensible Markup Language. It was designed to standardize data sharing over different applications. XML users can create new elements specific to their applications, therefore the language is extensible.

XML can represent any data which can be described by a *tree* structure. *Tree,* in computer science, is a data structure used to represent hierarchical architectures. Trees consist of one root and linked nodes. Each node has exactly one parent; a parent node may have several child nodes. A typical tree structure is as follows:



The elements of an XML file is defined by tags as in the example below:

<book> This is a book...</book>

Since XML, as a data representation standard, found acceptance around all over the world for last decade, many software libraries are developed for XML processing. Existence of useful libraries provide software developers a compact and easy to use framework.

We explored connective-argument dependencies in Turkish discourse to investigate whether inline XML-based markup is suitable for our purposes. In chapter 3 we presented the dependency constructions we encountered. Among these constructions, those that are introduced in section 3.3, section 3.4, and section 3.7 are considered as the violations of tree structure: “Shared Argument” construction implies a unique node with multiple parents. The construction of “Properly Contained Relation” violates the syntax of tree representation as well because it implies overlapping in dependency structures. Lastly, in the ”Pure Crossing” case, it is necessary to associate non-adjacent nodes which is not possible to represent by a straightforward implementation of trees.

Apart from these non-tree-like dependency constructions, we also encountered discontinuous text spans of arguments which also generate the relations that are not suitable for tree representation. In (15), we see an instance of the argument span discontinuity.

(15) *Yürü lan, dedi Katana,* **Ramiz'i kolundan** çekerek, *Miskoye korkuyo!*

“Hey you, move” said Katana, dragging Ramiz by the arm, “Miskoye is freaked out”

Conn	Arg1	Arg2
-erek	Yürü ... Katana, Miskoye korkuyo	Ramiz'i ... çekerek

In (15), the Arg1 of the connective -erek is interleaved with the second argument Arg2.

There exist proposed algorithms for XML to deal with such kind of problematic cases. The common characteristic of these approaches is that they divide the complex schema into smaller and simpler schemas. The aim is to use XML's physical structure with no conflicts.

The following example is taken from Dipper (2005). It simply represents a conflicting hierarchy:

```
<chunk id="ch 1"> syntactic content ...
          <pros id="pros 1">
                  prosodic/syntactic content ...
</chunk >
                  prosodic content ...
           <pros/>
```

Following approaches are presented in Sperberg-McQueen & Huitfeldt (2000) as methods dealing with conflicting hierarchies in XML:

- CONCUR in SGML

  CONCUR option in SGML allows a document to include concurrent hierarchical structures. CONCUR feature is added to SGML in order to overcome overlapping problem. This feature is specific to SGML and can not be implemented in XML. In an SGML document, if the CONCUR option is ON, then it is possible to define multiple hierarchies for the same data source. This can be achieved by creating a document type definition (DTD) for each hierarchy.

  Although CONCUR feature can be useful theoretically, since it is not supported by most of standard SGML libraries, it makes too complex the parsing of the documents. The management of the data (querying, storing etc.) is possible if the application-specific libraries are developed.

- Milestone Elements

  The start and end point of conflicting elements are marked by empty elements. This representation marks an alternative ghost tree with empty elements.

```
<chunk id="ch 1"> syntactic content ...
        <pros start id="pros 1a"/>
              prosodic/syntactic content ...
</chunk >
              prosodic content ...
         <pros end id="pros 1b"/>
```

The problem with milestone approach is that XML-based technologies like XPath and XSLT can not deal with the free texts between milestone elements. The documents which contain free texts necessitate an extra effort to query. In addition to this, semantic validation is impossible for such kind of documents.

- Fragmentation

  The element considered as less important is fragmented into smaller units.

  ```
  <chunk id="ch 1"> syntactic content ...
          <pros start id="pros 1a" next="pros 1b"/>
                prosodic/syntactic content ...
          </pros>
  </chunk >
           <pros id="pros 1b" prev="pros 1a">
                prosodic content ...
           <pros/>
  ```

  The fragmented tags virtually come together in order to represent compact data, therefore *merge* operation should be defined for each type of fragmented information.

