Department of
Information Engineering
and Computer Science DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive, 14 - 38123 POVO, Trento - Italy
http://disi.unitn.it

End-to-End Discourse Parse using Cascaded Structured Prediction

Sucheta Ghosh

April 2012

PhD Dissertation # DISI-09-XXX

Abstract

Parsing discourse is a challenging natural language processing task. In this research work first we take a data driven approach to identify arguments of explicit discourse connectives. In contrast to previous work we do not make any assumptions on the span of arguments and consider parsing as a token-level sequence labeling task. We design the argument segmentation task as a cascade of decisions based on conditional random fields (CRFs). We train the CRFs on lexical, syntactic and semantic features extracted from the Penn Discourse Treebank and evaluate feature combinations on the commonly used test split. We show that the best combination of features includes syntactic and semantic features. The comparative error analysis investigates the performance variability over connective types and argument positions. We also compare the results of cascaded pipeline with a non-cascaded structured prediction setting that shows us definitely the cascaded structured prediction is a better performing method for discourse parsing.

We present a novel end-to-end discourse parser that, given a plain text document in input, identifies the discourse relations in the text, assigns them a semantic label and detects discourse arguments spans. The parsing architecture is based on a cascade of decisions supported by Conditional Random Fields (CRF). We train and evaluate three different parsers using the PDTB corpus. The three system versions are compared to evaluate their robustness with respect to deep/shallow and automatically extracted syntactic features.

Next, we describe two constraint-based methods that can be used to improve the recall of a shallow discourse parser based on conditional random field chunking. These method uses a set of natural structural constraints as well as others that follow from the annotation guidelines of the Penn Discourse Treebank. We evaluated the resulting systems on the standard test set of the PDTB and achieved a rebalancing of precision and recall with improved F-measures across the board. This was especially notable when we used evaluation metrics taking partial matches into account; for these measures, we achieved F-measure improvements of several points.

Finally, we address the problem of optimization in discourse parsing. A good model for discourse structure analysis needs to account both for local dependencies at the token-level and for global dependencies and statistics. We present techniques on using inter-sentential or sentence-level (global), data-driven, non-grammatical features in the task of parsing discourse. The parser model follows up previous approach based on using token-level (local) features with conditional random fields for shallow discourse parsing, which is lacking in structural knowledge of discourse. The parser adopts a two-stage approach where first the local constraints are applied and then global constraints are used on a reduced weighted search space (n-best). In the latter stage we experiment with different rerankers trained on the first stage n-best parses, which are generated using lexico-syntactic local features. The two-stage parser yields significant improvements over the best performing model of discourse parser on the PDTB corpus.

Contents

List of Figures

List of Tables

1 Introduction

Sphoṭa (the Sanskrit for "bursting", "opening", "spurt") is an important concept in the Indian grammatical tradition, relating to the problem of speech production. It deals with the problem that how the mind orders linguistic units into coherent discourse and meaning. This concept was introduced by Bhartṛhari, the author of the Vākyapadīya ("[treatise] on words and sentences") in the 5th century. He defined the structure of speech as a three staged structure: (1) Conceptualization by the speaker (Paśyantī "idea") (2) Performance of speaking (Madhyamā "medium") (3) Comprehension by the interpreter (Vaikharī "complete utterance"). The first one can be compared to the intentional structure of the speaker, whereas the second can be the linguistic structure of the content, and third one can be compared with the state of attention (Grosz & Sidner 1986)[3]. The state of attention contain information about objects, properties, relations and discourse intention that are most salient at any given point. It is an abstraction of the focus of attention of the discourse participants; it serves to summarize information from previous utterances crucial for processing subsequent ones, thus obviating the need for keeping a complete history of the discourse between participants.

The concept of rhetorics i.e. the art of discourse was also defined by Aristotle, much before all those theories. This concept typically defines rhetorics as the provider of heuristics for understanding, discovering and developing arguments for particular situations. This concept was codified further in classical Rome.

In modern period Michel Foucault (1969)[4] has defined discourse as an entity of sequences of signs in that they are enouncements. An *enouncement* (in French l'énoncé, may be translated as "statement") is an *abstract* matter that enables signs to assign specific repeatable relations to objects, subjects and even other enouncements.

Discourse, therefore, refers to too wide spectrum of human life. The various levels or dimensions of discourse can be sounds (for example, intonation), gestures, syntax, lexicon, style, rhetorics, meanings, speech acts, moves, strategies, turns and other aspects of interaction. Here we depict discourse as a vantage point of linguistics, and especially of applied linguistics. We refer to the discourse as coherently related set of utterances or a text body spans across the sentences, clauses and phrasal boundaries, i.e. the clause-like structures. These are connected through the entities, eventualities and functionalities (Webber et al 2011)[5]. Thus these related units allows whole discourse to conceive a meaning beyond the concept of single utterances, or even a blind concatenation of the concepts of several utterances. Among all such kind of relations here we are interested to capture a particular class of coherence relation, i.e. the rhetorical relation in texts. Here we declare that the terms "coherent relation", "rhetorical relation", or "discourse relation" are considered as the alternative metaphor of each other, throughout this thesis-work.

Rhetorical relations are the text structuring relations between *abstract objects* (i.e. events, states, facts and propositions) (Asher 1993). We follow the same ontology structure for coherence relation as is documented by Wellner [PhD thesis 2009][6]. According to this, coherence relation can be grossly divided into reference relation (like Anaphora, bridging, hypernym), intentional relation (like evidence, motivation, justification) and informational relation (like elaboration, cause, contrast). The study of reference relation and intentional relation is out of the scope of this thesis.

We use an example from Grosz & Sidner (1986)[3] to elaborate this classification:

a. No one can deny, of course, that great educational and ethical gains may be made through the movies

b. because of their astonishing vividness.

At the informational level the cue word "because" is holding the CAUSE relation between two statements, where item (a) is holding the main statement, while item (b) is a cause of that statement. On the other hand at the intentional level a JUSTIFY (increases acceptance Mann & Thompson) relations is being hold between two statements.

There also exists a reference relation i.e. an anaphora relation inside the example described above: "their" of the item (b) is referring to the "movies of item (a).

Next, we discuss the state-of-the-art works done so far, emphasizing more on the data-driven information-relational approaches. We also document some interesting approaches, those were done successfully using intentional relations and the reference relation, related to the whole discussion.

1.1 Background

The computational treatment of discourse phenomena has been attracting attention for more than three decades, partly due to the increasing importance for potential applications, and also for their relevance in the area of semantics and pragmatics. We study in this part how the concept of discourse started forming over a period of time, then we observe how it started growing along with its nomenclature and structure. Along with we also introduce some important and recently-published research works.

Before we start the discussion, let us state some important assumptions for rest of the thesis: (1) we hereby consider our data is a *monologue*, not a dialogue. Monologues can be characterized by a type of communication that flows in only one direction say from the speaker (or the writer) to the hearer (or the reader); in case of the dialogue the communication can flow from the either sides, which is more complex in nature than a monologue [7]. Therefore in the discussion of the state-of-the-art, we will concentrate more on the methods and works with monologue. (2) All the monologue we use for this thesis work are text datasets only, more specifically we used standardized and clean text corpora.

The main aim is to understand the meaning of multi-sentential text automatically. Indeed, we all know that meaning of multi-sentential text is different from the summation of the meanings of its individual parts. In the light of this idea, we need to distinguish between two important terms: *coherence* and *cohesion*. We already defined coherence as a meaning relation between two units, whereas the cohesion is the linguistic device through which two textual units are grouped together. For example, the use of synonyms, hypernyms etc. are examples of use of lexical cohesion, whereas the use of anaphora can be one example of non-lexical cohesion inside language. Usage of cohesion in the text units does not necessarily prove the existence of coherence inside those units [7]. Let us illustrate it with a classical example as stated in Hobbs 1985[8]:

John took the train from Paris to Istambul

He likes spinach

There may exist a coherence relation between this pair of sentences, if only the context of this pair is given. Though we find the pair as coherently unrelated, at the same time we observe that there exists a non-lexical cohesive (i.e. anaphoric) relation inside “John” and “He”. So we conclude from this discussion that existence of cohesion in the text does not ensure the existence of coherence in the same text, though a coherence relation in text can be devised by cohesion.

1.1.1 Some Important Early Accounts

In this context of previous discussion, we refer to the work: “Surface-based Cohesive relations” by Halliday & Hasan (1976)[9]. This is the one earliest works that aims to describe how English texts are coherently related through explicitly established conjunctive relations. The surface relation categories were as follows: additive (i.e. parallel, elaboration), adversative (i.e. contrastive), casual and temporal; there is also available a more detailed view of the sub-types of those surface types. The main drawbacks of this framework are these: (1) neither it can view the clause-like structure of the text units, related with the conjunctions (instead it invariably considers a whole sentence as the text units), nor it gives different weights to these related text units, according to the meaning of those units. (2) Halliday and Hasan admitted that a complete account of text must make references to unmarked (such as implicit) relations, and finally (3) in this work the authors do not handle the the complex cases of language.

Now we make a canonical example from Halliday & Hasan, where we observe that this is basically a simple usage of language. The two sentences are related with a cohesive relation i.e. temporal:

First he switched on the light; Next he inserted the key into the lock.

This approach is further carried on by the work of Martin (1992)[10] that allows the clause-like structures in the text units, and it contributes implicit relations between text units. This work has been further carried out with more intricacies in the Penn Discourse TreeBank(PDTB) (described later; PDTB is our corpus of interest for this thesis). In brief, this approach gives a way not only for identifying the implicit relations in a text, but also for classifying the relations. The taxonomy of underlying (implicit) relations will basically mirror that of the devices for signaling them. This approach runs into problem with the implicit taxonomy for some categories (for example, additive) dealing with complex case; nevertheless it gives a better account of the relationship between ‘surface’ (compared to Halliday and Hasan’s model) and ‘deep’ (compared to Grim’s model[11]) relations.

In Grimes' (1975)[11] model the relations play a dual role in taxonomy: they provide information, just as clauses do, and they also organize groups of clauses into coherent discourses. The taxonomy proposed by Grimes is primarily divided into parataxies (i.e. a literary technique, in writing or speaking, that favors short and simple sentences, for example the usage of coordinating connectives in sentences rather than subordinating ones.), hypotaxies (i.e., constructs playing an unequal role in a sentence; a common example is subordination in a complex sentence.) and neutral. Following already established theories, his basic units are clauses (i.e. 'lexical predicates') and the relations between them (i.e. 'rhetorical predicates'). However, this analysis admits some exceptions. Rhetorical relationships can be found within clauses; for instance, the alternative relation, typically expressed between two or more clauses, is also found in the single clause. We pose one example here:

He saved the day; he made three touchdowns.

In this example we see that the first part of sentence is the central theme 'he saved the day', while a specification comes after. So there exists a relation of hypotaxis (SPECIFICALLY) between them.

As a summary of the discussion of these prominent early theoretical accounts we conclude that (1) only marking the existence of the relation between text units is not enough, there should be proper sense taxonomy involved to the relation (2) the structure of the relation can be graphical (surface-relational structure) or can be hierarchical (deep-relational structure) (3) text (unit) span structures are a clause-like structures.

1.1.2 Computational Theories of Discourse Relation

1.1.2.1 Hobb's Theory

Hobbs' theory (1985)[8] emphasizes the amount of contextual and external knowledge, which is required to interpret discourse. He considers the following text by way of illustration:

John took a book from the shelf. He turned to the index.

It is evident to the reader that the index referred to in the second sentence is that of the book which John has just taken; but to make this inference automatically it requires a knowledge about what indices are, and we also recollect the other example of ‘John’ and ‘Spinach’ under the section background, we posed under the background subsection.

The proposed relations are defined in terms of the different kinds of inferences which one needs to draw in order to make sense of a text.

Hobbs identifies four types of inference, and accordingly, four categories of coherence relations:

(1) a discourse can be coherent because it talks about coherent events in the world; events such that if one is known, the other one can be inferred given appropriate background knowledge. Two portions of text describing two such events are said to be linked by an *occasion* relation. This relation subdivides into relations like *cause* and *enablement.* (2) discourse coherence can be due to the fact that the speaker has some rational structure of goals in mind for producing a discourse. (3) a discourse will only be coherent if what the speaker says can be linked to the listener’s prior knowledge. (4) The fourth class of coherence relation is *expansion.* Expansion links inferences to expand discourse further: two clear examples are parallel and contrast.

Hobbs’ suggestion is basically that a tree structure of relations must exist in a text for it to be coherent. There is a recursive conception of relation. The recursive conception of relations suggests for them a procedural role in constructing large sections of text.

Hobbs advocated that a clause-like structure is a segment of discourse, and when two segments of discourse are discovered to be linked by some coherence relation, the two together thereby constitute a single segment of discourse.

With the notion of discourse structure Hobbs also discusses about some classical problems of discourse analysis: the notion of “topic”, one aspect of the notion of “genre”, and some of the deviations from coherence that occur in ordinary conversation with the light of local and global coherence.

1.1.2.2 Grosz & Sidner’s Theory

Grosz and Sidner’s (1986)[3] theory also features recursively defined relations. In this account, discourse segments (*dss*) are the principal units of structure, and relations hold between these to form larger *dss.* However, the primitives used to define relations are different from those of Hobbs: they make reference solely to the intentions a writer has in creating a text. Relations actually apply between discourse segment purposes (*dsp*s); an assumption is made that a single overriding intention can be specified for each segment, and it is these intentions which are connected by relations. The fundamental metaphor is of a text embodying the execution of a plan pursued by the speaker. A point to be noted that although Grosz and Sidner frequently use examples from task-oriented dialogues, they take care in such cases to distinguish the plan required to carry out the task from the plan required to create the text. There is more accounts on discourse plans and domain-level plans in Litman & Allen (1990)[12], also in Moore & Paris (1993)[13]

Using intentions in relation definitions follows quite naturally from thinking of them in the context of a recursive planning paradigm. Plans are produced to achieve user goals (or in the present case, writer goals); and do so by decomposing a principal goal into a hierarchy of subgoals.

There are only two relations in Grosz and Sidner's theory: dominance and satisfaction-precedence. These are the first intentionally defined relations.

Grosz and Sidner also adopt a slightly different conception of compositionality to that proposed by Hobbs. While Hobbs sees a relation between two adjacent spans as forming a new composite span, Grosz and Sidner's composite discourse segments include the segments which they dominate.

Another attractive feature of Grosz and Sidner's theory is its account of the interaction between relations and focus. Associated with every discourse segment is a focus space, and at every point in a text a focus stack is given which models the reader's focus of attention as the discourse proceeds (elements at the top of the stack being 'more salient' to the reader than elements lower down). The metaphor of a stack is another import from computational theories. Its pushes and pops are determined by the dominance relations in the text: if the segment S2 dominates a sub-segment S1, then moving into S1 causes the focus space associated with S1 to be pushed onto the stack, and leaving S1 causes it to be popped off the stack.

Grosz and Sidner's theory is the first to look in detail at relations between larger sized units of text; indeed most of their examples are of high-level relations; although the theory does not provide a very complete account of lower level relations such as those between single clauses or sentences.

1.1.2.3 Rhetorical Structure Theory

We turn now to the third computational theory i.e. rhetorical structure theory (RST). This theory, developed mainly by William Mann and Sandra Thompson, is presented in a number of papers; in this thesis we will focus on the account in Mann and Thompson (1988)[14] only.

The core constructs in RST are rhetorical relations. Text coherence is attributed principally to the presence of these relations. Unlike Grosz and

Sidner, in RST there is no visualization for an important role for any other constructs like focus.

Rhetorical relations are defined functionally, in terms of the effect the writer intends to achieve by presenting two text spans side by side. In this respect, they resemble Grosz and Sidner's relations. Although there are several differences between the two types of relation.

RST represents texts by trees whose leaves correspond to elementary discourse units (*edu*s) and whose nodes specify how these and larger units (e.g., multi-sentence segments) are linked to each other by rhetorical relations (e.g., CONTRAST, ELABORATION). Discourse units are further characterized in terms of their text importance: nuclei denote central segments, whereas satellites denote peripheral ones.

Some important features of RST:

1. RST relations make some references to the propositional content of spans, as well as to the intentions of the writer in putting them forward.

2. Mann and Thompson suggest that some types of rhetorical relations have no corresponding conjunctive signals, while Grosz and Sidner's and Hobbs suggested at least an informal link is made between underlying relations and the linguistic devices for marking them.

3. Mann and Thompson argued that the majority of text is structured using nucleus-satellite relations, although some relations (i.e. multi-nuclear), do not exhibit it. There are two multi-nuclear relations i.e. sequence and contrast.

4. RST provides a set of around 23 rhetorical relations. The top-level distinction in this taxonomy is between subject-matter and presentational relations. Subject-matter relations have as their effect that the reader recognizes the relation in question, while presentational relations have as their effect to increase some inclination in the reader.

5. RST has a strong structural account of discourse. Mann & Thompson accounted for an independent definition of 'text span': the size of the atomic units of text analysis is arbitrary, but they should have independent functional integrity. The clause is selected as the minimal unit of organization. Thus text spans are clauses, or larger units composed of clauses. Unlike Grosz and Sidner[3], relations must hold between non-overlapping text spans.

6. In RST, relations are not mapped directly onto texts. They are fitted onto structures called schema applications, and these in turn are fitted to text. Schema applications are derived from simpler structures called schemas.

7. A rhetorical structure tree is a hierarchical system of schema applications. A schema application links a number of consecutive spans, and creates a complex span.

The reason behind the popularity of RST is perhaps best attributed to a combination of features: the emphasis on a functional conception of relations, the carefully presented set of relation definitions and a simply stated structural theory. An example from RST corpus [1]:

Eg. 1. [Mr. Watkins said] 2. [volume on Interprovincials system is down about 2% since January] 3. [and is expected to fall further,] 4. [making expansion unnecessary until perhaps the mid-1990s.]

In the figure 1 a horizontal line covers a span of text (possibly made up of further spans), a vertical line signals the nucleus or nuclei, a curve represents a relation, and the direction of arrow shows the direction of the satellite towards nucleus. The arrowheads shows off the flow of spans according to the flow of the text in the document.

The spans of individual EDUs are represented at the leaves of the tree. At the root of the tree, the span covers the entire text. The path from EDU 1 to the root contains one satellite node. It is therefore assigned a penalty of 1. Paths to the root from all other EDUs involve only nucleus nodes and subsequently these EDUs do not incur any penalty.

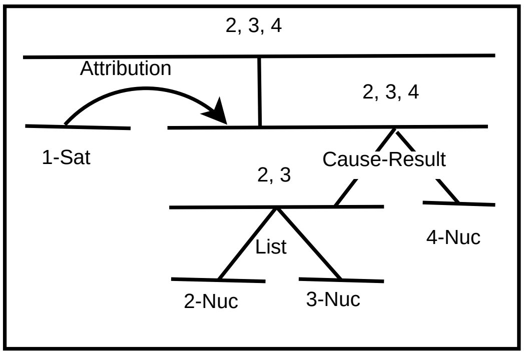

Figure 1: Annotation Example of RST corpus [1]

Even when we cannot make the use of all information that is present in text, it is worth to pay attention to a structure like RST [1].

1.1.2.4 Semantic Representation: DRT & SDRT

Discourse Representation Theory (DRT) is a formal semantic model of NLP in relation to discourse understanding of text-body. DRT was originally formulated in (Kamp, 1981) and further developed by Kamp & Reyle (1993), and also by van Eijk & Kamp (1997)[15]. DRT grew out of Montagues model-theoretic semantics (Thomason, 1974) which represents the meanings of utterances as logical forms and supports the calculation of the truth conditions of an utterance.

DRT is concerned with the semantic aspects of a discourse. These aspects are related to the meaning of the discourse, but not related to the particular situation (including time, location, common ground, etc) in which the discourse is uttered. An advantage of this approach is that this semantic representation can be automatically built up from the contents (i.e. words) and can structure the discourse alone, without bringing in information about the external context of the utterance.

In essence, DRT dynamically interprets a discourse, one sentence at a time, along the way it updates a representation of the whole discourse, which is known as a Discourse Representation Structure (DRS).

Segmented DRT or SDRT (Asher & Lascarides, 2003) combines the logic-based structures of DRT with the focus on rhetorical relations from RST to address a wide-range of discourse phenomena. SDRT greatly expands the power of the discourse update procedure by including rhetorical relations. Every time a DRS for a new utterance is added, some relation must be computed between it and one of the proceeding utterances. The set of relations, generally speaking, is open-ended.

Although RST [Mann & Thompson, 1988] and SDRT (Asher & Lascarides, 2003) are quite different theories, they both rely heavily on discourse relations and the discourse structures that can be computed from them.