- Stand-off Annotation

  Stand-off markup can be described as storing annotations independent from the data, i.e. annotations are put into a different file. Since encoded information is not embedded into the orginal data file, it is necessary to associate data source with this information. Stand-off annotation is a kind of redundant encoding, because the same data source can be encoded in different files with different levels of hierarchies.

We prefer to implement a stand-off annotation technique in our implementation. In our usage of stand-off annotation, the text spans of annotations are stored in terms of their character offsets. The drawback of this technique is that if the original file is changed than previously annotated data will be meaningless. Therefore we need to finalize our primary source data before the beginning of annotation process.

5.2 Software Requirements of the Annotation Environment

We aim to develop a software environment not only used in annotation process but also can be used in further analysis of annotated data. The requirements we determined are listed below:

We expect the tool to

1. allow the annotation of discourse relation elements, i.e. discourse connectives and their argument spans.

   (16) *Ortada hiçbir ipucu yok.* Çünkü **öldürülen yok.**

   There is no clue arround. Because **there is no one killed.**

2. allow the annotation of connective modifiers and supplementary arguments.

   (17)a. Sup1 Annotation: *Koşsam gücüm yeter miydi?* ***Nefesimi sonuna dek bıraksam havaya! Sıyırıp atabilir miydim yaşadıklarımın tortusunu üzerimden?*** Ya da **koşmak , kaçmak çare miydi kurtulmaya?** [1]

   If I had run, could I succeed? If I have exhaled all my breath! Could I cast off the residue of the things I have lived. Or, are running, escaping a way to be free?

   b. Sup2 Annotation: [..] *varolan yasalara göre suçlu muyduk , değil miydik?* Ya da **tersinden alalım:** ***suçsuzluğumuzu, varolan yasalara göre mi savunacaktık, yoksa toplumun gelişmesine göre mi?*** [2]

   According to existing laws, were we guilty or not? Or let's take obversely, would we have defend ourselves in respect of existing laws or development of the society?

   c. Modifier Annotation: Albert Camus'nün "İdam" adlı kitabında anlattığı gibi, *idam cezası caydırıcı olmaz.* **Tam** aksine, **"facia" ve "martir" duyguları, militan hareketlerde ölümü göze alan yeni eylemciler yaratabilir.** [3]

   As Albert Camus told in his book "Execution", death penalty can not be dissuasive. On the contrary, feelings of disaster and martir can create new activists who can risk their lives in militant actions.

[1] Sup1 is both in italics and boldface

[2] Sup2 is both in italics and boldface

[3] Modifier is in boldface and underlined

3. allow the addition of new markable features in order to enrich the annotated data.

4. allow the definition of implicit connectives and annotation of their elements.

5. allow the entering of grammatical class that the connective belongs to. This feature can be used to observe the differences in the cases where the same string span of connective behaves differently:

   (18) Bu değer yargılarının önemli bir kaynağı ise dindir . **Dolayısıyla** din , sanat hayatında geliştirici veya engelleyici olabilir .

   One of the important source of these value judgments is religion. Therefore, religion can be improving or frustrating in the art life.

   (19) 3 saat 52 dakikalık bir film olması **dolayısıyla** da o günlerin en uzun filmi özeliğini taşımaktaydı .

   It had the peculiarity of being the longest film of those days due to its 3 hours 52 minutes length.

   From the discourse perspective, in these examples "dolayısıyla" functions as if it is an adverbial in (18) and it is a subordinator in (19). The annotation of this information is necessary for the investigation of the behaviors of connectives.

6. allow the entering of grammatical classes that the arguments belong to. This feature can be used to investigate argument-hood notion in further studies.