[1] Centering theory (Grosz, Joshi, & Weinstein 1983, 1995) is one of such structure for dialogs. Centering theory provides yet another way of tracking the structure of discourse, by classifying *some* links between a sentence and its predecessor. The predecessor may not be the last sentence uttered; it is rather the immediate parent in the discourse tree. These links may be a backward-looking center or a forward looking center. Centering theory is the best-known framework for theorizing about local coherence and salience(Poesio et al 2004)

1.1.2.5 Linguistic Discourse Model (LDM)

The goal of discourse parsing in the LDM [Scha & Polanyi, 1988; Polanyi & Scha, 1984] is to account for discourse continuation despite discontinuity and lack of apparent coherence by specifying which contexts in the discourse history are accessible for continuation and which are not. Information at accessible contexts can provide referent anchoring for pro-nouns, pro-verbs, anaphors of all kinds and indexical expressions such as here and now which are always set relative to an Interaction taking place in some real or modeled context.

This provides a full account of discourse parsing by treating the task as an extension of sentence-level syntactic parsing. The discourse segments, roughly clauses, are well-defined in terms of standard syntactic constructions.

The Linguistic Discourse Model (LDM) [Scha & Polanyi, 1988; Polanyi & Scha, 1984] provides a full account of discourse parsing by treating the task as an extension of sentence-level syntactic parsing. The discourse segments, i.e. clause-like structures, are well-defined in terms of syntactic constructions.

LDM is similar to RST as both are purely constituency-based. The structure of output for LDM is simpler than RST as it is easier to re-write rules using coordination (for e.g., lists, narration), subordination and n-ary constructions over arbitrary rhetorical relations. Therefore RST, at time, may generalize more for all kind of problems, whereas basic LDM structure outputs more fine grained structure.

So there was a need for an Unified LDM (U-LDM) parser, where sentences are broken up in discourse relevant segment based on the syntactic information from the sentence parser. Then the segments are recombined into one or more small discourse trees, called Basic Discourse Unit (BDU) trees, representing the discourse structure of the sentence. Therefore there exists an extra compositional semantics and inferences in U-LDM to combine sentence-level and discourse level information.

1.1.2.6 Lexicalized-Grammar Based Approach

Gardent [1997] introduced feature-based Tree Adjoining grammar framework for discourse parsing. The system by Bateman (1999), KPML, also uses large-scale grammars, written with the framework of Systemic-Functional Linguistics (SFL). A prominent successor of those early approaches is D-LTAG (lexicalized Tree Adjoining Grammar for discourse). This is another important work in tradition of RST and LDM. In this case the lexicalization means that D-LTAG provides an account of how the lexical elements (even including some phrases) anchor discourse relations and how the other parts of given text provide arguments for those relations.

In contrast to LDM, D-LTAG assumes that the boundary between sentence-level structure and discourse-level structure is not a blatant one. Both the structures support compositional aspects of semantics, while allowing for other interpretive components (like anaphora) to be added on for a complete semantics.

The whole system of D-LTAG is a composit of four components: a parser that parses sentences, a tree extractor that extracts basic discourse constituent units from each sentence derivation; a tree-mapper that anchors the sentence-level elementary trees by the connective (i.e. both covert and overt ones); and finally the system outputs a "flat-structured" discourse structure. This system handles with initial derivation of a tree structure, and ultimately builds up one discourse derivational structure (i.e. an auxiliary tree). Thus it conforms with the theory of LTAG that posits two kinds of elementary trees: initial trees, which encode predicate-argument dependencies, and auxiliary trees, which are recursive and modify and/or elaborate elementary trees. Eventually, this auxiliary tree is viewed as a connected graph for the sake of understanding of the "flat" nature of the discourse structure.

1.1.2.7 GraphBank

Wolf and Gibson (2005) introduced a "relatively shallow" and recursive graphical structure of discourse spans, with discourse relation anchoring among the siblings. They specifically discuss two types of non-overlapping discourse segments: clause-like structure are the basic units of discourse segments those can be grouped under same attribution. This attribution can be made with same (sub-)topic or same event the clause-like structure share together.

It appears that groupings could determine a partial hierarchical structure for parts of a text, and that grouping is a matter of constituency. But this is the only hierarchical structure in Wolf and Gibsons approach: unlike RST and the LDM, the existence of a coherence relation between two segments does not produce a new segment that can serve as argument to another coherence relation. Moreover this theory claims to leap upward in complexity from trees to chain graphs as a model for discourse structure; whereas in PDTB which is built over the D-LTAG theory defines an ungrouped "flat" structure.

1.1.3 Discourse Algorithms

So far we were studying the early significant approaches that helped to construct present day discourse structure with taxonomy and terminologies. There is a disagreement in the literature about the definition of discourse segments (Marcu [2000]): some argue for prosodic units (Hirschberg and Nakatani 1996), others argue for intentional units (Grosz and Sidner 1986), phrasal units (Lascarides and Asher 1993; Webber et al. 1999), or sentences (Hobbs 1985). Also there is an argument about overlapping (Grosz and Sidner 1986) versus non-overlapping (Mann & Thompson, Webber et al) nature of discourse segments.

We also learned from the account discussed heretofore that each and every discourse structure holds a pattern within a multi-sentential text. It is essential to understand the pattern to compose a structure together. We found that these patterns depend on topics, eventualities (events or states) (like in GraphBank), intentional functionalities (Grosz & Sidner), or discourse relations (like conjunctions in D-LTAG)

Now we devote the following discussion to the extent of theoretically-established discourse structure, to study its complexities and properties in its present form.

1.1.3.1 Discourse Segmentation

The simplest kind of algorithm that shapes out a simple discourse structure, is discourse segmentation for a body of text. Roughly speaking, a group of "locally" coherent clauses or sentences is called a discourse segment. In this case, the "local" group implies a group of coherent clause-like structures, which occurs in the same document or in the same paragraph or within a range of a discourse marker, depending on pre-settled criteria.

In an important work of news search [16], the news text is pre-segmented with a time-interval of 15 seconds reading (roughly three sentences i.e. 50 words) for sub-topic segmentation. On the other hand, the bio-medical texts are *functionally* pre-segmented from its generation viz. Background, Method, Result, Discussion. Automatic determination of this kind of discourse structure is a not a trivial task.

We distinguish different segmentation algorithms according to the learning strategies:

1. Unsupervised segmentation: so far we were describing about the simplest kind of linear segmentations of discourse, albeit there can be more

sophisticated hierarchical segmentation of discourse structures, which we will discuss henceforth under this block.

Hearst (1997)[17] defined subtopic groupings for science news article paragraphs into some classes, which evoluted to present day templates that can be filled with proper values (more discussion coming later in this part). This work called TextTiling is basically a cohesion based approach. This is a three-step process that tokenizes the sentences then computes lexical scores between the stemmed tokens, finally it identifies the boundaries of the discourse. It segments discourse on the basis of the concepts of cohesion by Halliday and Hasan [9].

The work by Mihalcea and Tarau (2004) [18] also followed the basics by Halliday and Hasan[9]; in their work a newspaper article is segmented into subtopics of news at the primary stage using cohesion. One can also use lexical cohesion using hypernyms or other lexical relations as we notice in the work by Graesser et al.(2004) [19]; they used template-filling technique to achieve discourse structure.

Choi (2000) [20] in his seminal work ("Advances in domain independent linear text segmentation") used a divisive clustering algorithm to segment text linearly.

2. Supervised segmentation: we posed a simple example of this category in the beginning of this section [16], where the news story boundaries are segmented depending on a pre-observed timing.

   There is also a possibility of settling the rules of segmentation according to the paragraphs. In the case of annotations of implicit connectives of PDTB, covert relations are annotated between adjacent sentences within the same paragraph.

   An often-used feature for segmentation is the discourse marker or the cue word. Litman and Passonneau [21] use linguistic cues for discourse boundary segmentation. In this case, the linguistic cues are the anchor words (for example, moreover, still etc.).

1.1.3.2 Discourse Chunking

Text chunking is a supervised machine learning method. It is an intermediate step towards full parsing. Primarily the process of chunking is meant for dividing a text in syntactically correlated parts of tokens. Discourse chunking refers to text units within a discourse, such as discourse relation, which does not necessarily provide a full cover to the text. Discourse chunking can be specified as a lightweight approximate of full discourse parsing.

In machine learning terminology, discourse chunking is a method of regression with structured outputs. As far we discussed the discourse structure implies a discourse relation (the relation itself and the text tokens involved with this relation), and text spans viz. discourse arguments. Through discourse chunking we achieve either relational structure or argument structure.

Sporleder & Lapata (2005) [22] used discourse chunking for the task of sentence compression. In this work they converted each sentence-level discourse tree into a flat chunk representation by assigning each token (i.e., word or punctuation mark) a tag encoding its nuclearity status at the *edu* 1.1.2.3 level. This chunk representation is proposed by Ramshaw and Marcus (1995) to use four different tags: B-NUC and B-SAT for nucleus and satellite-initial tokens, and I-NUC and I-SAT for non-initial tokens, i.e., tokens inside a nucleus and satellite span. They represented all tokens, those belong to either to a nucleus or a satellite span. To simplify the problem of sequential tagging they did not include "O" i.e. the outside tag, to indicate elements outside a chunk. This is no-good for real-life problem as the model should learn the mapping between the discourse structure and sentence-structure, instead in this case the model is learning only to tag the discourse structure, whereas in real-life scenario the complete sentences will be the output.

Another important work is the one by Pitlar and Nenkova (2009)[23]. We know that there is token ambiguity about non-discourse and discourse usage of explicit (overt) connectives, and as well we know this is because of the uncertainties in languages itself - through the work by Pitler and Nenkova it is observed that discourse and non-discourse usage can be distinguished with about 94% of accuracy.

Therefore we observe that different discourse theories vary with respect to which markers are considered ambiguous. On the top of that, the most difficult case arises if an example does not contain an explicit discourse marker. In this situation, the rhetorical relation has to be inferred solely from the linguistic context and external knowledge. Now, it is not possible to rely on discourse markers alone to determine which rhetorical relation holds between two text spans; we need a model that can classify rhetorical relations in the absence of an explicit discourse marker. Marcu and Echihabi (2002)[24] propose a method for creating a pertinent training set automatically by labelling examples which contain an unambiguous discourse marker with the corresponding relation (i.e., the relation signaled by the marker). The discourse marker is then removed and a Naive Bayes classifier is trained on the automatically labelled data. This classifier then learns to exploit linguistic cues other than discourse markers (e.g., word co-occurrences) to determine the rhetorical relation even when no unambiguous discourse markers are present. Sporleder and Lascarides (2008)[25] applied the same theory to examples that naturally occur without a discourse marker to classify the rhetorical relation between the arguments.

There has also been some other important works recently, we will discuss them in the next chapter as a prelude to the system details.

1.1.3.3 Discourse Parsing

Full parsing of discourse deals with discourse relation and its argument classification altogether. There is also a growing number of approaches to identify the discourse arguments given the discourse connectives. A full discourse parser has been presented in the work of Soricut and Marcu (2003)[26]; we already discussed of this seminal work with respect to other works in the last section of discourse chunking. Soricut and Marcu (2003)[26] used a probabilistic model i.e. SynDS; and Marcu (2000)[27] implemented a decision tree based model i.e. DT. Soricut and Marcu (2003)[26] measure the performance of the segmenter based on the its ability to insert *intra-sentential* segment boundaries.

Baldridge & Lascarides (2005)[28] also developed a probabilistic discourse parser, making use of head-driven probabilistic parsing approaches, which were originally developed for sentence-level syntax by Collins (2003)[29]. The Redwoods corpus is an output of this system, where a set of dialogs with a set of rhetorical relations espoused by SDRT [30] is annotated. In respect of this head-based classification, we mention here that both Wellner & Pustejovsky (2007) and Elwell & Baldridge (2008) investigated and published works on head-based argument detection; we will discuss about those works in details next chapter in relation to our work.

Later in 2007, Baldridge et al.(2007)[31] presented a dependency-based approach to discourse parsing that apparently improved upon the results in Baldridge & Lascarides (2005)[28] when evaluates against the task of identifying the lexical head of each discourse argument.

There has also been a work on the application of maximum spanning tree-based dependency parsing algorithms by McDonald et al (2006)[32]. They also run a discriminative training to discourse parsing successfully.

1.1.4 Applications

We have already observed that the task of discourse processing is being used by many application areas. Now we focus on some important applications areas where discourse theories and structures became a key factor to solve the problem.

1.1.4.1 Sentiment Analysis & Opinion Mining

Much of the recent explosion in sentiment related research has focused on finding low-level features that will help predict the polarity of a phrase, sentence or text. A discourse structure or the components of discourse have been used for more than one decade by this research community.

Many application in this category used a fine to coarse level based analysis. McDonald et al (2007)[33] investigate a structured model for jointly classifying the sentiment of text at varying levels of granularity. Inference in the model is based on standard sequence classification techniques using constrained Viterbi to ensure consistent solutions. The primary advantage of such a model is that it allows classification decisions from one level in the text to influence decisions at another.

In a more recent work Zirn et al (2011)[34] studied a fully automatic framework for fine-grained sentiment analysis at sub-sentence level, combining multiple sentiment lexicons and neighborhood as well as discourse relations. They used Markov logic to integrate polarity scores from different sentiment lexicons with information about relations between neighboring segments, and evaluate the approach on product reviews.

In the work of Turney (2002)[35] the classification of a review is predicted by the average semantic orientation of the phrases in the full review that contain adjectives or adverbs computing PMI-IR metrics.

Somasundaran et al (2009)[36] tested, empirically, the impact of the discourse-level relations on fine-grained polarity classification. In this process, we also explore two different global inference models for incorporating discourse-based information to augment word-based information. The results show that the discourse-level relations can augment and improve upon word-based methods for effective fine-grained opinion polarity classification. Further, they explored linguistically motivated features and a global inference paradigm for learning the discourse-level relations form the annotated data.

1.1.4.2 Discourse in Dialogs

Much before Somasundaram, there have been several initiatives to use discourse information in dialog analysis. Grosz & Sidner's (1986)[3] seminal work, which we already discussed, concerns on the application area of dialog. Litman and Allen, 1987 presents a theory that differentiates between the different ways that an utterance can relate to a discourse plan (or plans) representing the topics of a conversation. They also define a set of discourse plans, each one corresponding to a particular way that an utterance can relate to the current discourse topic, and distinguish these plans from the set of plans that are actually used to model the topics.

The TRAINS project (Allen et al, 1995) is an effort to build a conversationally proficient planning assistant. A key part of the project is the construction of the TRAINS system, which provides the research platform for a wide range of issues in natural language understanding, mixed- initiative [2] planning systems, and representing and reasoning about time, actions and events. The theory of discourse interpretation on which the deindexing module is based is Conversation Representation Theory (CRT) (Poesio, 1994), an extension of Discourse Representation Theory (DRT) [Kamp, 1981]. CRT is designed to deal with semantic ambiguity and to allow the representation of pragmatic information present in conversations, such as the presence of multiple discourse topics and the organization of utterances in discourse segments.

In Italian, as was published by (Tonelli et al., 2010), there is a corpus of 500 Italian conversations about computer troubleshooting recorded at the help-desk facility of the Consortium for Information Systems of Piedmont Region, collected during EU project, LUNA. Within the corpus, all conversations have been segmented at the turn level and annotated with predicate-argument structures, dialogue acts, and concepts and relations drawn from a predefined domain attribute ontology. Among them about 60 dialogs are annotated following PDTB-style annotation.

1.1.4.3 Discourse in Pragmatics

A discourse particle in linguistics is a lexeme or particle which has no direct semantic meaning in the context of a sentence, having rather a pragmatic function: it serves to indicate the speaker's attitude, or to structure their

[2] Mixed-initiative interaction refers to a flexible interaction strategy in which each agent (human or computer) contributes what it is best suited at the most appropriate time. (hearst, 1999)[]

relationship to other participants in a conversation. These discourse particles are mostly a feature of spoken language; in written languages they add an informal tone.

There has been much previous work in assuming that discourse particles refer to the common ground (CG) [3], e.g., Karagjosova (2004)[38], or Zimmermann (2009)[39]. Common ground and the interlocutors [4] individual backgrounds are modeled as common or individual belief (Stalnaker, 2002)[40].

In German language there are many works, especially with the words like 'doch'[5](Egg, 2010) [41] or 'wohl'[6] [42].

1.1.4.4 Summarization

Document summarization is one of the earliest applications of discourse structure analysis. Much of the research on discourse parsing (in both the RST framework and other theories of hierarchical discourse structure) has been motivated by the prospect of applying it to summarization (Ono et al., 1994; Marcu, 1998)[43, 44].

First, we describe summarization based on a weighted hierarchical discourse structure (Marcu, 2000; Thione et al., 2004) and then review other ways in which research on discourse structure has been applied to summarization. Daume III and Marcu (2002) attempt to derive so-called informative summaries those represent the textual content of documents.

The other techniques for summarization also exploit hierarchical structure, using a flat genre-specific discourse structure. For example, Teufel and Moens (2002) work on summarizing scientific papers assumes that a paper is divided into research goal (aim), outline of the paper (textual), presentation of the papers contribution (methods, results, discussion), and presentation of other work (other). They classify individual sentences for membership in these classes by discourse segmentation 1.1.3.1. This strategy is especially fruitful if the summarization concentrates on specific core parts of a document rather than on the document as a whole.

Summarization can also have other goals like genre-specific goals. For

[3] Common grounding in communication is a concept that has been proposed by Clark & Brennan (1991)[37], which refers to the "mutual knowledge, mutual beliefs, and mutual assumptions" that is essential for communication between two people. The concept is also common in linguistics.

[4] Interlocutor is a person who takes part in a dialogue or conversation - Oxford dictionary

[5] An almost equivalent translation of Doch in English is "yet"

[6] An almost equivalent translation of Wohl in English is "well"

example, one way of summarizing scientific articles would be to highlight the contribution of the article and relating it to previous work (Teufel & Moens, 2002). Also the purpose of indicative summaries is to facilitate the selection of documents that are worth reading (Barzilay & Elhadad, 1997).

In 2010, Louis et al [1] examine the benefits of both the graph structure of text provided by discourse relations and the semantic sense of these relations. They notice that structure information is the most robust indicator of importance, whereas semantic sense only provides constraints on content selection but also is not indicative of important content by itself. However, sense features complement structure information and lead to improved performance. Further, both types of discourse information prove complementary to non-discourse features.

1.1.4.5 Question Answering

In context question answering, each question is situated in a context. In addition to the semantic information carried by important syntactic entities such as noun phrase, verb phrase, preposition phrase, etc), each question also carries distinctive discourse roles with respect to the whole question answering discourse. Specifically, the discourse roles can be categorized based on both the informational and intentional perspectives of discourse (Hobbs, 1985)[8], as well as on the presentation aspect of both questions and answers.

The discourse structure is based on centering and transitions [45, 46]. Transitions from one question to another also determine how context will be used in interpreting questions and retrieving answers.

Discourse transitions also correspond to the intentional, informational, and presentational perspectives of discourse. Intentional transitions are closely related to Grosz and Sidners “dominance” and “satisfaction precedence” relations, which are more relevant to plan-based discourse (Grosz and Sidner, 1986)[3]. Here we focus on informational transitions and presentational transitions that are more relevant to QA systems, since they are targeted for information exchange.

Chai et al (2004) [45] are the first to look upon these informational transitions, which are mainly centered around Topics of questions: how questions are related to each other depends on, how “topics” of those questions evolve. They basically categorize information transitions into three types: Topic Extension, Topic Exploration, and Topic Shift. In the area of interactive question answering as well this kind of discourse structure has been used [47].

Discourse structure can also be used to extract answers for non-factoid questions like ‘why’ or ‘how’[48].

1.1.4.6 Natural Language Generation

Discourse Planning is an important process of imposing ordering and structure over the set of messages to be conveyed in NLG. In the simplest possible terms, this is akin to a story having a beginning, a middle and an end; but most documents have much more discernible structure than this. Good structuring can make a text much easier to read: that this is so can easily be demonstrated by trying to read a version of a newspaper story where sentences and paragraphs have been randomly reordered.

In the work of Reiter & Dale (1997)[49] we find a tree-structured discourse planning. Planning can also be developed through template filling.

In the work of Hermann Hendriks (2002)[50] DRT is used for planning, along with anaphora.

1.1.4.7 Statistical Machine Translation

Contextual information is playing a key role to improve the translation in the area of Statistical Machine Translation (SMT). Therefore discourse information is being included to built a model. There are several ways of inclusion of discourse information into the model: one method to integrate the minimal sets of labels for discourse connectives is to tag their occurrences directly in the phrase table of an already trained statistical MT system; or it can also be done during the training, a phrase table is generated with all phrase pairs found by the word alignment, with their lexical probability and frequency scores. Now we account for some recent works in this area.