7. allow the marking of the sense of the connective. Sense of a connective describes how its arguments are semantically related. Discourse connectives can have more than one meaning. Since to get the correct semantic interpretation of the relation, we need to get the correct sense of the connective. In this respect, the annotation of the sense of the connective is an indispensable need. The examples

   (20) Sokakta birlikte olmak için sandalyemi itmen gerekiyordu. ("için" has the meaning of "so as to")

   You should push my chair so as to be together on the street.

   (21) Tüm gücünü kullandığı için ter içinde kalmıştı. ("için" has a causal meaning)

   She was in a lather because of the fact that she went all out.

8. allow the annotation attribution information. Attribution can be defined as the determination of "who has expressed each argument to a discourse connective (the writer or some other speaker or author) and who has expressed the discourse relation itself" (Dinesh et al., 2005).

   Since Turkish verbs have a morphological agreement arker which guarantees subject-verb agreement, attribution annotation can be an easier task for Turkish. For instance in the following example, the verb of main clause "belirttiler" displays agreement with a plural 3rd person subject:

   (22) *Silah denetçilerinin ve BM Güvenlik Konseyi ' nin beklenmesi gerektiğini*, <u>aksi halde</u> **Amerika'nın savaş için meşruiyetten yoksun kalacağını** belirttiler.

   For (22), we can say that the arguments of <u>aksi halde</u> are expressed by a plural 3rd person subject - *they.*[4]

9. allow the observation of inter-annotator agreement; agreed and disagreed parts should be discriminated. It will be good if we can measure the agreement results by using statistical methods such as Kappa statistics.

10. allow the querying of annotated information. The tool should be able to display the arguments for selected connective, the overlapping segments in the discourse etc.

11. allow the selection of overlapping text spans for different connectives, the selection of discontinuous segments as connectives and arguments. In addition to these, the tool should also allow crossing-dependencies. We introduce these cases in the previous section and discuss how we represent the annotation data in order to handle such situations.

[4] In Turkish, plural subjects can be used to refer to a single person because of the pragmatic reasons but we ignore these usages in this discussion.

CHAPTER 6

CONCLUSION

In this thesis, we examined the dependency structures of discourse relations on the ground of a lexically based theory. The lexical account we adopted asserts that discourse relations are set up by lexical items called discourse connectives. These connectives are discourse-level predicates and take two discourse units as their arguments.

As a product of this thesis, we have implemented an annotation environment to be used in an ongoing discourse level annotation project(TDAP). We modeled the data representation of this software by following a data-driven methodology. In section 3.3, section 3.4, and section 3.7, we showed that Turkish discourse involves non-tree-like dependency constructions. The existence of such constructions lead us to use a stand-off annotation markup instead of an inline annotation technique.

In chapter 4, we discussed the formal aspects of discourse structure. We present that the capturing of crossing dependencies in natural language data requires more computational power than context free grammars have. The cross-serial dependencies in syntax are unbounded. The example (section 3.7) we presented as an instance of the crossing dependency construction in Turkish discourse is a kind of bounded dependency. Since bounded constructions can be captured even with finite languages, we should investigate whether discourse has such kind of unbounded dependencies. Therefore, the questions "do these kind of dependencies have structural base? and if that is the case, does Turkish discourse include unbounded crossing dependencies" are still open and should be addressed in further studies.

REFERENCES

Asher, N. (1993). *Reference to Abstract Objects in Discourse.* Kluwer, Dordrecht.

Atalay, N. B., Oflazer, K., & Say, B. (2003). The annotation process in the turkish treebank. In *Proceedings of 11th Conference of the EACL-4thLinguistically Interpreted Corpora Workshop- LINC.* Hungary.

Bach, E. (1974). *Syntactic Theory.* Horn:Rinehart and Winston, Inc.

Bresnan, J., Kaplan, R., Peters, A., & Zaenen, A. (1982). Cross serial dependencies in dutch. *Linguistic Inquiry*, 13, 613–635.

Carlson, L., Marcu, D., & Okurowski, M. E. (2003). Building a discourse-tagged corpus in the framework of rhetorical structure theory. In van Kuppevelt, J. & Smith, R., editors, *Current Directions in Discourse and Dialogue.* Kluwer Academic Publishers.

Carrell, P. (1982). Cohesion is not coherence. *TESOL quarterly*, 16, 479–488.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, 2, 113–124.

Dinesh, N., Lee, A., Miltsakaki, E., Prasad, R., Joshi, A., & Webber, B. (2005). Attribution and the (non-)alignment of syntactic and discourse arguments of connectives. In *Proceedings of the ACL Workshop on Frontiers in Corpus Annotation II: Pie in the Sky.* Ann Arbor, Michigan.

Dipper, S. (2005). Xml-based stand-off representation and exploitation of multi-level linguistic annotation. In Eckstein, R. & Tolksdorf, R., editors, *Berliner XML Tage*, pp. 39–50. Kluwer Academic Publishers.

Forbes-Riley, K., Miltsakaki, E., R.Prasad, Sarkar, A., Joshi, A., & Webber, B. (2003). D-ltag system: Discourse parsing with a lexicalized tree-adjoining grammar. *Journal of Logic, Language and Information, Special Issue: Discourse and Information Structure*, 12.

Halliday, M. A. K. & Hasan, R. (1976). *Cohesion in English*. London: Longman.

Joshi, A. (1985). How much context-sensitivity is required to provide reasonable structural descriptions: Tree adjoining grammars. In D. Dowty, L. K. & Zwicky, A., editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pp. 206–250. Cambridge University Press, Cambridge.

Joshi, A., Vijay-Shanker, K., & Weir, D. (1991). The convergence of mildly context-sensitive grammar formalisms. In Sells, P., Shieber, S., & Wasow, T., editors, *Foundational Issues in Natural Language Processing*. The MIT Press.

Lascarides, A. & Asher, N. (1991). Discourse relations and defeasible knowledge. In *Proceedings to the 29th Annual Meeting of the Association of Computational Linguistics (ACL91)*, pp. 55–63. Berkeley, USA.

Lascarides, A. & Asher, N. (1993). Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and Philosophy*, 16, 437–493.

Lascarides, A. & Asher, N. (2007). Segmented discourse representation theory: Dynamic semantics with discourse structure. In Bunt, H. & Muskens, R., editors, *Computing Meaning*, 3, pp. 87–124. Kluwer Academic Publishers.

Lee, A., Prasad, R., Joshi, A., Dinesh, N., & Webber, B. (2006). Complexity of dependencies in discourse. In *Proceedings of the 5th International Workshop on Treebanks and Linguistic Theories*. Prague.

Mann, W. & Thompson, S. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8, 243–281.

Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a large scale annotated corpus of english: The penn treebank. *Computational Linguistics*, 19, 313–330.

Miltsakaki, E., Prasad, R., Joshi, A., & Webber, B. (2004). The penn discourse treebank. In *LREC*. Lisbon, Portugal.

Polanyi, L. (1996). The linguistic structure of discourse. Technical report, Stanford CA:CSLI.

Say, B., Zeyrek, D., Oflazer, K., & Ozge, U. (2002). Development of a corpus and a treebank for present-day written turkish. In *11th International Conference on Turkish Linguistics*.

Shieber, S. M. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8, 333–333.

Sperberg-McQueen, C. M. & Huitfeldt, C. (2000). Goddag: A data structure for overlapping hierarchies. In *DDEP/PODDP*, pp. 139–160.

Steedman, M. (2000). *The syntactic process*. MIT Press, Cambridge, MA, USA.

Vijay-Shanker, K. & Weir, D. (1994). The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory*, pp. 511–546.

Webber, B. (2004). D-ltag: Extending lexicalized tag to discourse. *Cognitive Science*, 28, 751–779.

Webber, B. (2006). Accounting for discourse relations: Constituency and dependency. In Butt, M., Dalrymple, M., & King, T., editors, *Intelligent Linguistic Architectures*, pp. 339–360. Stanford: CSLI Publications,.

Webber, B., Joshi, A., Stone, M., & Knott, A. (2003). Anaphora and discourse structure. *Computational Linguistics*, 29(4).