Anaphora resolution started to become a topic of interest in the field of Rule-based Machine Translation (RBMT) from the 1990s. This thread of research resulted in the publication of a special issue of the journal Machine Translation on “Anaphora Resolution in Machine Translation and Multilingual NLP” in 1999 [51]. An important empirical study towards achieving discourse structure to improve translation, is by Marcu et al[52]. Another recent work with anaphoric structure is by Hardmeier & Federico (2010)[53], where translation is done from German to English. An important work with pronoun structure was done by Le Nagard and Koehn [54], translating English into French.

In a novel work, Foster et al [55] present an approach to document translation that uses structural features to modify the behavior of a language model, at sentence-level, with the help of linear topic segmentation. Another interesting work is done with patent documents[56]. In this case the system is incrementally trained with "interactive" post-edits and tries to learn from the corrections. All these measures were taken to get the contextual details from a file. The file was also linearly segmented in parts to acquire the conceptual details.

1.1.4.8 Misc.

Another application for research on discourse structure is essay analysis and scoring. The aim of this research area is to improve the quality of essays by providing relevant feedback. This kind of evaluation and feedback is focused on the organizational structure of an essay, which is a crucial feature of quality. To serve this purpose first the specific discourse elements in an essay should be identified. These discourse elements are part of a non-hierarchical genre-specific conventional discourse structure (Burstein, 1998)[57].

We observe that the idea of ordering has a deep impact on discourse structure. We were discussing about organizational ordering of discourse in the previous paragraph. Now we concentrate on the works on temporal ordering for discourse analysis. Ohtsuka & Brewer (1992)[58] investigated the work of organization of discourse to understand the temporal ordering in the narrative texts. They studied discourse structure with the reference of the event structure in narratives. In recent times, Chambers & Jurafsky (2009) described an unsupervised system for narrative schema and their participants. Their model is a two tuple schema with the event(s) and the chains over the event slots. They used FrameNet and PropBank roles as pre-defined classes of events; from this event the system learns with narrative chains by finding salient terms, and also by coreferential chains. Then on the basis of event similarity score the systems learns in an unsupervised manner for unlisted events.

There are new emerging areas as well: we already discussed about patent document analysis in the previous section (Section 1.1.4.7). Another prominent example is Linguistics Forensics or Statement-analysis. This research area involves legal or forensic documents. Discourse analysts are not always allowed to testify but during preparation for a case the semantic and pragmatic evidences are often useful to lawyers. We find only a few publications in this area so far: Bex & Verheij (2011) uses the properties of coherence and also temporal ordering to frame a forensic/legal story [59, 60].

We also discussed several works from the area of information extraction:

a prominent example is TextTiling by Hearst (1997) [17].

1.2 Corpora

Our corpus of interest is Penn Discourse TreeBank (PDTB) 2.0. There are more available discourse corpora, viz. RST, GraphBank (Section 1.1.2.7).

1.2.1 RST TreeBank

In RST Mann and Thompson [14] proposes that coherent text can be represented as a tree formed by the combination of text units through discourse relations. The RST corpus developed by Carlson et al. (2001) [61] that contains discourse tree annotations for 385 WSJ articles from the Penn Treebank corpus. The smallest annotation units in the RST corpus are sub-sentential clauses, which is also called elementary discourse units (EDUs). Adjacent EDUs combine through rhetorical relations into larger spans such as sentences. The larger units recursively participate in relations with others, yielding one hierarchical tree structure covering the entire text.

The discourse units participating in a RST relation are assigned either nucleus or satellite status; a nucleus is considered to be more central, or important, in the text than a satellite. Relations composed of one nucleus and one satellite are called mononuclear relations. In case of multinuclear relations, two or more text units participate, and all are considered equally important. The RST corpus is annotated with 53 mononuclear and 25 multinuclear relations. Relations that convey similar meaning are grouped, resulting in 16 classes of relations altogether: *Cause, Comparison, Condition, Contrast, Attribution, Background, Elaboration, Enablement, Evaluation, Explanation, Joint, Manner-Means, Topic-Comment, Summary, Temporal* and *Topic-Change.*

1.2.2 Penn Discourse TreeBank 2.0

The Penn Discourse Treebank [2] is a resource including one million words from the Wall Street Journal [62], annotated with discourse relations.

Based on the observation that "no discourse connective has yet been identified in any language that has other than two arguments" ([5], p. 15), connectives in the PTDB are treated as discourse predicates taking two text spans as *arguments*, i.e. parts of the text that describe events, propositions, facts, situations. Such two arguments in the PDTB are just called `Arg1` and `Arg2` and are chosen according to syntactic criteria: `Arg2` is the argument syntactically bound to the connective, while `Arg1` is the other one. This means that the numbering of the arguments does not necessarily correspond to their order of appearance in text.

In the PDTB, discourse relations can be overtly expressed either by *explicit* connectives, or by *alternative lexicalizations* (AltLex). The first group of connectives corresponds primarily to a few well-defined syntactic classes, while alternative lexicalizations are generally non-connective phrases used to express discourse relations, such that the insertion of an explicit connective would lead to redundancy. There is also a third type of relations - the *implicit* ones - which can be inferred between adjacent sentences, even if no discourse connective is overtly realized.

Every kind of relation (i.e. explicit, implicit and AltLex) in the PDTB is assigned a sense label based on a three-layered hierarchy: the top-level *classes* are the most generic ones and include Expansion, Contingency, Comparison and Temporal labels (see below resp. examples from *a* to *d*). Then, each class is further specified at *type* and *subtype* level. Since the state of the art in automatic surface-sense classification (at *class* level) has already reached the upper bound of inter-annotator agreement [23], we do not include this task in our pipeline. Instead, we use the *class* label as one of our features, because we can expect to achieve similar performance both with gold standard and with automatically assigned classes.

As for the relations considered, we focus here exclusively on *explicit* connectives and the identification of their arguments, including the exact spans. This kind of classification is very complex, since `Arg1` and `Arg2` can occur in many different configurations. Consider for example the following explicit relations annotated in the PDTB[7]:

(a) *I never gamble too far.* In particular **I quit after one try, whether I win or lose.** [Expansion]

(b) Since **McDonald's menu prices rose this year**, *the actual decline may have been more.* [Contingency]

[7] In all examples from PDTB in this thesis, `Arg1` is reported in italics, `Arg2` appears in bold and discourse connectives are underlined. At the end of the sentence we specify the *class* label

(c) As an indicator of the tight grain supply situation in the U.S., market analysts said **that late Tuesday the Chinese government**, *which often buys U.S. grains in quantity,* **turned** instead **to Britain to buy 500,000 metric tons of wheat.** [COMPARISON]

(d) When **Mr. Green won a $240,000 verdict in a land condemnation case against the State in June 1983**, he says, *Judge O'Kicki unexpectedly awarded him an additional $100,000.* [TEMPORAL]

An explicit connective can occur between two arguments (a) or before them (b). It can also appear inside the argument as shown in (c), where `Arg2` is composed of three discontinuous text spans and `Arg1` is interpolated. Furthermore, `Arg1` and `Arg2` need not to be adjacent, as shown in (d), where "he says" does not belong to any argument span. The latter case is annotated as an *Attribution* in the PDTB, because it ascribes the assertion in text to the agent making it. Attributions occur in 34% of all explicit relations in the PDTB, and represent one of the major challenges in identifying exact argument spans, especially for `Arg2`. However, given the fact that `Arg2` is syntactically bound to the connective, its identification is generally considered an easier task than the detection of `Arg1` [63]. As shown in Table 1, the position of `Arg1` w.r.t. the discourse connective is highly variable and, when it does not occur in the same sentence of the connective, it can be very distant from `Arg2`, even in a preceding paragraph.

Explicit connectives (tokens)	18, 459
Explicit connectives (types)	100
`Arg1` in same sentence as connective	60.9%
`Arg1` in previous, adjacent sentence	30.1%
`Arg1` in previous, non adjacent sentence	9.0%

Table 1: Statistics about PDTB annotation from [2].

Another element increasing the complexity of `Arg1` and `Arg2` identification is the fact that discourse connectives can be expressed by subordinating and coordinating conjunctions as well as by discourse adverbials, and each type is subject to different discourse constraints. Furthermore, argument spans range from clauses, even single verb phrases, to multiple sentences, and they do not necessarily match single constituents in the syntax because they can be discontinuous. For all these reasons, the identification of `Arg1` has been only partially addressed in previous works (see for instance [63].

The PDTB achieved high-valued inter-annotator agreement. Overall agreement for identifying both the arguments (`Arg1` and `Arg2`) of explicit connectives reached 90.2%, with a general tendency of lower scores for `Arg1` and higher scores for `Arg2`. When considering a matching technique that gives credit also to partial overlap, the agreement reaches 94.5% for explicit connectives [2, 64].

1.2.3 Comparison between PDTB & RST

The Penn Discourse TreeBank is not the only effort to annotate discourse structure. Efforts to do so started more than 10 years ago, as a way of providing empirical justification for high-level theories of discourse structure (Grosz & Sidner, 1986; Moser & Moore, 1996). Although much time and energy was devoted to the work (Di Eugenio et al., 1998), the results could not have been widely used in the computational arena mainly due to computation difficulties. The work closest to the Penn Discourse TreeBank is the resource developed by Marcu (1999, 2000) based on Rhetorical Structure Theory (Mann & Thompson, 1988) [14]. RST is a theory of discourse analysis that claims that:

(1) adjacent units of discourse are related by a single rhetorical relation that accounts for the semantic or pragmatic (intentional) sense associated with their adjacency;

(2) units so related form larger units that participate in rhetorical relations with units that they themselves are adjacent to; and

(3) in many, but not all, such juxtapositions, one of the units (the satellite) provides support for the other (the nucleus), which then appears to be the basis for rhetorical relations that the larger unit participates in.

Given these principles, the two main aspects of RST annotation are:

1. demarcation of the elementary discourse units that participate in relations, and

2. labeling of those relations.

The two are not independent. For example, a relation (attribution) postulated between the specification of a speech act (e.g., Riordan said) and its content specified as direct or indirect speech (e.g., We must expand the vision of our party) means that a subject-verb fragment must be marked as an elementary discourse unit if the object of the verb is direct or indirect speech.

Marcus RST-annotated corpus [61] differs from the Penn Discourse TreeBank in three main ways:

First, the discourse relation holding between units has to be inferred, using semantic and pragmatic information, in cases where an overt connective is missing from the discourse. While the RST-annotated corpus records inferred relations, it omits any indication of what was used in inferring them. The PDTB annotation scheme takes two steps towards remedying this omission: the basis of the theory of annotation is DL-TAG parse, although the annotators were never provided the parse tree of the text. [8]

Secondly, RST annotation of elementary discourse units, derived discourse units and rhetorical relations bear the entire burden of supporting language technology algorithms derived from the RST annotated corpus. The PDTB annotation effort is *architecturally* an additional facility with the same text that is already annotated with syntactic structure (Penn TreeBank PTB) and predicate-argument relations (PropBank), although PDTB is completely annotated without providing any parse trees of the text to the annotators. This linkage among the different kinds of annotation surely provide a richer substrate for the development and evaluation of practical algorithms.

[8] Note on DLTAG: DLTAG [Forbes et al 2001] refers to Discourse Lexicalized TAG, whereas TAG stands for Tree Adjoining Grammer. Tree Adjoining Grammar (TAG) is a formalism originally proposed by A. K. Joshi, L. S. Levy, and M. Takahashi in Tree adjunct grammars published by Journal Computer Systems Science, 10(1), 1975. Several variations on that formalism are developed, among which we are interested in lexicalized (LTAG) version. A TAG consists of a number of elementary trees, which can be combined with a substitution, and an adjunction operation. In the "lexicalized grammar" approach (Joshi & Schabes Tree Adjoining Grammars & Lexicalized Grammers Tech Report UPenn 1991), each elementary structure is systematically associated with a lexical item called the anchor. DLTAG provides structural descriptions for the empty connectives. The DLTAG parse links up to sentence-level syntactic and semantic annotation for each sentence. Identifying the empty connectives and accessing sentence-level syntactic and semantic information are crucial steps towards an automated inference of discourse relations in the absence of lexically realized connectives.

Finally, the number of documents those are annotated with RST is too small to built any supervised system, on the other hand the whole WSJ corpus is annotated with PDTB.

1.2.4 BioDRB

Text annotation is a process that facilitates the use of corpora knowledge various ways, the most common case of usage is development of supervised classification systems. News corpora are the oldest text resources, which were the subject of annotation for decades. There is an increasing interest to annotate biomedical corpora to make a good use of the bio-knowledge-intensive texts. Genia corpus [65] is a biomedical text collection where annotation of named entities, events, (parse) treebank are available beside raw texts. This specific corpus has been recently extended to include discourse relations with PDTB annotation style, primarily for twenty-four raw text files. This collection of annotation is called as Bio Discourse Relation Bank (BioDRB)[66] [9]. This annotation is done with some changes as the domain changed from news to bio-medical texts:

1. this resource has a two-tiered sense hierarchy against the three tiered one in PDTB. The overall structure is more flat, if compared to PDTB 2.0.

2. the number of the surface senses is now sixteen compared to four in PDTB, where some of second level senses of PDTB have been introduced as surface sense, there are newly introduced senses as well.

1.2.5 Some Discourse-Related Resources in Other Languages

We discuss now about the published resources in languages other than English.

In Czech, the next version of the Prague Dependency TreeBank, PDT 2.0 is annotated only with the intra-sentential annotation related to discourse structure. The new version PDT 3.0 is not released yet. This new layer of annotation is being created (Mladov et al., 2008) that will capture

[9] The problem of working with BioDRB at this moment is the insufficient number of annotated documents to train a system; also the annotated documents does not overlap with Genia treebank documents, so we are unable to view the performance of gold-standard data with our system.

the connective a relations in discourse, both within a sentence (through coordinating and subordinating relations) and across the sentence boundary, inspired by PDTB 2.0 annotation [5].

In Danish, the Copenhagen Dependency TreeBank (CDT) (Buch-Kromann et al., 2009; Buch- Kromann & Korzen, 2010) also comprises four other parallel treebanks (English, German, Italian, and Spanish). CDT have been annotated with morphology, syntax, discourse structure and coreference. Word aligned with the Danish source text for translational equivalence. The CDT resembles the RST Discourse TreeBank (Section 1.2.1) both in assuming a nucleus-satellite distinction on all discourse relations and in taking a tree-structured analysis (a form of dependency structure) to fully cover the text.

In Dutch there is a corpus of 80 Dutch texts being annotated for discourse structure and for relational and lexical cohesion (van der Vliet et al., 2011). They made a distinction of discourse structure across genres. The corpus includes 40 expository texts (20 articles on astronomy from an on-line encyclopedia and 20 from a popular science website) and 40 persuasive texts (20 fund-raising letters and 20 commercial advertisements). Discourse structure is annotated in the style of the RST corpus (Section 1.2.1). The annotation of relational cohesion involves all lexical and phrasal elements that signal coherence relations at either locally or globally, while the annotation of lexical cohesion involves both repetition (full or partial) and standard semantic relations between nouns, verbs, adjectives and adverbs.

In German, the Potsdam Commentary Corpus (Stede, 2004) consists of 170 commentaries from the German regional daily newspaper *Märkische Allgemeine Zeitung*, which have been annotated with part-of-speech tags, syntactic analyses, rhetorical structure in the style of Rhetorical Structure Theory (Mann & Thompson, 1988), discourse relations associated with explicit discourse connectives, links from anaphoric and bridging expressions to the antecedents that license them, and information structure.

In Hindi, there is Hindi Discourse Relation Bank (Oza et al., 2009), a 200K-word corpus, drawn from a 400K-word corpus of news articles from the Hindi newspaper *Amar Ujala* whose sentences have been annotated with syntactic dependencies. This is annotated in PDTB-style of annotation, with some differences (for eg. they annotated pragmatic relations) depending on the language.

In Turkish, the METU Turkish Discourse Bank (Zeyrek & Webber, 2008; Zeyrek et al., 2009; Zeyrek et al., 2010) aims to annotate a 500K-word subcorpus of the 2-million word METU Turkish Corpus (Say et al., 2004). The sub-corpus contains a wide range of texts novels, newspaper columns, memoirs, etc. . The initial annotation has focused on discourse relations that are signaled by explicit discourse connectives, realized either as words or affixes.

1.3 Outline of Thesis

The structure of this thesis work is organized as follows:

Chapter 1 In this chapter we introduce the subject of the thesis, then we proceed with the discussion of basic backgrounds: the state-of-the-art over the time-periods, the conceptual changes about discourse structure and the terminologies. We also discuss some areas of application of discourse structure. Next to this, we proceed further to the corpora description and also include a comparison between PDTB, our corpora of interest and the RST tree bank, a popular corpora.

Chapter 2 In this chapter we describe the core pipeline structure of our discourse parser, before this discussion, we derive the problem statements. Then we enrich this section with our motivation to work in this area; we brief about our contribution to this problem and also our approaches to tackle the problem. We also detail on our evaluation strategy and metrics. We establish a baseline to evaluate our system. We used conditional random field as the classifier of the system, therefore we briefly depict the details of this classifier, emphasizing our cascaded prediction method. We also include experiment details, results on gold-labeled annotation data, and post-analysis of results.

Chapter 3 In this chapter we discuss the parser evaluation using automatically generated inputs. We also observe the impacts using "semi-automatic" datasets through the parser pipeline.

Chapter 4 In this chapter we attempt to increase the performance of parser using global features with the help of re-ranking strategies with our best performing model. The single best model performance was chosen as the baseline for this re-ranker. We present the performance difference of re-ranking strategies across three popular discriminative algorithms: linear best vs. rest support vector machine, voted perceptron and Online Passive-Aggressive perceptron.

Chapter 5 In this chapter we summarize the whole research activities. We conclude with our result discussion. We also find the possible application area for our method and system, highlighting the main challenges to this system.

Chapter 6 This is an appendix chapter. This depicts another semantic structure, predicate-argument-structure, other than the discourse structure, that considers FrameNet hierarchy to achieve a semantic structure in human-human conversations. This work is done using Italian human-human conversation data collected under EU LUNA project.

2 Parsing Discourse: An Overview

2.1 Introduction

In the first chapter we introduced the topic of this thesis; we also discussed about the basic background and the terminology, the approaches and the application areas, and also about the available corpora related to the current topic.

In this chapter we will summarize only that part of the related works, which is closely related to our research area. Then we will go through the problem statements and motivation for this research work. We will also describe our approach (Section 2.3.2.1) to the problems with the proposed solutions; this discussion will also include the main contribution to this thesis-work. Following all these discussions, we will illustrate the first piece of work: the discourse pipeline and the related experimentation, results and analyses.

2.2 Related Backgrounds

One of the specific tasks that we address in this thesis – automatic extraction of discourse arguments for given explicit discourse connectives – has been attempted a number of times. Soon after the initial release of the PDTB, it was realized that sentence-internal arguments may be located and classified using techniques similar to semantic role detection and classification methods. (Wellner et al 2007, Punyakanok et al 2008) [67, 68] were the first to carry out such an experiment on the PDTB, and Elwell et al (2008)[69] later improved over their results. However, their task was limited to retrieving the argument *heads*. In contrast, we integrate discourse segmentation in the parsing pipeline because we believe that spans are necessary when using the discourse arguments as input to applications such as opinion mining, where attributions need to be explicitly marked. Besides, no gold data are available for head-based discourse parsing evaluation and they have to be automatically derived from parse trees with a further processing step. With our approach, instead, we can directly use PDTB argument spans both for training and for testing.

Dinesh et al (2005) [70] extracted complete arguments with boundaries, but only for a restricted class of connectives. The recent work by Prasad et al (2010)[63] is also limited, since their system only extracts the *sentences* containing the arguments.

In our work, we assume that explicit discourse connectives are given beforehand, either taken directly from a gold standard or automatically identified. The second task based on PDTB was tackled among others by Pitler et al (2008) [71] and Pitler & Nenkova (2009) [23].

In addition to the work on finding explicit connectives and their arguments, there has been recent work on classification of *implicit* discourse relations, see for instance Lin et al (2009)[72]. In a similar classification experiment, Pitler et al (2009) [73] investigated features ranging from low-level word pairs to high-level linguistic cues, and demonstrated that it is useful to model the sequence of discourse relations using a sequence labeler. Although they both outperformed their respective baselines, this task is very difficult and performances are still very low.

In continuation to this discussion we also point out some assumptions drawn from the discussions in the first chapter. These assumptions are the essentials for the rest of the thesis.

1. We assume that one discourse connective has two and only two arguments (“no discourse connective has yet been identified in any language that has other than two arguments” (Webber et al 2011, [5])

2. We hypothesize that in this work the discourse structure is defined as the non-overlapping discourse arguments anchored with the discourse marker i.e. the connective.