Weir, D. (1988). *Characterizing mildly context-sensitive grammar formalisms*. PhD thesis, University of Pennsylvania, Philadelphia.

Wolf, F. (2005). *Coherence in natural language: Data Structures and Applications*. PhD thesis, Massachusetts Institute of Technology.

Wolf, F. & Gibson, E. (2005). Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31, 249–287.

Wolf, F., Gibson, E., Fisher, A., & Knight, M. (2003). A procedure for collecting a database of texts annotated with coherence relations. *Unpublished Manuscript*.

Zeyrek, D. & Webber, B. (2008). A discourse resource for turkish: Annotating discourse connectives in the metu corpus. In *Proceedings of IJCNLP*.

APPENDIX

USER MANUAL FOR THE ANNOTATION TOOL

To begin with the introduction to the software, it is good to know the basic technologies used in the application, which are, Xerces XML parser and Lucene Search Library.

Getting to Know

Xerces XML Parser

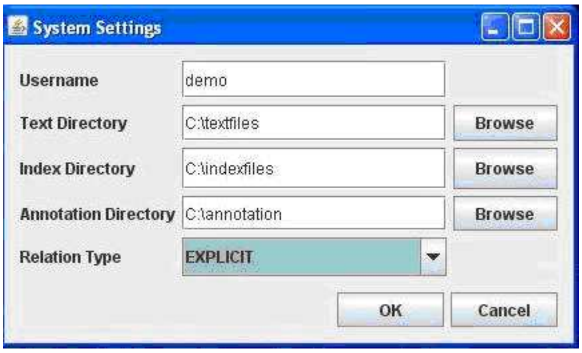
Xerces XML parser is used in a variety of applications to access and maintain XML data. It is a portable platform that enables an application to load and store XML data in a meaningful manner. It supports different application programming interfaces (API) like DOM and SAX. Xerces now supports most XML standards starting from "XML 1.0" and is enriched to recognize many related versions. The parser basically helps parsing, updating and creating XML files for programs using them as data. The universality of the application makes it easy to take part in business projects.

Lucene Search Library

In principle, Lucene is a library dedicated to serve as a search tool in text-based applications. The main idea behind this search library is to create an index and search for the keywords in this index instead of all files. This approach speeds up the process, as and index is easier to handle. In other words, the new search is held in a word-based behaviour rather than page-based. Therefore, an index is built prior to any search, and queries are handled via an IndexSearcher, returning the hit situations in either one file or more files. Lucene has also its own language for making searches, allowing the annotator to concentrate on some parts when searching as well as performing a basic level of logic operations.

Executing the Software

The execution of the program is followed by a login screen, with fields named as "Username", "Text Directory", "Index Directory", and "Annotation Directory". Username specifies the annotator's username. The files are kept in text directory, their index is created by the program at index directory; and annotations are saved in annotation directory. The "Relation Type" allows the annotator to choose from 5 types of relations that the program can perform.



System Settings

Username: demo

Text Directory: C:\textfiles

Index Directory: C:\indexfiles

Annotation Directory: C:\annotation

Relation Type: EXPLICIT

Figure A.1. Login Screen

Creating Index and Making Searches

Before going into searching and creating annotations, it is necessary to index the files to be worked on. For this, the annotator can select "Index Files" tab under the "Tools" menu. So the program indexes the files in accordance with Lucine library, and puts the index file in its destination. The next menu, "View", has three options. One is Displaying function frame, where the annotator can make the annotations and save them. The next two are used to increase and decrease font size of the file shown in the main frame (the big one in the middle).

The text field on the top-left is keyword area, the connective to be searched for is entered there. After clicking on the "Search" button, the program brings the files where there is a hit situation, and lists them on the left side of the screen. This is where the annotator clicks on one of the files, and the contents of the file is shown in the middle text area.

The "Highlight" button is used to highlight the connective currently looked for with a red colour. The annotator can remove highlights by clicking on the button again, which now reads "Remove HL".

Making Annotations

To create an annotation, we specify at least three parts, connective, argument 1 and argument 2. The rest are up to the annotator, connective modifier, supplementary argument 1 and supplementary argument 2 are optional. How is a word specified? For this purpose, the the word(s) are selected by dragging the Mouse from the beginning to the end after clicking on either lead. Then, the selected field is highlighted, and by clicking on the type, we mark a token. Others are dealt with in a similar way, and after the annotation is done, the annotator can save it by clicking on "Add Annotation" button on the right. The session can be saved using "Save Annotations" button.

"Clear List" button clears the current annotation list; however, the annotations are not deleted and the session can be opened once again if it is saved before.

Saved annotations related to a file are shown on the bottom-right corner,

- EXPLICIT-ve

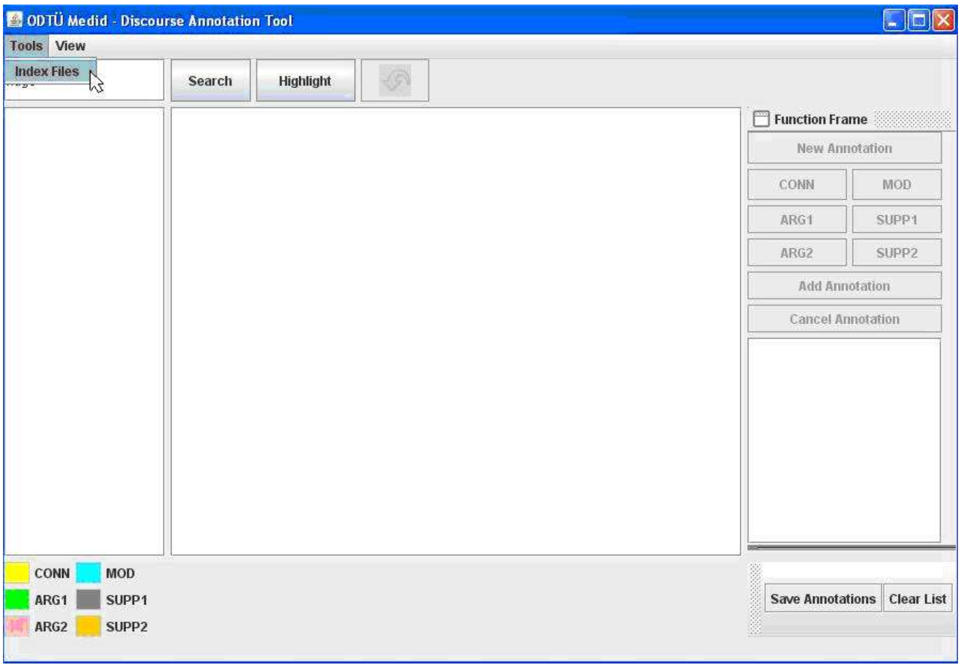


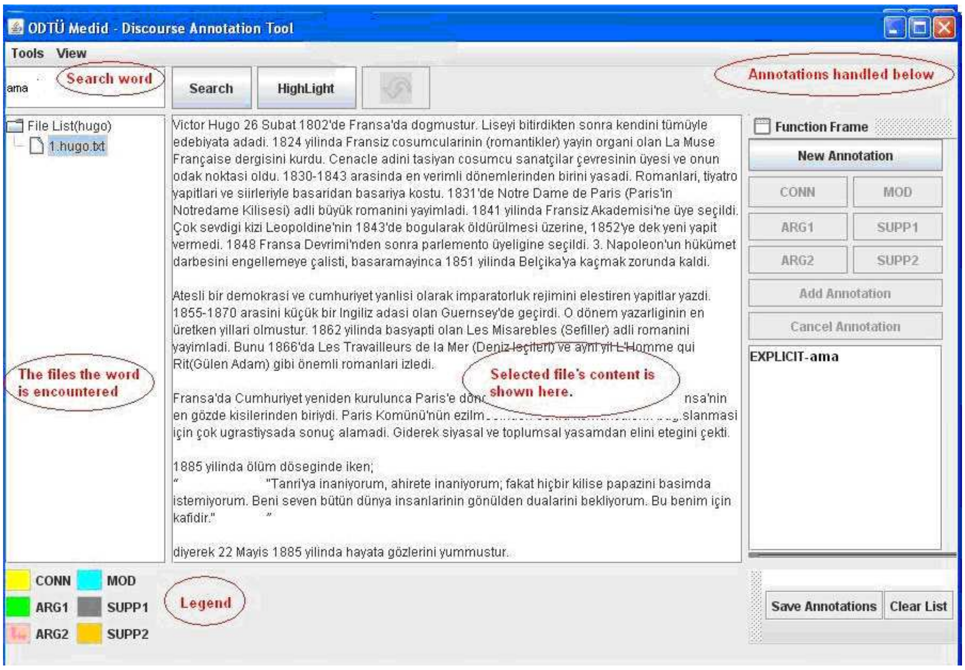
Figure A.2. Indexing the Files

- EXPLICIT-lakin

can be two examples of annotations associated with a file. Double-clicking on one of the annotations results in a list of three options. The annotator can show annotation highlights as they are specified before, remove annotation highlights, or remove the annotation itself. If the current work is not saved, and the annotator wants to work in a different file, the program asks if it should save the current annotations.

System Requirements

Java Runtime Environment 1.6 is required and can be downloaded from: http://java.com/en/download/manual.jsp



Search word

Annotations handled below

The files the word is encountered

Selected file's content is shown here.

Legend