3. We assume that a discourse argument is essentially a clause-like structure

4. We consider only the explicit connective as the anchor of two arguments.

5. The Coverage relates to how much of a discourse belongs to the structural analysis (Webber 2011) [5]. We cover only a part of the full text body: we do not use all the parts of a text input in the course of experimentation with discourse structures, instead we use a 5-sentence discourse window keeping the sentence bearing the connective in the middle. Thus our parser provides a partial coverage to the discourse.

6. We do not use the attribution annotations of PDTB.

7. We cover only the surface-level sense hierarchy of connectives of the three-layered sense hierarchy of PDTB. The surface-senses of a connective refer to the four top-level classes in PDTB sense hierarchy, viz. TEMPORAL, COMPARISON, CONTINGENCY and EXPANSION.

8. Symmetry has to do with the importance of different parts of a discourse structure – whether all parts have equal weight (Webber 2011) [5]. We give the same importance to the two arguments of a connective, therefore the relation is symmetric. Thus classification performance of the both arguments are considered equally essential for the discourse structure.

2.3 Problem Statement

Automatic discourse processing is considered one of the most challenging NLP tasks due to its dependency on lexical and syntactic features and on inter-sentential relations. While automatic discourse processing of structured documents or free text is still in its infancy, a number of applications of this technology in practical NLP systems has been proposed. For instance, Somasundaran et al (2009) [36] describe the use of discourse structure for opinion analysis. Other applications include conversational analysis and dialog systems (Tonelli et al, 2010) [74].

The corpora developed for discourse analysis may differ in many ways, but the basic steps of the annotation is more or less comprised of as follows:

a. segmenting basic units of discourse

b. identifying the arguments of discourse

c. establishing a type of relation between them.

We currently focus on the first two steps, as we decide to use a third-party tool (Pitler & Nenkova 2009) [23] to establish the type of relation between the arguments for our automated system (see Chapter 3).

We identified some basic problems in this area:

1. It is hard to locate the argument position, as we observe from the previous works with argument heads (Wellner et al 2007, Elwell et al 2008) [67, 69]; moreover the metrics to evaluate the correct head is also very loosely grounded as there is no as such gold-labeled heads; also an

incorrect parse tree of a sentence surely results in a wrong detection of the head(s).

2. In previous works it is admitted that determining the arguments with boundaries is a non-trivial task. There was no such attempt to detect arguments with boundaries.

3. Across the literature it is observed that the parse trees play an important role in discourse analysis; there was no such initiative to consider the complex cases from the real-life scenarios where can apply effective features, extracted with and without parsing.

2.3.1 Motivation

The diversified application areas for discourse parsing, and the opportunity to contribute to the semantic and pragmatic analysis of discourse primarily motivated this work with discourse. We already discussed about the application variability of discourse structures under the section "Application" (Section 1.1.4). Some examples from the corpus or the resources of other application areas that use discourse structure, primarily motivated us for this thesis-work. We present some examples that express two opposite pictures with discourse structures: one is implication and the other is generation.

The first example is taken from the movie review database presented in "Opinion Mining and Sentiment Analysis" by (Pang & Lee 2008)

Example of Implication 1. This film should be brilliant. It sounds like a great plot, the actors are first grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance. *However*, it can't hold up.

The meaning of the whole chunk of the text is completely depending on the connective "However", which gives the contrastive touch to the accumulated positive sentiment towards the movie, whereas the former part of the review is full of positive comments about the movie.

Another example of implication with a hierarchical meaning structure (this popular example is taken from the book: Speech & Language Processing by Jurafsky & Martin 2000)

Example of Implication 2. I love to collect classic automobile. My favorite car is 1899 Duryea. *However*, I prefer to drive my 1999 Toyota.

In this case we note that the first sentence is a nucleus or the bottom-line, and the next sentences are satellites. Additionally, we observe the contrastive touch with the usage of “however”: it is very evident that the sentence “I prefer to drive...” is contrast to the other part of example; we also observe that there is a covert elaborative relation between the first two sentence (or arguments).

Now we consider another popular example from natural language generation (cf. Speech & Language Processing by Jurafsky & Martin 2000)

Example of Generation. Save the document: *First*, choose the save option from the file menu. This causes the system to display the Save-As dialog box. *Next* choose the destination folder *and* type the filename. *Finally*, press the save button. This causes the system to save the document.

This is a machine generated text whose meaning depends on the connectives of the sentences. This chunk of text resulted as an output of a system that runs a discourse planning module. The module uses several procedures like “select folder”, “type filename” etc. These small procedures are bound together with discourse anchors like“first”, “next”, “finally”. These generates a sequentially-ordered procedural hierarchy that causes in saving a document mechanically.

Thus, a discourse parser that parses the anchors with their arguments from texts can be used to solve many problems in diversified areas of research.

2.3.2 Contribution

We are interested in machine understanding of semantic and pragmatic knowledge. The use of discourse structure in the area of natural language processing is also a motivation for this work.

The main contribution of the thesis is the end-to-end discourse parser that takes raw text as input and outputs discourse structures (i.e. explicit discourse connectives and the two arguments with the boundaries for each connective).

The design of cascaded structured prediction pipeline to generate discourse structure is also a novel implementation to solve this kind of problem with good accuracy.

The use of shallow syntactic features and also a set of data-driven, nongrammatical structural (global) features for discourse structure analysis are also important contributions of this thesis-work. This method may be implemented for languages and domains for which full sentence parsing is not possible.

2.3.2.1 Our Approach

In this work we divide the whole task of discourse parsing into two sub-tasks: connective classification and argument segmentation and classification. Several successful attempts have already been made in the direction of automatic classification of connectives, while token-level argument segmentation has not been explored. Therefore in this work we will focus on the segmentation and labeling of discourse arguments (`Arg1` and `Arg2`) with full spans, as defined in the annotation protocol of the Penn Discourse Treebank (PDTB) [2].

We present a methodology that, given explicit discourse connectives, automatically extracts discourse arguments by identifying `Arg1` and `Arg2` including the corresponding text spans. We call this approach *shallow* following Prasad et al (2010) [63] as opposed to tree-like representations of discourse, as in Rhetorical Structure Theory [14]. Indeed, we provide a flat chunk classification of discourse relations, building a non-hierarchical representation of the relations in a text.

The discourse parser is designed as a cascade of argument-specific CRFs trained on different sets of lexical, syntactic and semantic features. The evaluation is made in terms of exact and partial match of arguments. The partial match condition may be useful in the case of noisy input or for applications that do not require exact alignment.

2.4 Overview of Processing Pipeline

We show that discourse annotation can be performed *in a cascaded pipeline* handling all types of explicit connectives and argument positions. The fundamental idea is to divide the whole complex task into several small and simpler independent subtasks, in order to feed the output of each step into the following one. Later in this chapter, we also show that we benefit from this cascaded architecture.

Starting with the implementation with gold-labeled data we gradually attempt to build an end-to-end parser for discourse analysis. The pipeline implementation with entirely gold-standard data is described below. In the course of making an end-to-end parser we also implemented and evaluated different version of the system with the semi-automatic data, in order to understand the drop in performance from the gold-labeled discourse parser to the end-to-end automatic discourse parser. There are actually four pipelines: gold-standard, full automatic setting, pipeline using gold-standard syntactic parse tree and automatically generated connectives with AddDiscourse [23] tool and also the pipeline using automatic parse trees and gold-labeled connectives. The goal is to establish successful method with gold-labeled pipeline, then to implement that method to the full automatic pipeline. We use the remaining pipelines to understand the contribution of errors from the automatic parse trees and automatically generated connectives.

In this chapter we concentrate only on the gold labeled part of the implementation. The rest of the pipelines will be described and evaluated in the Chapter 3; in the next chapters we will concentrate only with gold-labeled and full automatic pipelines.

2.4.1 Overview of Pipeline-I: Full Gold-Standard

An overview of the pipeline is given in Fig. 2. Note that, this representation includes data pre-processing, training and testing only with gold-labeled data. In this pipeline we used Penn TreeBank (PTB) and PDTB both for training and testing the discourse parser.

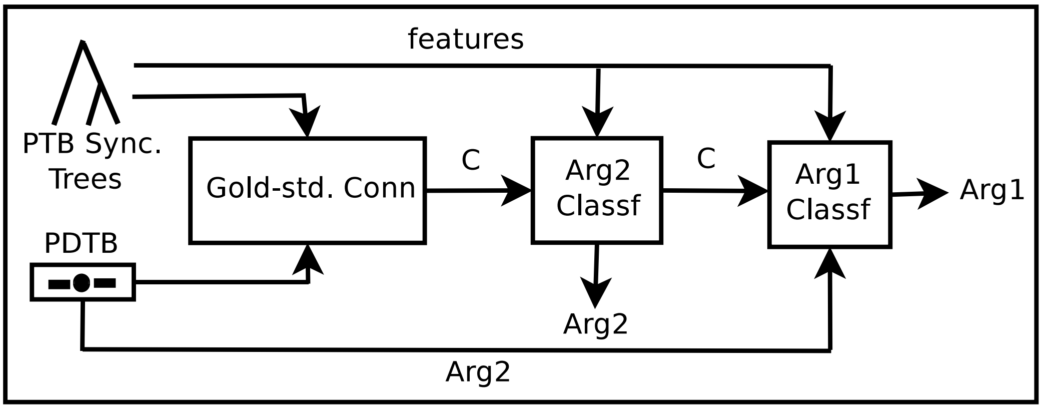

Figure 2: Argument parsing pipeline given Gold-Std Connective(C)

In contrast to previous works, our shallow parsing strategy combines the identification of non-overlapping sequences as connective arguments and the tagging of such text chunks with `Arg1` and `Arg2` labels.

Since our experiments are based on gold-standard parse trees, we take advantage of the overlap between the PDTB and the Penn Treebank documents [62] in order to map PDTB discourse annotation onto PTB parse trees. We extract the gold-standard connectives with the corresponding top-level sense label from PDTB relations, since this sense label is also one of the features used by our system. This feature is denoted as C in Fig. 2. Besides, we also extract from the PTB trees all syntactic features needed by the system for the first parsing subtask, which is the identification of `Arg2`.

After the identification of `Arg2` given the connective sense label and feature(s) from the gold parse trees, we proceed with the classification of `Arg1`. This step-by-step methodology is different from previous approaches like the one by Wellner et al (2007) [67], where the authors select *pairwise* the best heads of `Arg1` and `Arg2` in order to capture their dependencies, and also by Elwell et al (2008) [69], who additionally develop *connective-specific* models. Our approach is motivated by two intuitions: first, the identification of `Arg2` and `Arg1` may require different features, since the two arguments have different syntactic and discourse properties, as discussed in Section 1.2.2. Second, the identification of `Arg2` is much easier than the identification of `Arg1`, because the former is syntactically bound to the connective. For this reason, a two-step decision architecture seems more appropriate, because we can start with the easier classification task and then exploit additional output information to tackle the second task.

2.5 Feature description

We report in Table 2 the list of all features considered in the argument labeling task and we explain them in the light of the example in Fig. 3.

Despite the complex task, the feature set is quite small for both arguments. For the identification of `Arg1`, we include one additional features which corresponds to `Arg2` gold standard labels. Note that the best performing set of features does not include all those listed in the table (see feature analysis in Tables 5 and 6).

The sense of the connective (F2) refers to one of the four top-level classes in PDTB sense hierarchy, namely Temporal, Comparison, Contingency and Expansion. In the sentence reported in Fig. 3, for example, only "when" bears the *temporal* label, while all other tokens are assigned as a "null".

Features used for **`Arg1`** and **`Arg2`** segmentation and labeling.	
F1.	Token (T)
F2.	Sense of Connective (CONN)
F3.	IOB chain (IOB)
F4.	PoS tag
F5.	Lemma (L)
F6.	Inflection (INFL)
F7.	Main verb of main clause (MV)
F8.	Boolean feature for MV (BMV)
F9.	Previous sentence feature (PREV)
Additional feature used only for **`Arg1`**	
F10.	**`Arg2`** Labels

Table 2: Feature sets for `Arg1` and `Arg2` segmentation and labeling.

The IOB (Inside-Outside-Begin) chain[10] (F3) is extracted from a full parse tree and corresponds to the syntactic categories of all the constituents on the path between the root note and the current leaf node of the tree. Experiments with other syntactic features proved that IOB chain conveys all deep syntactic information needed in the task, and makes all other syntactic information redundant, for example clause boundaries, token distance from the connective, constituent label, etc. In Fig. 3 the path between "flashed" and the root node is highlighted. The corresponding feature would be *I-S/E-VP/E-SBAR/E-S/C-VP*, where B-, I-, E- and C- indicate whether the given token is respectively at the beginning, inside, at the end of the constituent, or a single token chunk. In this case, "flashed" is at the end of every constituent in the chain, except for the last VP, which dominates one single leaf.

In order to extract the morphological features needed, we use the *morpha* tool [75], which outputs lemma (F5) and inflection information (F6) of the candidate token. The latter is the ending usually added to the word root to convey inflectional information. It includes for example the *-ing* and *-ed* suffixes in verb endings as well as the *-s* to form the plural of nouns. In our example sentence, this feature would be for example *s* for "traders" and "heads", etc.

As for features (F7) and (F8), they rely on information about the main

[10] We extracted this feature using the Chunklink.pl script made available by Sabine Buchholz at `http://ilk.uvt.nl/team/sabine/chunklink/README.html`

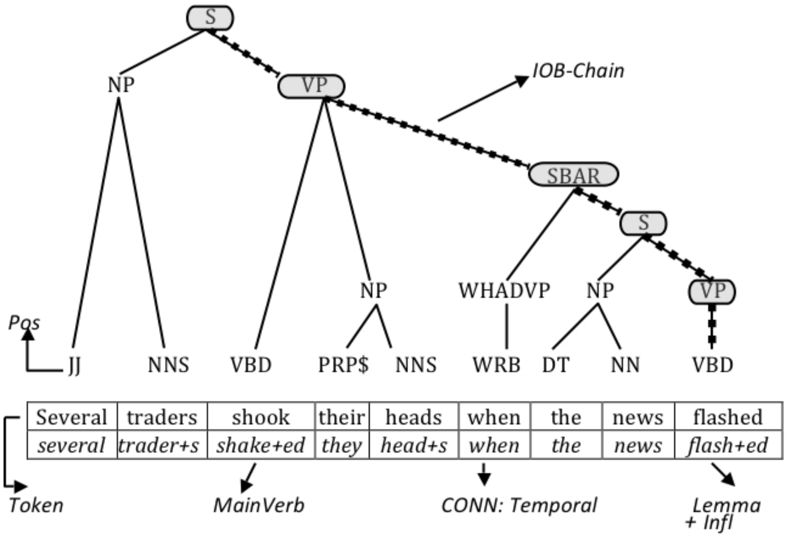

Figure 3: Example sentence with system features

verb of the current sentence. More specifically, feature (F7) is the main verb token (i.e. *shook* in our example), extracted following the head-finding strategy by Yamada (2003)[76], while feature (F8) is a boolean feature that indicates for each token if it is the main verb in the sentence or not.[11]

The previous sentence feature "Prev" (F9) is a connective-surface feature and is used to capture if the following sentence begins with a connective. Our intuition is that it may be relevant to detect `Arg1` boundaries in inter-sentential relations. The feature value for each candidate token of a sentence corresponds to the connective token that appears at the beginning of the following sentence, if any. Otherwise, it is equal to 0.

We also add gold-standard `Arg2` labels (F10) as an extra information for `Arg1` identification.

2.6 Evaluation

2.6.1 Evaluation Scheme

We present our results using precision, recall and F1 measures. To compute precision and recall, we use three scoring schemes: *exact*, *intersection* or

[11] We used the head rules by Yamada & Matsumoto (`http://www.jaist.ac.jp/~h-yamada/`)

partial, and *overlap* scoring. In the *exact* scoring scheme, a span extracted by the system is counted as correct if its extent exactly coincides with one in the gold standard. However, we also use the other two scoring schemes since *exact* scoring may be uninformative in some situations where it is enough to have a rough approximation of the argument spans. In the *overlap* scheme, an expression is counted as correctly detected if it overlaps with an expression in the gold standard, i.e. if their intersection is nonempty. Finally, the *intersection* scheme was used since the overlap scheme suffers from a number of problems: *i)* it is possible to "fool" the metric by creating a span covering the whole sentence; *ii)* it does not give higher credit to output that is "almost perfect" rather than "almost incorrect."

To explain *intersection* scoring, we first define the *span coverage* c of a span s with respect to another span s', which measures how well s' is covered by s:

$$c(s, s') = \frac{|s \cap s'|}{|s'|}$$

In this formula, $|x|$ means the number of tokens in a span x, and the intersection operator $\cap$ gives the set of tokens that two spans have in common.

Using the span coverage, we define the *span set coverage* C of a set of spans $\boldsymbol{S}$ with respect to a set $\boldsymbol{S'}$:

$$C(\boldsymbol{S}, \boldsymbol{S'}) = \sum_{s_j \in \boldsymbol{S}} \sum_{s'_k \in \boldsymbol{S'}} c(s_j, s'_k)$$

We now define the *intersection*-based precision P and recall R of a proposed set of spans $\hat{\boldsymbol{S}}$ with respect to a gold standard set $\boldsymbol{S}$ as follows:

$$P(\boldsymbol{S}, \hat{\boldsymbol{S}}) = \tfrac{C(\boldsymbol{S}, \hat{\boldsymbol{S}})}{|\hat{\boldsymbol{S}}|} \quad R(\boldsymbol{S}, \hat{\boldsymbol{S}}) = \tfrac{C(\hat{\boldsymbol{S}}, \boldsymbol{S})}{|\boldsymbol{S}|}$$

where $|X|$ is the number of spans in a span set X.

This scheme corrects the problem of *overlap* scoring: If the system proposes a span covering the whole sentence, the span coverage will be low and result in a low soft precision. Conversely, a low soft recall will be assigned if only a small part of a gold-standard span is covered. Note that our measures are bounded below by the *exact* measures and above by the *overlap*-based measures.

In brief, some important facts about the metrics and usage through the thesis:

In case of *overlap* measure, a segment proposed by the system is counted as correct if it overlaps (i.e. shares at least one token) with a segment in the gold standard.

In case of *partial/intersection* measure, a segment proposed by the system receives a score between 0 and 1 depending on how many tokens it shares with a segment in the gold standard.

The three evaluation measures can also be formalized as follows

- *Exact* Evaluation:

$$Precision = \frac{\#correct}{\#guessed} \, ; Recall = \frac{\#correct}{\#in\_gold} \tag{1}$$

[12]

- *Partial/Intersection* Evaluation:

$$Precision = \frac{\#proposed\_segment\_found}{\#guessed} \, ; Recall = \frac{\#proposed\_segment\_found}{\#in\_gold} \tag{2}$$

- *Overlap* Evaluation:

$$Precision = \frac{\#overlaps}{\#guessed} \, ; Recall = \frac{\#overlaps}{\#in\_gold} \tag{3}$$

The exact match evaluation performs same measurements as the evaluation script for chunking evaluation at the CoNLL shared task does (cf. URL: `http://www.cnts.ua.ac.be/conll2000/chunking/conlleval.txt`).

We used the evaluation module as a black-box through out the whole thesis work, only except for oracle accuracy calculation (Chapter 4).

2.7 Experiment

All data used in our experiments are taken from PTB and PDTB. In particular, folders $02 - 22$ are used to train the model, while folders $00 - 01$ belong to the development set, and folders 23 and 24 are meant for testing. Our goal is to classify discourse arguments given the connectives by focusing

[12] Hash sign means ‘the count of’.

on one relation at time. Since this results in a large search space for the classifier, we prune the search space trying to preserve the relevant contextual information related to the arguments. For this reason, the data given as input to the classifier include a window of two sentences before and after the given connective. This allows us to reduce the search space by more than 90%. In Table 3 we give the statistics of the explicit relation instances for the whole PDTB corpus and span limit sets. Most of the explicit relations (95%) occur within the five sentence window (two preceding and two following the sentence including the connective token).

Number of all explicit relations in PDTB	18459
Number of explicit relations with `Arg1` entirely *inside* the window	94%
Number of explicit relations with `Arg1` entirely *inside or overlapping* the window	95%

Table 3: Statistics about explicit relations and `Arg1` extension.

2.7.1 Baseline

We compute a baseline (Table 4) for each parsing subtask, i.e. `Arg1` and `Arg2` identification with the test dataset. To obtain this baseline, we take into account that *i*) `Arg2` is, by definition, the argument syntactically bound to the connective and *ii*) 90% of the relations in PDTB are either intra-sentential or involve two contiguous sentences. Thus, `Arg2` baseline is computed by labeling as `Arg2` the text span between the connective and the beginning of the next sentence. The other baseline, on the other hand, is computed by labeling as `Arg1` all tokens in the text span from the end of the previous sentence to the connective position. In case the connective occurs at the beginning of a sentence, then the baseline classifier tags the previous sentence as `Arg1`.