Victor Hugo 26 Subat 1802'de Fransa'da dogmustur. Liseyi bitirdikten sonra kendini tümüyle edebiyata adadi. 1824 yilinda Fransiz cosumcularinin (romantikler) yayin organi olan La Muse Française dergisini kurdu. Cenacle adini tasiyan cosumcu sanatçilar çevresinin üyesi ve onun odak noktasi oldu. 1830-1843 arasinda en verimli dönemlerinden birini yasadi. Romanlari, tiyatro yapitlari ve siirleriyle basaridan basariya kostu. 1831'de Notre Dame de Paris (Paris'in Notredame Kilisesi) adli büyük romanini yayimladi. 1841 yilinda Fransiz Akademisi'ne üye seçildi. Çok sevdigi kizi Leopoldine'nin 1843'de bogularak öldürülmesi üzerine, 1852'ye dek yeni yapit vermedi. 1848 Fransa Devrimi'nden sonra parlemento üyeligine seçildi. 3. Napoleon'un hükümet darbesini engellemeye çalisti, basaramayinca 1851 yilinda Belçika'ya kaçmak zorunda kaldi.

Atesli bir demokrasi ve cumhuriyet yanlisi olarak imparatorluk rejimini elestiren yapitlar yazdi. 1855-1870 arasini küçük bir Ingiliz adasi olan Guernsey'de geçirdi. O dönem yazarliginin en üretken yillari olmustur. 1862 yilinda basyapti olan Les Misarebles (Sefiller) adli romanini yayimladi. Bunu 1866'da Les Travailleurs de la Mer (Deniz Isçileri) ve ayni yil L'Homme qui Rit(Gülen Adam) gibi önemli romanlari izledi.

Fransa'da Cumhuriyet yeniden kurulunca Paris'e dön nsa'nin en gözde kisilerinden biriydi. Paris Komünü'nün ezilm slanmasi için çok ugrastiysada sonuç alamadi. Giderek siyasal ve toplumsal yasamdan elini etegini çekti.

1885 yilinda ölüm döseginde iken;
" "Tanri'ya inaniyorum, ahirete inaniyorum; fakat hiçbir kilise papazini basimda istemiyorum. Beni seven bütün dünya insanlarinin gönülden dualarini bekliyorum. Bu benim için kafidir." "

diyerek 22 Mayis 1885 yilinda hayata gözlerini yummustur.

Figure A.3. Overview