2.7.2 Structured Prediction by Conditional Random Field

We use the CRF++ tool (`http://crfpp.sourceforge.net/`) for sequence labeling classification (Lafferty et al 2001) [77], with second-order Markov dependency between tags. Beside the individual specification of a feature in the feature description template, the features in various combinations are also

		P	R	F1
Arg2	**Exact**	**0.53**	**0.46**	**0.49**
	Partial	0.80	0.85	0.82
	Overlap	0.98	0.85	0.91
Arg1	**Exact**	**0.19**	**0.19**	**0.19**
	Partial	0.50	0.68	0.58
	Overlap	0.70	0.68	0.69

Table 4: Baseline Results of `Arg1` and `Arg2` with test dataset.

represented. We used this tool because the output of CRF++ is compatible to CoNLL 2000 chunking shared task, and we view our task as a discourse chunking task with sequential labeling. On the other hand, linear-chain CRFs for sequence labeling offer advantages over both generative models like HMMs and classifiers applied at each sequence position. Also Sha and Pereira (2003) [78] claim that, as a single model, CRFs outperform other models for shallow parsing.

Structured prediction is a very debated concept. In fact, of all the primary prior works that propose solutions to the structured prediction problems, none explicitly defines the problem (McCallum, Freitag, and Pereira, 2000 [79]; Lafferty, McCallum, and Pereira, 2001 [77]; Punyakanok and Roth, 2001 [68]; Collins, 2002 [80]; Taskar, Guestrin, and Koller, 2003 [81]; McAllester, Collins, and Pereira, 2004 [82]; Tsochantaridis et al., 2005 [83]). In all cases, the problem is explained and motivated purely by means of examples.

One of these examples is sequential labeling: given an input sequence, produce a label sequence of equal length. Each label is drawn from a small finite set. This problem is typified in NLP by part-of-speech tagging.

We use the conditions stated in the thesis by Daume III [84]. Generalizing over many examples by the community, Daume III leads us to a partial definition of structured prediction:

Condition 1 In a structured prediction problem, output elements $y \in \mathcal{Y}$ decompose into variable length vectors over a finite set. That is, there is a finite $M \in \mathbb{N}$ such that each $y \in \mathcal{Y}$ can be identified with at least one vector $v_y \in M^{T_y}$, where T_y is the length of the vector.

Condition 2 In a structured prediction problem, the loss function does not decompose over the vectors v_y for $y \in \mathcal{Y}$. In particular, $l(x, y, \hat{y})$ is not invariant under identical permutations of y and $\hat{y}$. Formally, we must make this stronger: there is no vector mapping $y \mapsto v_y$ such that the loss function decomposes, for which $|v_y|$ is polynomial in $|y|$.

The condition 1 is easily satisfied in our scenario, whereas the condition 2 satisfies well for binary classification [84]. Now if we consider the cascaded pipeline architecture of our parser then we find that both the conditions hold true for our parser.

2.7.3 Cascaded Prediction

In structured prediction tasks, such as part-of-speech tagging, machine translation and gene prediction, the models with increasing complexity of inference are a considerable problem. For example, a first order conditional random field (CRF) (Lafferty et al., 2001) [77] is fast to evaluate but may not be an accurate model for phoneme recognition, while a fifth order model is more accurate, but the model complexity may lead to overfitting problems due the sparseness of the training data. Therefore, it is a better choice to reduce the dimensions of classification label-set, i.e. if we have a ternary set of multilabel (i.e. **Arg2**, **Arg1** and Other) we attempt to reduce it to a binary problem (i.e. {Arg2, Other} and {Arg1, Other}), to attack one after the other, using the results from one classification to another in pipeline. There is the possibility of error propagation through the pipeline, but we may measure or compare this error as well.

Cascaded prediction is kind of *ensemble learning* in broader terms (Alpaydin, 2004) [85]. We implement the cascaded prediction technique in the area of discourse parsing first time. The cascaded prediction is not a novel technique in the area of natural language processing: starting from Goodman's [86] multiple pass parsing we find many successful works (X. Carreras, M. Collins, & T. Koo, 2008 [87]; E Charniak & M Johnson, 2005 [88]; S. Petrov, 2009 [89]), those use cascaded prediction in some ways. The key insights of using cascaded classifiers are:

1. to use smaller, faster and simpler classifiers first to reduce the search space gradually; to use the more complex model later stages (Viola & Jones, 2001) [90].

2. to establish a trade-off between minimizing the number of errors incurred by each level and maximizing the number of filtered assignments at each level (Weiss & Tasker, 2010) [91].

In the case of our cascaded pipeline, we decided to identify the `Arg2` first because it is syntactically bounded with the connective so is more easy to identify, then from this stage we carry forward the results of `Arg2` classification as a feature for `Arg1` classification. We also investigate the impact of error propagation through this pipeline.

In the subsequent subsections under this section, first we perform the feature analysis in cascaded scenario (i.e. using `Arg2` labels as features for the `Arg1` classification and not the other way around). Then we proceed to the final results section.

2.7.3.1 Feature Selection

Our feature set includes a small set of lexical, syntactic and semantic features, which convey the essential information needed to represent the arguments' position and the clausal boundaries, as well as the internal clause structure. We first take into account the features commonly used in similar works, for example by Wellner et al (2007) [67] and Elwell et al (2008) [69], and then carry out a selection step in order to identify only the feature combination that performs best in our parsing task. Note that both Wellner et al (2007) [67] and Elwell et al (2008) [69] limit their classification to argument heads, thus they may employ features that are not very relevant to our approach.

We follow the hill-climbing (greedy) feature selection technique proposed by Caruana & Freitag [92]. In this optimization scheme, the best-performing set of features is selected on the basis of the best F1 "exact" scores. Therefore, we increase the number of features at each step, and report the corresponding performance. In order to understand better the contribution of each feature and also to avoid sub-optimal solutions, we also run an ablation test by leaving out one feature in turn from the best-performing set. We use the development split to generate results for the feature analysis to find the best performing feature set, whereas the train split is used to built model. Final results are generated using only the test split.

The results of our feature analysis are reported in Table 5 for `Arg2` and Table 6 for `Arg1`. We do not report the scores having zero as F1-measure.

Both the feature-in-isolation procedure and the ablation test show that the connective sense feature is the most relevant feature for `Arg1` and `Arg2`, whereas the analysis results for `Arg1` show that the "Prev" feature is also important.

We observe that the performance using the lemma increases if integrated with the inflection feature, while inflection in isolation scores a null Precision,

Features	P	R	F1
Features in Isolation			
Token (T)	0.25	0.08	0.13
Connective (CONN)	0.58	0.50	0.54
IOB_Chain (IOB)	0.22	0.06	0.10
PoS	0.26	0.03	0.05
Lemma (L)	0.26	0.09	0.13
Morph(L+INFL)	0.27	0.05	0.09
Hill-Climbing Feature Analysis			
T+CONN	0.80	0.73	0.76
T+CONN+IOB	0.83	0.75	0.79
T+CONN+IOB+Morph	**0.84**	**0.76**	**0.80**
T+CONN+IOB+Morph+Prev	0.83	0.75	0.79
T+CONN+IOB+Morph+Prev+PoS	0.85	0.75	0.79
Token+CONN+IOB+PoS +Morph+BMV+Prev	0.84	0.74	0.78
Token+CONN+IOB+PoS +Morph+MV+BMV+Prev	0.82	0.72	0.77
Feature Ablation			
T+CONN+IOB	0.83	0.75	0.79
T+CONN+Morph	0.80	0.69	0.74
IOB+CONN+Morph	0.84	0.72	0.77
T+IOB+Morph	0.29	0.16	0.20

Table 5: Results with Single and Combined Features for `Arg2`

Recall and F1. Therefore, we consider lemma and inflection together as a single feature, which we call *Morph*.

We show that the best performing set for **`Arg1`** includes eight features, whereas the best feature combination for **`Arg2`** classification is achieved using only four features, namely token, IOB chain, connective sense and *Morph*.

The best combination for **`Arg1`** classification includes all features from our initial set described in Table 31, except MV and PoS. This is probably due to the fact that PoS information becomes redundant for the classifier and BMV and MV convey the same kind of information.

Features	P	R	F1
Features in Isolation			
Token (T)	0.29	0.03	0.05
Connective (CONN)	0.40	0.08	0.14
IOB_Chain (IOB)	0.18	0.04	0.06
PoS	0.14	0.00	0.01
Lemma (L)	0.26	0.03	0.05
Morph(L+INFL)	0.27	0.02	0.03
Prev_feat(PREV)	0.57	0.09	0.16
Hill-Climbing Feature Analysis			
T+CONN	0.62	0.30	0.40
T+CONN+IOB	0.65	0.32	0.44
T+CONN+IOB+Prev	0.69	0.45	0.55
T+CONN+IOB+Arg2+Prev	0.69	0.50	0.58
T+CONN+IOB+BMV+Arg2+Prev	0.70	0.50	0.58
T+CONN+IOB+BMV +Arg2+Prev+Morph	**0.73**	**0.50**	**0.60**
T+CONN+IOB+BMV+Prev +Morph+PoS+Arg2	0.72	0.51	0.59
Token+CONN+IOB+PoS+Prev +Morph+MV+BMV+Arg2	0.69	0.50	0.58
Feature Ablation			
T+CONN+IOB+BMV+Morph+Prev	0.70	0.44	0.54
T+CONN+IOB+BMV+Prev+Arg2	0.70	0.50	0.58
T+CONN+IOB+BMV+Morph+Arg2	0.69	0.38	0.50
T+CONN+IOB+Prev+Morph+Arg2	0.72	0.51	0.60
T+CONN+BMV+Morph+Prev+Arg2	0.69	0.46	0.55
T+IOB+BMV+Morph+Prev+Arg2	0.62	0.36	0.45
CONN+IOB+BMV+Morph+Prev+Arg2	0.70	0.50	0.59

Table 6: Results with Single and Combined Features for `Arg1`

2.7.3.2 Notes on Hill-Climbing Feature Selection & Feature Ablation Test

Hill Climbing Feature Selection Test.

Discussion: Hill Climbing Algorithm

1. The hill-climbing search algorithm (steepest-ascent version) is shown in

Algorithm 1 Hill Climbing Algorithm (Uphill) [93]

```
function **Hill Climbing** (problem)
current ← MAKE-NODE(problem, INITIAL-NODE)
**do** neighbor ← a highest-valued successor of current
**if** neighbor · VALUE < current · VALUE **then**
    return current.STATE
**end if**
current ← neighbor
**EndDo**
**Return:** local-maximum·STATE
```

Algorithm 1. It is simply a loop that continually moves in the direction of increasing (uphill) value. It terminates when it reaches a ”peak” where no neighbor has a higher value. The algorithm does not maintain a search tree, so the data structure for the current node need only record the state and the value of the objective function. Hill climbing does not look ahead beyond the immediate neighbors of the current state.

2. Hill climbing is sometimes called greedy local search because it grabs a good neighbor state without thinking ahead about where to go next. Hill climbing often makes rapid progress toward a solution because it is usually quite easy to improve a bad state.

3. Hill climbing often gets stuck for the following reasons:

a. Local maxima: a local maximum is a peak that is higher than each of its neighboring states but lower than the global maximum. Hill-climbing algorithms that reach the vicinity of a local maximum will be drawn upward toward the peak but will then be stuck with nowhere else to go.

b. Ridges: ridges result in a sequence of local maxima that is very difficult for greedy algorithms to navigate

c. Plateau: a plateau is a flat area of the state-space landscape. It can be a flat local maximum, from which no uphill exit exists, or a shoulder, from which progress is possible. A hill-climbing search might get lost on the plateau.

Feature Ablation Test.

(Source: `http://aclweb.org/aclwiki/index.php?title=RTE_Knowledge_Resources`)

An ablation test consists of removing one module at a time from a system, and rerunning the system on the test set with the other modules, except the one tested.

Ablation test are meant to help better understand the relevance of the knowledge resources used by the systems, and evaluate the contribution of each of them to the systems' performances. In fact, comparing the results achieved in the ablation tests to those obtained by the systems as a whole allows assessing the contribution given by each single resource.

In RTE challenges, from 2009 (RTE-5) onwards, this is an important evaluation measure to run.

This measure helps to test the robustness of the best decision found through Hill-climbing feature selection.

2.7.3.3 Result

		P	R	F1
Arg2	**Exact**	**0.83**	**0.75**	**0.79**
	Partial	0.93	0.84	0.88
	Overlap	0.97	0.88	0.92
Arg1 +Prev	**Exact**	**0.70**	**0.48**	**0.57**
	Partial	0.83	0.62	0.71
	Overlap	0.91	0.63	0.74
Arg1 -Prev	**Exact**	**0.70**	**0.38**	**0.50**
	Partial	0.83	0.49	0.62
	Overlap	0.92	0.50	0.65

Table 7: Results of `Arg1` and `Arg2` extraction with test dataset.

In Table 7 we report results for each parsing subtask Precision, Recall and F1 achieved with the best performing feature set (see Section 2.7.3.1) using the test split, with the corresponding baseline between parenthesis. Note that before evaluation, all spans were normalized by removing leading or trailing punctuation. The best results and features are highlighted in Table 5 and 6 for `Arg2` and `Arg1` respectively.

We compute the confidence intervals using a resampling method [94]. For `Arg1` identification, we observe that the confidence interval (95%) without "Prev" feature ranges from 0.48 to 0.52 and the same interval is between 0.55 and 0.59 with "Prev" feature, if the exact F1 measure is taken into account. For `Arg2` identification the confidence interval (95%) is between 0.78 and 0.81, when the exact F1 measure is taken into account. A statistical significance test run on previous and current results of `Arg1` identification shows also that the difference is significant ($p < 0.0001$) [13].

We observe in the results that recall is consistently lower than precision in all tables. This is probably due to the fact that CRF is more conservative while tagging data with argument label compared to other classifiers, which may lead to a lower coverage.

As expected, `Arg2` parsing subtask achieves a better performance than `Arg1` subtask because `Arg2` position and extension are easier to predict. This is confirmed by the fact that the baseline precision of `Arg2` *overlap* is 0.98. Also, the major improvement w.r.t. the baseline is achieved in the *exact* setting.

2.7.3.4 Post-hoc Analysis

We carry out a further analysis on the test set in order to characterize parser errors on different test set partitions. Since `Arg1` may occur in a previous sentence w.r.t. the connective, we want to assess the impact of `Arg1` position on the parsing task. Therefore, we separately evaluate `Arg1` precision, recall and F1 on intra-sentential and inter-sentential discourse relations. Results are reported in Table 8. We also show the changes before and after adding the lexical feature targeting inter-sentential cases.

The "Prev" feature is critical to the parser to achieve reasonable baseline `Arg1` performance for the inter-sentential partition of the test set.

We also carry out a comparative analysis of the parsing performance in the *exact* evaluation setting by considering separately coordinating, subordinating and adverbial connectives. We make the above-mentioned distinction following the suggestion by Elwell et al (2008) [69], because each connective type has a different behavior w.r.t. its arguments: coordinating connectives (e.g. *and*, *but*) usually have syntactically similar arguments, subordinating

[13] For the significance of difference the permutation test is used, whereas to compute the confidence interval bootstrap resampling is used. (Hjorth 1993) [94]. Throughout this thesis-work we determined the significant digits for presenting results using the methods illustrated by Weisstein E. W. [95]

		Arg1-Results		
		P	R	F1
Intra-Sentential w/o Prev_feat	Exact	0.73	0.61	0.66
	Partial	0.86	0.77	0.81
	Overlap	0.95	0.78	0.86
Inter-Sentential w/o Prev_feat	Exact	0.19	0.01	0.02
	Partial	0.27	0.02	0.04
	Overlap	0.31	0.02	0.04
Intra-Sentential with Prev_feat	Exact	0.77	0.61	0.68
	Partial	0.88	0.79	0.81
	Overlap	0.96	0.77	0.85
Inter-Sentential with Prev_feat	Exact	0.52	0.27	0.36
	Partial	0.68	0.40	0.50
	Overlap	0.79	0.40	0.54

Table 8: Results of `Arg1` parsing for intra- and inter- sentential partitions. In the test set, the number of intra- and inter- sentential relations are 1028 and 617 respectively.

ones (e.g. *since*, *before*) are dominated or adverbially linked to `Arg1` and are syntactically bound to `Arg2`, while adverbial connectives (i.e. *nevertheless*, *for instance*) can occur in different positions in the sentence and are not necessarily bound to `Arg1`.

The evaluation results are presented in Table 9.

In previous works, e.g. Elwell et al (2008) [69], adverbial connectives were usually considered the most difficult connective type to classify. This is confirmed by our results obtained on `Arg1`, which show that adverbial connectives negatively affect both precision and recall, with a higher impact on recall. As for `Arg2`, the parsing results on the three connective types are more homogeneous.

We also observe that the "Prev" feature significantly improves `Arg1` parsing with any connective type because it increases recall, while precision decreases with coordinating and adverbial connectives.

We perform another level of error analysis on top of the error analysis results for specific connective types in Table 9. Results are reported in the Table 10. This analysis is done in order to understand the contribution of each connective type to inter vs. inter-sentential classifications. In Table 9 we observed that adverbial connective type was the most difficult case to classify,

Conn. Type	P	R	F1
Results for Arg2			
Coordinating	0.81	0.75	0.78
Subordinating	0.86	0.78	0.82
Adverbial	0.83	0.74	0.78
Results for Arg1(w/o Prev)			
Coordinating	0.73	0.42	0.54
Subordinating	0.73	0.45	0.56
Adverbial	0.68	0.26	0.37
Results for Arg1 (with Prev)			
Coordinating	0.69	0.59	0.64
Subordinating	0.76	0.50	0.61
Adverbial	0.64	0.34	0.44

Table 9: Exact evaluation for each connective type. Coordinating connectives appear in around 40% of the relations, while subordinating and adverbials are respectively 25% and 35% of all connectives.

but we note in the Table 10 that indeed results did not improve much even after adding the PREV feature to the feature set. PDTB statistics shows that among all the connective types, the **Arg1** for adverbial connective, resides in the sentence other than the connective sentence. For the coordinating cases, the PREV feature is useful because in most cases **Arg1** resides in the same sentence as connective. The subordinating connective falls between these two extremes.

In order to understand the most common mistakes done by the classifier, we present two example relations where resp. **Arg1** (a) and **Arg2** (b) are wrongly identified[14]. Note that in example (b) **Arg1** appears in the previous sentence, which we do not report here.

(a) Many analysts said the September increase was a one-time event, *coming* as **dealers introduced their 1990 models** [Contingency]

(b) However, Jeffrey Lane, president of Shearson Lehman Hutton, said **that Friday's plunge is "going to set back" relations with customers**, "because it reinforces the concern of volatility [Comparison]

[14] The examples show the gold standard annotation.

			Arg1+PREV			Arg1-PREV		
			P	R	F1	P	R	F1
Coordinating	Intra-Sentential	Exact	0.75	0.70	0.72	0.73	0.69	0.71
		Partial	0.88	0.88	0.88	0.86	0.89	0.88
		Overlap	0.96	0.90	0.93	0.96	0.91	0.94
	Inter-Sentential	Exact	0.54	0.42	0.47	0.35	0.02	0.03
		Partial	0.72	0.64	0.68	0.38	0.02	0.05
		Overlap	0.84	0.66	0.74	0.41	0.02	0.05
Subordinating	Intra-Sentential	Exact	0.69	0.56	0.62	0.66	0.57	0.61
		Partial	0.88	0.79	0.79	0.86	0.74	0.78
		Overlap	0.94	0.76	0.84	0.94	0.80	0.86
	Inter-Sentential	Exact	0.57	0.26	0.35	0.12	0.02	0.03
		Partial	0.70	0.35	0.46	0.15	0.02	0.04
		Overlap	0.79	0.35	0.47	0.16	0.02	0.04
Adverbial	Intra-Sentential	Exact	0.63	0.50	0.56	0.63	0.52	0.57
		Partial	0.82	0.64	0.72	0.80	0.68	0.74
		Overlap	0.88	0.70	0.78	0.88	0.72	0.79
	Inter-Sentential	Exact	0.52	0.18	0.27	0.25	0.01	0.02
		Partial	0.66	0.26	0.37	0.44	0.02	0.05
		Overlap	0.77	0.26	0.39	0.50	0.02	0.05

Table 10: Results of Arg1 parsing for intra- and inter- sentential partitions w.r.t the connective types

In (a), the classifier tagged the whole text from “the September” to “coming” as Arg1 instead of only “coming”, since it takes clausal boundaries as a relevant factor for identifying the argument spans. In (b) the classifier is unable to detect Arg2 probably because the argument does not occur immediately next to the connective.

A manual inspection of misclassified relations confirms that the parser is more accurate in the identification of the sentences containing the arguments rather than in the detection of their exact spans. Also, mistakes concern mostly the classification of inter-sentential relations (especially as regards the Arg1 classifier), thus we will need to focus on these specific cases for future improvements.

2.7.4 A Comparison: Non-Cascaded Prediction

So far we were following a cascaded pipeline architecture to predict the discourse structures. Now we also attempt to investigate a non-cascaded architecture where the parser predicts both the `Arg1` and `Arg2` at the same time, given the gold-standard connectives. We used the same best feature-set (i.e. {token, connective-sense, boolean main verb, IOB chain, lemma, inflection, previous sentence feature}) of 7 features, optimized for `Arg1`, except the `Arg2` labels as feature. Now the set of labels for prediction is either `Arg1` (with -B, -I, -E tags) or `Arg2` (with -B, -I, -E tags) or the 'O' for the labels other than `Arg1` or `Arg2`.

As the classifier we still use CRF++; we change the label set to predict – before it was $\{ARG1-B, ARG1-I, ARG1-E, O\}$ for `Arg1` prediction and was $\{ARG2-B, ARG2-I, ARG2-E, O\}$ for `Arg2` prediction, now our label set is $\{ARG1-B, ARG1-I, ARG1-E, ARG2-B, ARG2-I, ARG2-E, O\}$. The feature set is already described.

		P	R	F1
Arg2	**Exact**	**0.81**	**0.74**	**0.77**
	Partial	0.92	0.86	0.89
	Overlap	0.97	0.89	0.93
Arg1	Exact	**0.65**	**0.42**	**0.51**
	Partial	0.79	0.59	0.68
	Overlap	0.91	0.60	0.72
Overall	Exact	**0.75**	**0.59**	**0.66**
	Partial	0.87	0.73	0.80
	Overlap	0.95	0.75	0.84

Table 11: Results of `Arg2`, `Arg1` and `Overall` through non-cascaded classification system with gold-labeled data.

We notice the results of non-cascaded parser in Table 11. If we compare the score for exactly matched arguments in Table 11 to the results we achieved for cascaded prediction in Table 7, it is evident that the cascaded architecture gives better results for both the argument classification: in case of `Arg1` we get improvement by almost 6 points in terms of F1 whereas in case of `Arg2` we get improvement by almost 2 points with F1 measure. This non-cascaded system is also suffers from unbalanced precision and recall (i.e. high precision and low recall) like in the case of cascaded system. We also computed the overall score for the argument versus non-argument classification that shows a 0.66 exact F1 measure. One very interesting observation: though the cascaded system wins with more scores in terms of exact matches for both the arguments, in case of partial and overlap match measures for `Arg2` the non-cascaded system slightly performs better than the cascaded one; may be this is due to the trade-off balancing effect between the minimization of errors and maximization of the number of filtered assignments in a cascaded scenario.

2.8 Conclusion

We cast the complex task of discourse argument parsing as a set of cascading subtasks to be tackled in sequence, and we showed that in this way we achieved a reasonable parser accuracy by handling the whole labeling process in a pipeline.

Since we consider this discourse parsing task as a token-level sequence-labeling task, we were able to detect connective arguments and the corresponding boundaries avoiding the computationally complex approaches described in previous works.

We trained a CRF classifier with lexical, syntactic and semantic features extracted from PDTB and PTB gold annotation. We tested these features both in isolation and in different combinations in order to achieve an optimized performance. To make training time manageable, we pruned the search space by 90%, though leaving out only around 5% of all `Arg1` in PDTB.

We also presented a comparative error analysis (subsection 2.7.3.4), where we showed that `Arg1` classification on intra-sentential relations achieves a performance comparable to `Arg2` classification (Table 7). Since the main open issue in our approach is the correct classification of `Arg1` in inter-sentential relations, we plan to improve it through more feature engineering. We already extended our experimental framework by including automatically annotated parse trees and connectives in the pipeline [96].

Finally, we made an attempt to compare the performance of cascaded structured prediction with the non-cascaded scenario. There we make more complex structured prediction putting `Arg1`, `Arg2` together to investigate the interaction and results. We found our cascaded model works better in case of the exact match evaluation. For the other softer matches the performance of `Arg2` degrades a little; whereas with cascaded prediction model, the `Arg1` classification always performs much better than the non-cascaded prediction model.

3 End-to-End Discourse Parsing

3.1 Introduction

We have presented a gold label discourse parser in the previous chapter. Now we present a novel end-to-end discourse parser that, given a plain text document in input, identifies the discourse relations in the text, assigns them a semantic label and detects discourse arguments spans. The parsing architecture is based on a cascade of decisions supported by Conditional Random Fields (CRF). We train and evaluate three different parsers using the PDTB corpus. The three system versions are compared to evaluate their robustness with respect to deep/shallow and automatically extracted syntactic features.

We have implemented an end-to-end system for discourse parsing that also works in a cascaded pipeline like in the second chapter, dividing the whole complex task into smaller subtasks. We sequentially classify discourse arguments with boundaries by focusing on one relation at a time.

3.2 Processing pipeline for End-to-End system

We present the entire pipeline structure in Figure 4, where the different processing modules and input data sources are displayed. We report the overall workflow, distinguishing between automatic and gold labeled data. In particular, dotted lines denote gold-labeled system input.

We train the model with automatically extracted features (in the automatic setting). These features are basically obtained with the feature selection done with gold-standard data (for the gold-standard setting).

In the semi or fully automatic setting, four modules are involved in the following order:

1. Parser module: The Stanford parser (version 1.6.4) (Klein et al 2003) [97] is used to parse an input document.

2. Module for Connective Detection and Classification: The AddDiscourse tool (Pitler & Nenkova 2009) [23] takes the syntactic parse tree as input from module one and tags explicit discourse connectives with sense labels at class level.

3. `Arg2` tagging module: in the automatic setting, it takes input from the first and second module i.e. features extracted from automatic parse

trees, and automatic connectives.

4. `Arg1` tagging module: in the fully automatic setting, it takes input from the first and second module, and classified `Arg2` labels from the third module. In the semi-automatic setting, it takes input from the first and second module only, and `Arg2` tag labels are extracted from PDTB.

In the gold standard setting, the output of Module 1 is replaced by PTB parse trees, while the annotation of Module 2 is replaced by gold connective annotation from PDTB; `Arg2` tag labels used to classify `Arg1` are extracted from PDTB.

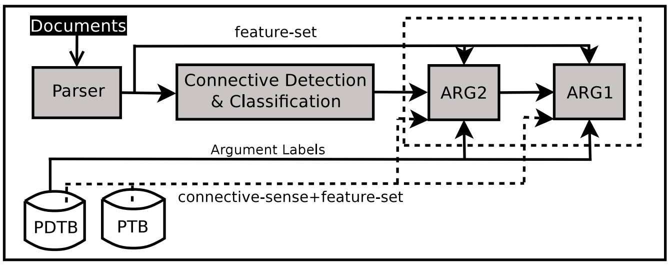
Documents
feature-set
Parser
Connective Detection & Classification
ARG2
ARG1
Argument Labels
connective-sense+feature-set
PDTB
PTB

Figure 4: Overall Diagram of Pipeline of End-to-End Discourse Parser. Dotted lines encompass gold standard input. Gray boxes are process modules

Before describing the experiments and results with end-to-end parser (cf. Section 3.4) we briefly illustrate the experiments done and results obtained with the AddDiscourse tool with automatically parsed WSJ [62] raw documents (cf. Section 3.2.1). We also compare the performance of Stanford parser with that of the other state-of-the-art parsers (cf. Section 3.2.2).

3.2.1 AddDiscourse Tool for Connective & Sense Detection

The AddDiscourse tool is built to automatically identify explicit discourse connectives and their *surface* senses (i.e. Expansion, Contingency, Comparison, Temporal, though it also outputs a non-discourse usage tag with "O" that we ignored in this work). It takes PTB-style syntactic parse trees as input and outputs augmented trees with tags for each discourse connective.

Pitler & Nenkova (2009) [23] report results on ten-fold cross-validation over sections 02-22 of the PDTB using a Naive Bayes classifier.

Here we illustrate the performance of AddDiscourse with WSJ 00-01 folders (i.e. our development split) and WSJ 23-24 folders (i.e. our test split). We also evaluated WSJ 22, as sometimes in parsing community that folder is included in the test data split. We also compute the micro and macro average on these folders. Additionally, we evaluated the performance of AddDiscourse tool for the test split only. We obtained precision 0.88%, recall 94% and F1-measure 0.91%, as already presented in Ghosh et al (2011a) [96]

	Precision	Recall	F1	Accuracy
Micro-Avg	0.830	0.940	0.882	0.937
Macro-Avg	0.565	0.638	0.593	0.945
Sections	Precision	Recall	F1	Accuracy
WSJ00	0.555	0.631	0.585	0.939
WSJ01	0.649	0.718	0.674	0.947
WSJ22	0.592	0.643	0.613	0.952
WSJ23	0.653	0.739	0.687	0.943
WSJ24	0.377	0.460	0.409	0.944

Table 12: *Results of connective classification using AddDiscourse tool with Automatic Parse trees on WSJ raw text.*

We also performed the connective-type based analysis. As found by Pitler & Nenkova (2009) [23], the worst performing connectives are TEMPORAL connectives, which are correctly labled only 20% of the cases.

3.2.2 Comparison: Stanford Parser vs. Other Syntactic Parsers

We use Stanford parser for syntactic parsing of sentences. In the following, we compare its performance to the other performing parsers.

The syntactic structure of a sentence is crucial to understand the discourse structure in language. Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the most likely analysis of new sentences. Statistical parsers still make mistakes, but generally work rather well. Their development was one of the biggest breakthroughs in natural language processing in the 1990s.

The problem of mapping a string of words to its parse tree is called syntactic parsing. It is the most commonly used mathematical system for modeling constituent structure in English and other languages (The fundamental idea of constituency is that groups of words may behave as a single unit or phrase).

Following the definition by Hopcroft et al (2000) [98], we also define a context-free grammar G as a 4-tuple (N, Σ, A, R), where N is a set of nonterminal symbols, Σ is an alphabet, A is a distinguished start symbol in N, and R is a finite set of rules. Each rule in R is of the form $X \rightarrow \beta$ for some $X \in N$, $\beta \in (N \cup \Sigma)^{\star}$. The grammar defines a set of possible strings in the language and it also defines a set of possible leftmost derivations under this grammar. Each derivation corresponds to a tree-sentence pair that is well formed under that grammar.

As defined in (Collins 2003) [29]: a probabilistic context-free grammar is a simple modification of a context-free grammar in which each rule in the grammar has an associated probability $P(\beta|X)$. This can be interpreted as the conditional probability of Xs being expanded using the rule $X \rightarrow \beta$, as opposed to one of the other possibilities for expanding X listed in the grammar. The probability of a derivation is then a product of terms, each term corresponding to a rule applied in the derivation. The probability of a given tree-sentence pair (T, S) derived by n applications of context-free rules $LHS_i \rightarrow RHS_i$ (where LHS stands for "left-hand side", RHS for "right-hand side"), $1 \leq i \leq n$, under the $PCFG$ is

$$P(T, S) = \prod_{i=1}^{n} P(RHS_i|LHS_i) \tag{4}$$

In (Booth1973) [99], the conditions are specified under which the PCFG does in fact define a distribution over the possible derivations (trees) generated by the underlying grammar: (1) the rule probabilities define conditional distributions over how each nonterminal in the grammar can expand (2) a technical condition that guarantees that the stochastic process generating trees terminates in a finite number of steps with probability one.

One of the problems in PCFGs is to define the conditional probability $P(\beta|X)$ for each rule $X \rightarrow \beta$ in the grammar. A common and simple way to achieve this is to take counts from a treebank and then to use the maximum-likelihood estimates:

$$P(\beta|X) = \frac{Count(X \rightarrow \beta)}{Count(X)} \tag{5}$$

When the model is trained, we obtain a model that defines $P(T, S)$ for any sentence-tree pair in the grammar. The output on a new test sentence S is the most likely tree under this model is computed as follows:

$$T_{best} = argmaxP(T|S) = argmax_T \frac{P(T,S)}{P(S)} = argmaxP(T,S) \qquad (6)$$

A parser itself is an algorithm that searches for the tree, T_{best} , that maximizes $P(T,S)$. In the case of PCFGs, this can be accomplished using a variant of the CKY algorithm applied to weighted grammars (providing that the PCFG can be converted to an equivalent PCFG in Chomsky normal form) (Manning et al 1999) [100].

We present here a comparative study of state-of-the-art parsers in literatures, focusing on lexicalized Probabilistic Context Free Grammar (PCFG) parsers, because these are the state-of-the-art parsers. A PCFG can be lexicalized by associating a word w and a part-of-speech (POS) tag t with each nonterminal X in the tree.

(Magerman 1995) [101] describes a history-based approach which uses decision trees to estimate $P(T|S)$. The problem of this models is: it uses sophisticated n-gram estimation methods, and conditions on richer history than just surface distance.

The model in (Collins 1996) [102] show that the distance between words standing in head-modifier relationships is important; in particular, it is important to capture a preference for right-branching structures (which almost translates into a preference for dependencies between adjacent words) and a preference for dependencies not to cross a verb. In (Collins 1999) [103] made some refinements on (Collins 1996) [102] about rules made and also introduced three modeling schemes, and improved performance.

In (Charniak 1997) [104] we find a better result than (Collins 1996) [102]. It conditions more over lexical heads than the previous works. We also find that using the parent type to condition the probability of a rule made more leverage over the previous parsers.

In case of the Stanford lexicalized PCFG parser (Klein et al 2003) [97], it implements a factored product model, with separate PCFG phrase structure and lexical dependency experts, whose preferences are combined by efficient exact inference, using an $A^\star$ algorithm (Book: Russel & Norvig 2003) [93].

We present in Table 13 the performances of the state-of-the-art parsers. The results are taken from the respective papers on standardized splits of WSJ [62] corpus.

Parser	LP	LR	F1	CB	0 CB
Magerman'95	84.9	84.6	84.7	1.26	56.6
Collins'96	86.3	85.8	86.0	1.14	59.9
Charniak'97	87.4	87.5	87.4	1.00	62.1
Collins'99	88.7	88.6	88.6	0.90	67.1
Klein & M'03	86.9	85.7	86.3	1.10	60.3

Table 13: *Results Comparison for State-of-the-art Lexicalized Parsers:* LR: Labeled Recall, LP: Labeled Precision; F1: Harmonic mean of LP and LR; CB: average number of crossing brackets per sentence; 0CB: $<$ 2 CBs, the percentage of sentences with 0 or $<$ 2 crossing brackets respectively.

The LP, LR and CB are measures are defined as was defined in PARSEVAL. PARSEVAL (Black 1991) [105] measures are given as follows:

$$LP = \frac{\text{number of correct constituents in proposed parse}}{\text{number of constituents in proposed parse}}$$

$$LR = \frac{\text{number of correct constituents in proposed parse}}{\text{number of constituents in treebank parse}}$$

Crossing Brackets (CB): number of constituents violate constituent boundaries with a constituent in the treebank parse

Thus from the above discussion we notice that the performance of Stanford parser comparable with the other state-of-the-art parser, so we use this parser for our purpose.

3.2.3 Other Tools/ Scripts

In the processing pipeline, we also used other NLP tools to extract some features.

The Perl script chunklink.pl serves to convert Penn Treebank II files into a one-word-per-line format containing (at least) the same information as the original files. This script was used to generate the data for the CoNLL-2000 Shared Task. We generated the IOB chain with this perl script.

Morpha Stemmer. (Minnen et al 2001) [75] A fast and robust morphological analyzer for English based on finite-state techniques that returns the lemma and inflection type of a word, given the word form and its part of speech.

We implemented java libraries to extract head informations from syntactic parse trees using head rules by Yamada and Matsumoto[15].

3.3 Feature List

We list all features considered in the classification task in Table 14, described with an example sentence in Fig. 5. We started with a short intuitively selected feature list which is optimized by hill-climbing and feature ablation techniques. The final feature set for `Arg2` classification is $\{F1, F2, F3\}$ whereas all the features are used for `Arg1` classification. Note that IOB chain is the list of the syntactic categories of all constituents on the path between the root node and the current leaf node of the parse tree. Boolean feature BMV states whether a candidate token is a main verb of the main clause or not.

It is to be noted that we use the same (optimized) feature set that is used for the best gold-label system performance in Chapter 2. Finally we also attempt to analyze and optimize the primary feature-set (i.e. Table 31 in Chapter 2) for the fully automatic system. We use a development split WSJ (00-01) folders of raw text [62] for the feature analysis only.

Features used for `Arg1` and `Arg2` classification	
F1.	Token (T)
F2.	Sense of Connective (CONN)
F3.	IOB chain (IOB)
F4.	PoS tag
F5.	Lemma (L)
F6.	Inflection (INFL)
F7.	Boolean feature for MV (BMV)
F8.	Previous Sentence Feature (PREV)
Additional feature used only for `Arg1`	
F9.	`Arg2` Labels

Table 14: Feature sets for `Arg1` and `Arg2` classification (MV: Main verb of main clause)

[15] See for reference `www.jaist.ac.jp/\~h-yamada/`

3.4 Experiments

3.4.1 General Setup

The documents of PDTB folders $02-22$ are used to train the model, while documents of folders 23 and 24 are meant for testing (finally, we also attempt the feature analysis in automatic settings; for this PDTB folders $00-01$ are used). In order to extract the syntactic features based on parse tree information and to pair them with discourse information, we align the parse trees (either automatic or extracted from PTB) with PDTB sentences at token level. We train the model with gold-argument labels from PDTB, which is not needed during decoding. In this way, we can decode any document using the trained model, the automatic parser and the AddDiscourse [23] tool.

While preparing the data, if there exist n relations in a document, we take into account one relation at a time per document, and we repeat the document n times with all related features. This results in a large search space for the discourse parser. We prune off the search-space by around 90% including a window of two sentences before and after the given connective. This representation of discourse tries to preserve the relevant contextual information for the arguments.

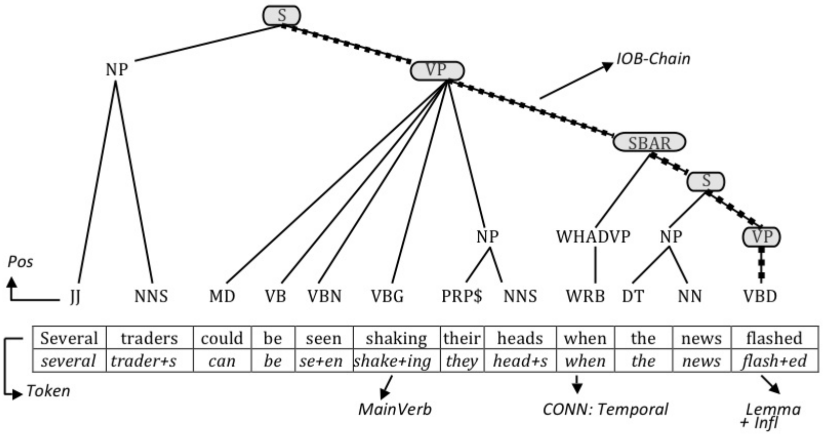

Figure 5: PTB parse tree with system features

We use the CRF++ tool [16] for sequence labeling classification [77], with

[16] CRF++ written by Taku Kudo can be downloaded from `crfpp.sourceforge.net`

unigram and bigram feature representation capabilities. Beside the individual specification of a feature in the feature description template, the features in various combinations are also represented.

3.5 Results

In Table 15 we report the classifier performance on gold labeled data. The corresponding baseline scores are shown in parenthesis. **Arg2** baseline is computed by labeling all tokens of the text span between the connective and the beginning of the next sentence as **Arg2**. The **Arg1** baseline is computed by labeling all tokens in the text span from the end of the previous sentence to the connective position. If the connective occurs at the beginning of a sentence, then the baseline classifier tags the previous sentence as **Arg1**.

		P	R	F1
Arg2	**Exact**	**0.834** (0.53)	**0.751** (0.46)	**0.791** (0.49)
	Partial	0.932 (0.80)	0.842 (0.85)	0.886 (0.82)
	Overlap	0.972 (0.98)	0.875 (0.85)	0.921 (0.91)
Arg1	Exact	0.699 (0.19)	0.485 (0.19)	0.573 (0.19)
	Partial	0.829 (0.50)	0.616 (0.68)	0.707 (0.58)
	Overlap	0.910 (0.70)	0.632 (0.68)	0.746 (0.69)

Table 15: Results of **Arg1** and **Arg2** Classification Using Gold-label data. Baseline between parenthesis.

Arg2 classification shows about 80% exact $F1$ measure, whereas the inter-annotator agreement score in PDTB is 90.2% for both arguments. The different performance in the classification of **Arg1** and **Arg2** is also noticeable, though we improved the previous performance (old F1-measure:0.49) with Previous sentence feature (cf. Second Chapter).

*It is to be noted here that we improved the performance of Ghosh et al (2011a) [96] considerably including the Previous sentence feature. We already know that this novelty feature is a "connective-surface" feature that is used to capture if the following sentence begins with a connective. In this Chapter in the Table 21 we illustrate and later discuss on how the inter-sentential **Arg1** detection improved the overall performance to the current performance. In the past (Ghosh et al 2011b [106]) confirmed only about 1% of inter-sentential **Arg1** detection without using this feature.*

3.5.1 Fully Automatic System

		P	R	F1
Arg2 Automatic	**Exact**	**0.755**	**0.591**	**0.663**
	Partial	0.902	0.691	0.783
	Overlap	0.942	0.739	0.828
Arg1 Semi-automatic	Exact	0.661	0.369	0.474
	Partial	0.785	0.473	0.591
	Overlap	0.865	0.484	0.621
Arg1 Automatic	Exact	0.655	0.349	0.456
	Partial	0.772	0.447	0.566
	Overlap	0.854	0.455	0.593

Table 16: Results of `Arg2` and `Arg1` classification with automatic parse trees and connectives. Variation of `Arg1` results with Gold & Automatic ARG2 labels.

In Table 16 we compare the classification performance using semi-automatic and fully-automatic settings. The semi-automatic system includes a gold-standard `Arg2` feature for training in Module 4 (i.e. `Arg1` labeling, Fig. 4), whereas in the fully-automatic system the same feature is generated based on labels automatically assigned by Module 3. In Table 16, we observe that the performance on (semi-& full-) automatic classification is lower than the gold-labeled one.

3.5.2 Impacts in Mixed System Settings

We perform further analyses to see whether the mistakes are introduced by automatic syntactic parse trees or automatically generated connectives. We first classify arguments using PTB trees and automatic connectives (in Table 18), and then using Stanford parse trees and gold-label connectives (in Table 17).

The overall result of Table 18 is better than that of Table 17. This proves that automatic discourse parsing depends more on richer parse tree features, while the connective feature is less relevant.

In all tables, we observe that recall is consistently lower than precision. This is probably due to the fact that CRF is more conservative while tagging

		P	R	F1
Arg2	**Exact**	**0.765**	**0.613**	**0.681**
	Partial	0.912	0.734	0.813
	Overlap	0.960	0.770	0.854
Arg1	Exact	0.649	0.396	0.492
	Partial	0.778	0.514	0.619
	Overlap	0.862	0.526	0.654

Table 17: Impact of Automatic Syntactic Parses (with Gold-label Connectives).

		P	R	F1
Arg2	**Exact**	**0.825**	**0.704**	**0.760**
	Partial	0.934	0.786	0.854
	Overlap	0.961	0.830	0.891
Arg1	Exact	0.678	0.444	0.536
	Partial	0.803	0.550	0.653
	Overlap	0.869	0.570	0.688

Table 18: Impact of Automatic Connectives (with Gold-label Syntactic Parses).

data with argument label compared to the other classifiers, which may lead to a lower coverage.

3.5.3 Impact of using Shallow features

In order to remove the strong dependency of the model on deep-syntactic features, we replaced the IOB chain feature with two pairs of shallow syntactic features that mark starting and ending of clausal boundaries. On the top of the paper work by Ghosh et al (2011b) [106] we also included three more features based on the two pairs of boolean feature values that enhanced our lightweight system further more. Therefore we replace the IOB chain feature with seven features in total.

We briefly depict the boolean features. For example, the *i*-th token in one sentence has the value of "*I-SINV(1)/E-NP(2)/E-NP(3)/C-NP(4)*" as the IOB chain feature, where we mark the levels of the SPT (syntactic parse tree) with the bracketed numbers to ease the discussion. The first boolean feature looks for whether at the second level it is starting with B-, if yes then it is valued as "B1" otherwise it gives a zero value; the second boolean feature looks for whether at the third level it is starting with B-, if yes then it is valued as "B2" otherwise it gives a zero value; the third boolean feature looks for whether at the second level it is starting with E-, if yes then it is valued as "E1" otherwise it gives a zero value; the boolean feature looks for whether at the third level it is starting with E-, if yes then it is valued as "E2" otherwise it gives a zero value. The fifth feature computes a value like "B1_B2" if both the third and fourth features have the values other than zeros. The sixth feature computes a value like "E1_E2" if both the first and second features have the values other than zeros. The seventh feature is computed from the fifth and sixth feature if one of them or both of them is non zero like "E1_E2_0". In the case of our example the feature set will look like: $\{0, 0, E1, E2, 0, E1\_E2, 0\_E1\_E2\}$.

In this way, we devise a lightweight version of the system, with a reduction in training time by more than 50%.

We report in Table 19 the results obtained with the lightweight system using gold labeled data, which should be compared with those reported in Table 15, obtained with the same data type but using the full system version.

All these results under this section are re-computed using the previous sentence "PREV" feature, whereas in Ghosh et al (2011a) [96] this feature was unused.

		P	R	F1
Arg2	**Exact**	**0.812**	**0.729**	**0.768**
	Partial	0.919	0.845	0.881
	Overlap	0.971	0.872	0.919
Arg1	Exact	0.646	0.431	0.517
	Partial	0.792	0.610	0.689
	Overlap	0.913	0.615	0.738

Table 19: Lightweight System Results with Gold parse trees and connectives.

We also report in Table 20 the results obtained with the lightweight system version using automatic parse trees and automatic connective data, which should again be compared with the deep-syntactic system evaluation in Table 16 (automatic `Arg1` and `Arg2`).

Overall *exact* results of the lightweight system are lower than the results

		P	R	F1
Arg2	**Exact**	**0.739**	**0.577**	**0.648**
	Partial	0.904	0.711	0.796
	Overlap	0.970	0.757	0.850
Arg1	Exact	0.611	0.316	0.416
	Partial	0.769	0.459	0.574
	Overlap	0.899	0.465	0.613

Table 20: Lightweight System Results with Automatic parse trees and connectives.

computed with deep-syntactic features, whereas *partial* and *overlap*-based results show a better trend. We may improve this with more feature engineering.

3.6 Error Analysis

Some typical errors made by the classifier involve the correct detection of `Arg2` boundaries in case of double connectives, as in the example below (shown as in gold-data):

(a) *But* <u>as</u> **panic spread** *speculators began to sell blue-chip stocks such as Philip Morris and International Business Machines* [TEMPORAL]

The classifier does not include *But* in `Arg2` span, probably because it is a connective itself, even if it is not involved in the temporal relation annotated here and depending on *as*.

Another typical example is the following, in which only *approvals required for development* has been classified as `Arg1` (we report in the example below the correct classification).

(b) **The land to be purchased by the joint venture hasn't yet received zoning and approvals required for development**, <u>and</u> *part of Kaufman & Broad's job will be to obtain such approvals* [EXTENSION]

In this case, `Arg1` span is not correct because the classifier may have interpreted the token *and* preceding *approvals* as the beginning of a clause.

3.6.1 Impact: Intra- vs. Inter-Sentential Arg1 for Automatic System

In the paper by (Ghosh et al 2011b [96]) it is already noted that that system is able to detect only around 1% of inter-sentential `Arg1` while we know that 39% of `Arg1` are inter-sentential. Moreover overall 30% of `Arg1` are in the previous sentence.

We noticed a huge improvement adding only one simple feature viz. "Prev", the previous sentence feature (discussed under the Section "Feature Set") with the gold-standard system. Therefore we added the feature to the full automatic setting, and observed a notable improvement for this setting, which considerably reduced the gap of the performance between gold-standard and automatic settings, as well reduced the gap between the results of the automatic and the semi-automatic settings. Now we want to assess the impact of `Arg1` position on the parsing task in the semi- and full- automatic settings.

We observe from the figures in Table 21 that, in semi-automatic settings, the inter-sentential `Arg1` detection is more effective than that in the fully-automatic one. This is because of the error propagation through the cascaded pipeline: in the fully automatic settings we use the generated `Arg2` label, whereas the semi-automatic one uses the gold-labeled `Arg2` label features.

From the figures of intra-sentential `Arg1` for both the settings it is evident that the performance of the full-automatic `Arg1` detection is still depending considerably on intra-sentential `Arg1` detection, but it is not so in the case of semi-automatic system. For a semi-automatic system inter-sentential `Arg1` detection strongly contributes to the overall `Arg1` detection.

This analysis is carried out on the test set in order to characterize parser errors on different test set partitions.

This analysis also implies that the performance of PREV feature is affected in the effect of inclusion of the automatic label of `Arg2`, whereas "PREV" feature performs better in combination with gold labeled `Arg2` label features.

In summary, we improved the system performance using only one simple feature in case of `Arg1` is positioned in the previous sentence and the current sentence. The only simple feature "Prev" is not enough, since we are able to use the information that spans over two sentences only, not using at all the information inside the remaining 3 sentences of the 5-sentence discourse context for each discourse relation.

		P	R	F1
Automatic Intra-Sentential	Exact	0.667	0.497	0.570
	Partial	0.810	0.636	0.712
	Overlap	0.889	0.663	0.760
Semi-Automatic Intra-Sentential	Exact	0.657	0.467	0.546
	Partial	0.822	0.610	0.700
	Overlap	0.900	0.638	0.747
Automatic Inter-Sentential	Exact	0.521	0.176	0.263
	Partial	0.664	0.246	0.359
	Overlap	0.748	0.252	0.377
Semi-Automatic Inter-Sentential	Exact	0.526	0.206	0.296
	Partial	0.659	0.282	0.395
	Overlap	0.741	0.291	0.418

Table 21: Results of Automatic & Semi-Automatic Arg1 parsing for intra- and inter- sentential partitions.

3.7 Feature Selection with Automatic Data

We use the same feature set used by the parser in gold-standard settings in Table 31 (Chapter 2). This set includes a small set of lexical, syntactic, semantic, and discourse structural (i.e. Arg2 labels and the previous sentence feature) features, which convey the essential information needed to represent the arguments position and the clausal boundaries, as well as the internal clause structure.

We also carry out the same step-wise feature selection procedure (i.e. forward hill-climbing, discussed in Chapter 2) in order to identify only the feature combination that performs best in our parsing task.

In order to understand better the contribution of each feature and also to avoid sub-optimal solutions, we also run an ablation test by leaving out one feature in turn from the best-performing set.

We use the development split to generate results for the feature analysis to find the best performing feature set, whereas the train split is used to built model. The final result in Table 24 is generated using the test split only.

The results of our feature analysis are reported in Table 22 for Arg2 and Table 23 for Arg1. We do not report the scores having zero as F1-measure.

Both the feature-in-isolation procedure and the ablation test show that the connective sense feature is the most relevant feature for Arg1 and Arg2 as we observed in the feature analysis in gold-standard setting (cf. Chapter 2).

We also consider lemma and inflection together as a single feature, i.e. Morph as we did in Chapter 2.

Features	*P*	*R*	*F*1
Features in Isolation			
Token (T)	0.204	0.093	0.128
Connective (CONN)	0.444	0.237	0.462
IOB_Chain (IOB)	0.205	0.062	0.095
PoS	0.262	0.013	0.024
Morph(L+INFL)	0.185	0.040	0.066
Hill-Climbing Feature Analysis			
T+CONN	0.755	0.536	0.627
T+CONN+IOB	0.775	0.554	0.646
T+CONN+IOB+Morph	**0.784**	**0.563**	**0.655**
T+CONN+IOB+Morph+Prev	0.783	0.562	0.654
T+CONN+IOB+Morph+Prev+PoS	0.782	0.559	0.652
Token+CONN+IOB+PoS +Morph+BMV+Prev	0.781	0.559	0.652
Token+CONN+IOB+PoS +Morph+MV+BMV+Prev	0.779	0.557	0.649
Feature Ablation			
T+CONN+IOB	0.775	0.485	0.597
T+CONN+Morph	0.758	0.466	0.577
IOB+CONN+Morph	0.778	0.488	0.600
T+IOB+Morph	0.244	0.132	0.166

Table 22: Results with Single and Combined Features for `Arg2`

We notice that the best performing set for **`Arg1`** includes seven features (considering Morph as one feature), whereas the best feature combination for **`Arg2`** classification is achieved using only four features, namely token, IOB chain, connective sense and Morph, the same features as is in the case of gold-standard setting. The best performing feature set for **`Arg1`** is different than that of gold-standard settings. The feature "Boolean of Main Verb" (BMV) is absent. PoS is inside the optimized with a little improvement in Hill climbing steps. The hill climbing strategy shows that connective is the most important feature, though the effect of IOB chain is reduced as was in case of gold-standard settings. The **Arg2** labels and "Prev" features are found as strong features for the classification of **Arg1**. The *Morph* and PoS feature effect to the analysis with slight improvements in performance of **Arg1** classification.

Features	*P*	*R*	*F1*
Features in Isolation			
Token (T)	0.157	0.059	0.086
Connective (CONN)	0.186	0.076	0.108
IOB_Chain (IOB)	0.134	0.029	0.047
Morph(L+INFL)	0.168	0.014	0.026
Prev_feat(PREV)	0.520	0.054	0.098
Hill-Climbing Feature Analysis			
T+CONN	0.616	0.179	0.277
T+CONN+Arg2	0.575	0.214	0.355
T+CONN+Arg2+Prev	0.651	0.275	0.386
T+CONN+Arg2+Prev+IOB	0.660	0.326	0.437
T+CONN+Arg2+Prev+IOB+Morph	0.637	0.383	0.478
T+CONN+Arg2+Prev +IOB+Morph+PoS	**0.674**	**0.372**	**0.479**
T+CONN+Arg2+Prev +IOB+Morph+PoS+BMV	0.630	0.371	0.467
T+CONN+Arg2+Prev +IOB+Morph+PoS+BMV+MV	0.659	0.334	0.443
Feature Ablation			
T+CONN+Arg2+Prev+IOB+Morph	0.637	0.383	0.478
T+CONN+Arg2+Prev+IOB+PoS	0.634	0.325	0.430
T+CONN+Arg2+Prev+Morph+PoS	0.661	0.310	0.422
T+CONN+Arg2+IOB+Morph+PoS	0.666	0.304	0.417
T+CONN+Prev+IOB+Morph+PoS	0.627	0.257	0.364
T+Arg2+Prev+IOB+Morph+PoS	0.466	0.209	0.288
CONN+Arg2+Prev+IOB+Morph+PoS	0.629	0.371	0.466

Table 23: Results with Single and Combined Features for Arg1

The optimized feature set for **Arg2** is the same as in the gold-standard and the automatic settings. This is may be for the reason that **Arg2** is always syntactically bound to the connective. But in the case of the **Arg1** structural knowledge is equally important with the lexico-syntactic knowledge, especially when the SPTs are not gold-labeled.

3.7.1 Result with Optimized Feature Set

It is important to note that with the help of feature analysis the results in fully automatic settings increased insignificantly in case of the **Arg2**. We compare the results with optimized feature set to the results in the Table 16 . In case of **Arg1** the exact match score has fallen down, though very insignificantly; but the partial match remains the same, and a significant improvement for the overlap match scores.

		P	R	F1
Arg2	**Exact**	**0.755**	**0.598**	**0.667**
	Partial	0.905	0.699	0.789
Automatic	Overlap	0.942	0.745	0.832
Arg1	**Exact**	**0.658**	**0.344**	**0.452**
	Partial	0.790	0.441	0.566
Automatic	Overlap	0.915	0.444	0.598

Table 24: Results of **Arg2** and **Arg1** classification with automatic parse trees and connectives using selected feature-set via hill-climbing & ablation.

3.8 Conclusion

We implemented and evaluated an end-to-end discourse parser built by means of a cascading pipeline. In order to ensure the replicability of our approach, we used publicly available standard tools in the pipeline to acquire automatic connective sense labels and syntactic trees. We also accounted for error propagation in the pipeline by comparing the results obtained using gold standard data (from PTB and PDTB) with those obtained using automatic data. We trained a CRF classifier with a selected lexical, syntactic and semantic feature-set, so that arguments are labeled as token-sequences. In order to evaluate the impact of full syntactic information on the model, we compare also a system version using a deep-syntactic feature (IOB chain) with one lightweight configuration based on shallow syntactic features. Besides, we took further efforts to the improvement of the lightweight version adding more effective features. We performed greedy hill climbing feature selection in fully automatic settings. The result with optimized feature set could not improve the performance with the feature set in gold-label settings.

4 Parser Optimization with Global Features

4.1 Introduction

The automatic analysis of the discourse structure of a text is a complex task with a wide range of potential applications. The release of the Penn Discourse Treebank (Prasad et al 2008) [2] has resulted in a recent flurry of work in discourse parsing. In particular, there is a growing body of literature describing systems that extract arguments of explicit discourse connectives (Wellner et al 2007; Elwell et al 2008; Prasad et al 2010;Ghosh et al 2011a; Ghosh et al 2011b) [67, 69, 63, 96, 106].

We previously presented a method for automatic argument extraction based on chunking with conditional random fields (Ghosh et al 2011b) [106]. In contrast to previous approaches to argument extraction, our chunking system is very loosely coupled with the syntactic representation: It is completely straightforward to use one or more constituent, dependency, or shallow parsers in any combination since the argument boundaries are not tied to any particular constituent span. Other advantages include the simplicity of implementation by using standard chunking tools. The runtime of the system is also very low, with most of the processing time spent on feature extraction (i.e. running syntactic parsers).

However, while the chunking-based approach has the advantage of flexibility and speed, it is unable to take the global argument structural constraints into account. In particular, the PDTB annotation guidelines specify that exactly one `Arg1` and one `Arg2` must be annotated for every connective, while we often noticed that our system predicted no arguments. This causes our recall values to be low compared to the precision.

Now, we show that adding these constraints to the inference step improves the performance of the discourse parser. In particular, we see strong recall improvements. Global inference methods, including constraint-based as well as learning-based methods (often implemented as rerankers), have seen much use in NLP recently. Inference with constraints in particular has been successful in improving tasks such as semantic role labeling (Punyakanok et al 2008) [68]. This approach may be seen as a simple way to introduce long-distance structural relationships while still keeping the machine learning models simple.

There are relevant studies on the impact of global and local features on the models for natural language understanding. In this work, we address a similar problem in the context of discourse parsing. Although a good number of the papers in this area heavily rely on local classifiers (Grosz et al 1995; Soricut et al 2003; Lapata 2003; Barzilay et al 2005) [107, 26, 108, 109], there are still some important works using global and local informations together to form a model of discourse (Grosz et al 1992; Barzilay et al 2004; Soricut et al 2006) [110, 111, 112].

One of the main issues is the basis of the choice between a global or local or a joint model for discourse parsing: it all depends on the criteria to be able to capture maximum amount of information inside the discourse model. The policy for discourse segmentation plays a big role to formulate the maximizing criteria [110]. We study in the literature that defining a discourse segment is mostly a data-driven process: some argue for prosodic units, some for intentional structure and some for clause-like structures. We work with PDTB 2.0 annotation framework, therefore use a clause-like structure. Soricut et al (2003) [26] empirically showed that at the sentence level, there is a strong correlation between syntax and discourse, [96] found the same. Since the discourse structure may span over multiple sentences, inter-sentential features are needed to improve the performance of a discourse parser.

Linguistic theory suggests that a core argument frame (i.e. a pair of the **Arg1** and the **Arg2** connected with one and only one connective) is a joint structure, with strong dependencies between arguments (Toutanova et al 2008) [113]. Following this, Ghosh et al (2011b) [106] also injected some structure-level information through the token-level features, for eg. the previous sentence feature. Still there is a room for improvement with more structure-level information to that discourse model; though it is cost-intensive to modify this discourse model. Therefore in this work we re-use the model [106] and optimize the current loss function adding the global features through re-ranking of the single-best model.

Reranking has been a popular technique applied in a variety of comparable NLP problems including parsing (Collins 2000; Charniak et al 2005) [114, 88], semantic role labeling (Toutanova et al 2008) [113], NP Bracketing (Daume III et al 2004) [115], Named Entity Recognition (Collins 2002a) [116], opinion expression detection (Johansson et al 2010) [117], spoken language understanding (Dinarelli et al 2009) [118], now we employ this technique in the area of discourse parsing.

4.2 Hand-Crafted Postprocessing with Global Constraints

4.2.1 Implementation

Our system for the automatic extraction of discourse arguments for explicit connectives (Ghosh et al 2011b) [106] (cf. Chapter 2) consists of a pipeline, illustrated in Figure 6. We already described it fully in Chapter 2, again here we re-state some parts of it in order to be coherent with the next discussions.

Firstly, we assume that the explicit discourse connectives (and their high-level senses) are given to the system as input. They can be taken from the gold standard or automatically identified and disambiguated (Pitler & Nenkova 2009) [23], and for simplicity we used gold-standard connectives in this work. We then apply a module to extract the `Arg2` arguments, which are the easiest to identify since they are syntactically connected to the discourse connectives. After the `Arg2`s have been identified, we finally apply the `Arg1` extractor.

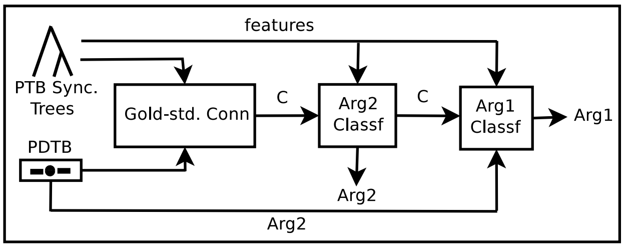

Figure 6: Pipeline for argument detection given a connective.

`Arg2` and `Arg1` extractors are implemented as conditional random field sequence labelers, which use a set of syntactic and structural features (see Chapter 2 Section 2.5 for a full discussion; also a re-statement later in this chapter at the Section 4.3.2). In order to reduce the processing time, we apply the sequence labelers to the sentence containing the connective, and a context window of up to two sentences before and after.

4.2.1.1 Adding Constraints

In our evaluations (cf. Chapter 2; Section 2.6.1), recall was always lower than precision. We noticed that the system often failed to predict any argument at all. This was especially true for `Arg1`s, which are not always syntactically connected to the connective and thus typically more distant than the `Arg2`s. However, since the PDTB annotation guidelines specify that exactly one

Arg1 and one Arg2 must be annotated for every connective, we may force the system to output arguments of each type. To improve the recall, we therefore implemented a weighted constraint-based postprocessor to make the system produce output satisfying the requirements defined by the annotation guidelines.

In order to find the best solution with a minimum of constraint violations, we generated the top k analyses output by the CRF for every sentence; these analyses can then be combined to form the k top analyses for the whole 5-sentence window around the connective. This combination is most efficiently carried out using a priority queue similar to a chart cell in the k-best parsing algorithm by (Huang & Chiang, 2005) [119].

The algorithm proceeds then through the k-best list and outputs an argument segmentation with the minimal number of constraint violations. If there are more than one such segmentation, we select the one with the highest probability. We note that the search for the optimum could as well have been implemented directly in the CRF inference as a modified Viterbi procedure (Manning & Schütze, 1999) [100], with a slightly more complex dynamic programming table. We leave the implementation of this algorithm to future work.

We counted the following five conditions as constraint violations:

Overgeneration. This constraint is violated if an Arg1 or Arg2 is split over multiple sentences. However, due to the fact that an argument may be split into several pieces (because of attribution spans, nonprojective syntactic constructions, or embedded connectives), we allow an argument to be split into more than one part in the same sentence.

Undergeneration. Since every connective must have arguments of each type, this constraint is violated if an argument is missing.

*Intersentential **Arg2**.* We count every Arg2 outside the sentence containing the connective as a violation, since they are required to be syntactically connected to the connective.

***Arg1** after the connective sentence.* We count every Arg1 after the sentence containing the connective as a violation.

Argument overlapping with the connective. Arguments are not allowed to overlap with the connective, since PDTB uses discontinuous argument spans to encode situations where a connective is embedded in an argument span.

4.2.1.2 Soft Constraints

In addition, we investigated an implementation based on *soft* constraints. For a hypothesis h with a set of violated constraints $V(h)$, we define a scoring function $f(h)$ based on the score assigned by the base CRF and a set of *constraint weights*, with one weight w_C for every violated constraint C. Our system then selects the hypothesis h that maximizes $f(h)$.

$$f(h) = \log P_{\text{CRF}}(h) - \sum_{C \in V(h)} w_C$$

Based on tuning on a development set, we set all the constraint weights to 1, except the weight for *Undergeneration* which was set to 2.

4.2.2 Analysis

We first report the argument extraction performance for the constraint-based postprocessors and compare it to the baseline CRF, and then analyze various aspects of the performance.

4.2.2.1 Performance Measurements

Table 25 shows the performance of the baseline system (cf. the gold standard system Chapter 2) [106]. As in that paper, we show precision and recall values using three evaluation protocols: *exact*, where an argument must have exactly the same boundaries to be counted as correct; *overlap*, where an argument is counted as correct if it overlaps with a gold standard argument; and *partial*, where a weight between 0 and 1 is used to measure the extent to which a segment corresponds to the gold standard [117]. As previously noted, the recall values are fairly low compared to the precision values.

Table 26 shows the effect of the postprocessing with hard constraints, using a k of 8. We note that recall is improved in all settings, in particular for `Arg1`. The increased recall is offset by lower values of precision. However, F1-measure always improves, especially for the partial and overlap measures.

Table 27 shows the corresponding table for the postprocessor using soft constraints, again with a k of 8. This postprocessor strikes a middle ground between the precision-oriented baseline system and the postprocessor with hard constraints, which is very recall-oriented. We also note that this system

		P	R	F1
Arg2	Exact	83.4	75.1	79.1
	Partial	93.4	84.2	88.6
	Overlap	97.2	87.5	92.1
Arg1	Exact	69.9	48.5	57.3
	Partial	82.9	61.7	70.7
	Overlap	91.0	63.1	74.6

Table 25: Performance of the baseline discourse parser.

		P	R	F1
Arg2	Exact	80.8	77.9	79.3
	Partial	92.8	89.0	90.9
	Overlap	96.9	93.4	95.1
Arg1	Exact	58.9	57.8	58.4
	Partial	73.6	75.7	74.6
	Overlap	80.5	79.0	79.7

Table 26: Results with constraint-based postprocessing.

scores achieves the highest exact F1-measure, while the other postprocessor has higher values for partial and overlap F1-measures.

4.2.2.2 Intersentential Arguments

The most challenging arguments to extract are the *inter-sentential* **Arg1**. Table 28 shows the performance of the three systems on these arguments. For these arguments, the postprocessor with hard constraints stands out from the other two: it is much more recall-oriented, while the other two have fairly similar performances. However, the constraint-based systems always outperform the baseline for all types of F1-measure.

Because of our window-based pruning strategy, the constraints naturally lead to a certain amount of overgeneration: in about 6% of the cases, the gold-standard **Arg1** is located outside the 5-sentence window, while the constraints still force the system to predict an **Arg1** inside the window. This lowers the upper bound on the precision that our system can possible achieve.

		P	R	F1
Arg2	Exact	81.8	77.1	79.4
	Partial	93.0	87.6	90.2
	Overlap	97.1	91.5	94.2
Arg1	Exact	66.8	53.1	59.2
	Partial	80.6	68.0	73.7
	Overlap	88.3	70.1	78.1

Table 27: Results with postprocessing using soft constraints.

		P	R	F1
Baseline	Exact	52.9	27.5	36.2
	Partial	68.6	40.2	50.7
	Overlap	78.8	41.0	53.9
Postprocessing (hard)	Exact	39.1	37.8	38.5
	Partial	55.9	56.4	56.1
	Overlap	62.4	60.3	61.4
Postprocessing (soft)	Exact	49.2	29.8	37.1
	Partial	65.9	44.1	52.7
	Overlap	75.0	45.5	56.6

Table 28: Intersentential Arg1 extraction results.

4.2.2.3 The Effect of the Number of Hypotheses

In any method based on generation of multiple hypotheses from an underlying base system, it is important to investigate the question of how many hypotheses are needed to reach the best achievable performance, since generating a large set of hypotheses may be inefficient. Table 29 shows the effect of the k value on the overlap F-measure for the task of `Arg1` extraction, along with the oracle F1-measure for the same task.

k	1	2	4	8	16
F1	74.6	79.1	79.4	79.7	79.7
Oracle F1	74.6	84.5	88.8	92.6	94.8

Table 29: Arg1 overlap F-measure for different values of k.

As is typical for these approaches, the largest gain is achieved immediately, when going from one to two hypotheses. However, in contrast to approaches based on reranking (see e.g. Johansson & Moschitti (2010) [117]), our performance reaches a plateau very quickly when increasing the hypothesis set size. This can be explained by the fact that our method immediately returns when finding a hypothesis without constraint violations. Table 30 shows the distribution of the positions of the first violation-free hypothesis. We note that a violation-free hypothesis was available among the four top-scored hypothesis in 97% of the cases.

1	2	3	4	5	6	7	8	>8
1,088	370	55	35	15	10	5	3	10

Table 30: Distribution of the position in the k-best list of the first hypothesis without constraint violations.

4.3 Automated System using Global Constraints

4.3.1 Backgrounds & Motivation

Currently we are using the single-best discourse parser that is extensively described in Chapter 2 of this thesis in gold-standard settings [106]. We are re-stating some part of that in order to be coherent with the current problems and discussions.

We know that this discourse parser can automatically extract of discourse arguments using a pipeline, illustrated in Figure 7. First, we input the explicit discourse connectives (with senses) to the system. These can be the gold labeled or automatically identified using the tool by Pitler & Nenkova (2009) [23]; for simplicity here we use Penn Discourse TreeBank (PDTB 2.0) gold-standard connectives (cf. See PDTB 2.0 Annotation manual [120]). Then a cascaded module is applied extracting the **Arg2** arguments, then the **Arg1**s are extracted.

The **Arg2** and **Arg1** extractors are implemented as conditional random field sequence labelers, which use a set of syntactic and structural features (see Chapter 2 Section 2.5 for a full discussion; also a re-statement at the following Section 4.3.2). In order to reduce the complexities, the sentence containing the connective, and a context window of up to two sentences before and after are supplied to the sequence labelers.

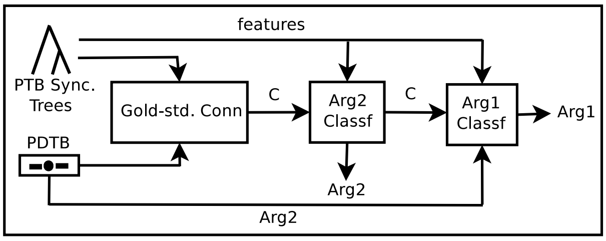

Figure 7: Pipeline for argument detection given a connective.

We present a passage of 6 sentences from a nutrition journal article parsed with that parser [17].:

<Conn id=1,sense=Comparison> ***Although***</Conn id=1>
<ARG2 id=1> **the mechanism of obesity development is not fully**
understood, it is confirmed<ARG1 id=2>**that obesity occurs**
</ARG1 id=2> <Conn id=2, sense=Temporal>***when***</Conn id=2>
<ARG2 id=2>**energy intake exceeds energy expenditure**</ARG2 id=2></ARG2 id=1>.

There are multiple etiologies for this imbalance, hence,
<Conn id=3, sense=Expansion> ***and*** </Conn id=3>
<ARG2 id=3>**the rising prevalence of obesity cannot be**
addressed by a single etiology</ARG2 id=3>.

<ARG1 id=4>
Genetic factors influence the susceptibility of a given child to an
obesity-conducive environment</ARG1 id=4>.
<Conn id=4, sense=Comparison> ***However*** </Conn id=4>,
<ARG2 id=4>**environmental factors, lifestyle preferences,**
and cultural environment seem to play major roles in the rising
prevalence of obesity worldwide</ARG2 id=4>.

In a small number of cases, childhood obesity is due to genes
such as leptin deficiency or medical causes such as hypothyroidism
and growth hormone deficiency or side effects due to drugs
(e.g. - steroids).

[17] We used best model with gold-standard data, for reference see Chapter 2, and Stanford lexicalized parser (Klein & Manning 2003) [97] to parse the text also used AddDiscourse tool [23], Morpha by Minnen et al [75] and RootExtract to parse the connective and the senses; time taken to parse: 17 sec.

Most of the time, <Conn id=5, sense=Comparison>
however </Conn id=5>, <ARG2 id=5>
personal lifestyle choices and cultural environment
significantly influence obesity</ARG2 id=5>.

In the evaluations of [106], it states that recall was much lower than precision for both the arguments, especially in case of Arg1. The system often failed to predict Arg1. It is harder to identify since it is not always syntactically bound to the connective, like Arg2, moreover it is typically more distant than the Arg2s.

We notice the same in the parser output. The parser found all five Arg2s for all five connectives, though there may be disagreement on the selected boundaries; the number of parsed Arg1s is only two, whereas the second one with id of 4 is a previous sentence argument.

To improve the recall, [121] implemented a weighted constraint-based *handcrafted* postprocessor to force the gold-standard single-best (for discussions see Chapter 2) [106] system to output arguments of each type abiding the requirements defined by the PDTB annotation guidelines.

In order to find the best solution with a minimum of constraint violations, the top k analyses output are generated by the CRF for every sentence; these analyses can then be combined to form the k top analyses for the whole 5-sentence window around the connective. This combination is most efficiently carried out using a priority queue similar to a chart cell in the k-best parsing algorithm by [119]. (see [121] for details)

4.3.2 Restatement: Features of Single-Best (Baseline) Model

We summarize the feature set of the base system (cf. see Chapter 2 for full discussion) [106] to emphasize the distinction between the local and global feature set for this work.

The token-level (local) feature set in the Table 31 can be divided into four categories:

1. Syntactic. $\{F3, F4, F6\}$ [18]

2. Semantic. $\{F2\}$

[18] Infection can be defined as morpho-syntactic feature.

Features used for **Arg1** and **Arg2** segmentation and labeling.	
F1.	Token (T)
F2.	Sense of Connective (CONN)
F3.	IOB chain (IOB)
F4.	PoS tag
F5.	Lemma (L)
F6.	Inflection (INFL)
F7.	Main verb of main clause (MV)
F8.	Boolean feature for MV (BMV)
F9.	Previous sentence feature (PREV)
Additional feature used only for **Arg1**	
F10.	**Arg2** Labels

Table 31: Feature sets for **Arg1** and **Arg2** segmentation and labeling in base system (Ghosh et al 2011a).

3. Lexical $\{F5, F7, F8\}$

4. Structure related token-level features. $\{F9, F10\}$

The remaining one (F1) is the token itself. The sense of the connective feature (F2) extracted from PDTB for the base system, though for the fully automatic one [96] it needs the PTB-style syntactic parse trees as input [23]. The features of category 1 and 3 are extracted either directly from the syntactic parse trees (F3, F4, F7) or they use indirectly the parse trees to extract the feature (F5, F6, F8).

The structure related token-level features do not use any parse tree. The **Arg2** label (F10) features are generated from the word sequence index in PDTB for the base system (for automatic system it is generated by the pipeline [96]); this feature is used to classify **Arg1** . The previous sentence feature "Prev" (F9) is a connective-surface feature and is used to capture if the following sentence begins with a connective. This is meant for the classification of the **Arg1** that resides in the previous sentence of the connective. The feature value for each candidate token of a sentence corresponds to the connective token that appears at the beginning of the following sentence, if any. Otherwise, it is equal to 0.

Although both of the structure-related features are strong features according to the feature analysis in [106], the base system is not able to capture all available global features inside the 5-sentence discourse context, merely uses

2-sentence context. This is due to the fact that CRF classifier uses a narrow window, that can only capture the information nearby the token under consideration. Therefore it becomes impossible to inject more information about the 5-sentence discourse window structure.

4.3.3 Feature Set

We use a global feature set, a part of this is already described in the Section 4.2.1.1 this Chapter, we are re-stating those parts for the sake of coherence of the discussion. The global features are defined as the data-driven, hand-crafted rule generated and non-grammatical (i.e. no syntactic parse tree is used to generate this features) features. The global features are computed using each list of k-best lists, in contrast to the lexico-syntactically generated local features for each token item for each sentence of n-best lists. The usage of global features is meant for exploring the yet undiscovered dimension of the each 5-sentence discourse window. Global feature set consists of the eight features that works on a full 5-sentence discourse window (cf. Section 4.3.1). The first six (i.e. GF0-GF5) of these are same with the constrained system (cf. Section 4.3.1).

None of the features are extracted from a syntactic parse tree. All the seven features (GF1-GF7) are derived from the generated `Arg` tags of the n-best lists, the first one is the logarithm of posterior probability computed from the CRF posterior probability output for each list of the n-best lists. The finer description of each feature is given below.

GF0. *logarithm of Posterior Probability.* this feature is generated by the base CRF classifier. The CRF generates the probability per sentence, for each list of the n-best lists. We calculate the sum of the logarithm of each probability during generation of k-best lists forming 5-sentence discourse window.

GF1. *Overgeneration.* It is possible for an argument to be split into more than one part in the same sentence. We found these cases several times in PDTB. This constraint is violated if an `Arg1` or `Arg2` is split over multiple sentences. This is a predominant problem for those lists of the n-best lists those are generated with low posteriors. This feature exhibits the problem of overgeneration to the reranker with the counts.

GF2. *Undergeneration.* According to PDTB annotation scheme, every connective must have arguments of each type. This constraint is violated if an argument is missing. This is the prevalent problem in the single-best system, especially for the Arg1 classification.

GF3. *Intersentential Arg2* (used only for Arg2 reranker). This is the count of Arg2, if any, occurs classified outside connective sentence - this way the system is constrained to have any inter-sentential Arg2. This is a *hypothetically motivated* feature to reduce the complexity of the classification problem – in fact in PDTB 2.0, there are a few cases of Arg2 of explicit connective (i.e. the 114 out of 18459), where it spans beyond the connectives sentence to include additional sentences in the subsequent discourse (Prasad et al 2008) [2].

GF4. *Arg1 after the connective sentence.* It is the count of Arg1, if any, occurs classified after connective sentence. Through this feature we attempt to constrain the system to have Arg1s always occurring in the previous sentence or before the previous sentence of the one in which the connective occurs.

GF5. *Argument overlapping with the connective.* It is the count of the cases if there is a token overlap between Args and connective tokens. This is also not possible for the PDTB-style annotation, so we intend to constrain the overlapping, if any.

GF6. *Argument begins with -I tag.* It is the count of the cases if the generated Arg chunks begins with the -I (inside) tag, violating the principle of IOB tags for chunking. This is only possible if the CRF chunker fails to tag the boundaries properly.

GF7. *Argument begins with -E tag.* It is the count of the cases if the generated Arg chunks begins with the -E (end) tag instead of a -B(begin) tag. This is also possible if only the CRF chunker fails to tag the chunk boundaries properly.

We attempt to categorize this feature set according to the properties they bear: $\{GF0\}$ is the *intrinsic* global feature - it is the evidence of confidence on decisions made by the single-best model; $\{GF1, GF2\}$ check the *prevalent problems* seen through the evaluation of decisions by the single best model; $\{GF3, GF4, GF5\}$ are the *hypothetical* global features those reduce classification complexities, and they are inspired by the general trends or rules for annotation in PDTB. $\{G6, G7\}$ check the *mistakes* in IOB tagging by the CRF chunker.

4.3.4 Reranking Approaches

We formalize the reranking algorithm as follows: for a given sentence s, a reranker selects the best parse $\hat{y}$ among the set of candidates: $\mathrm{C}(s)$ according to some scoring function:

$$\hat{y} = argmax_{y \in \mathrm{C}(s)} \mathrm{score}(y) \tag{7}$$

In n-best reranking, $\mathrm{C}(s)$ is simply a set of n-best parses from the baseline parser, that is, $\mathrm{C}(s) = \{y_1, y_2, ..., y_n\}$. (Huang 2008 [122])

In this work we followed two approaches for the reranking task:

1. Structured Learning Approach: in this case the reranker learns directly from a scoring function that is trained to maximize the performance of the reranking task (Collins & Duffy, 2002) [123]. We also investigate two popular and efficient online structured learning algorithms: the structured voted perceptron by (Collins & Duffy 2002) [123] and Passive-Aggressive(PA) algorithm by (Crammer et al 2006) [124]. The weight-vectors observed from the training phase are averaged following Freund & Schapire (1999) [125]. In case of structured perceptron for each of the candidate in a ranked list the scoring function of Equation 7 is computed as follows:

$$\mathrm{score}(y_i) = \mathbf{w} \cdot \mathbf{\Phi}(x_{i,j}) \tag{8}$$

where $\mathbf{w}$ is the parameter weight-vector and Φ is the feature representing function of $x_{i,j}$; $x_{i,j}$ denotes the j-th token of the i-th sentence. Since the PA algorithm is based on the theory of large-margin, it attempts find a score that violates the margin maximally by adding an extra cost i.e. $\sqrt{\rho(x_{i,j})}$ to the basic score function for structured perceptron i.e. equation 8. Here ρ is computed as $1 - F(x_{i.j})$, F: F1-measure. The online PA also takes care of the learning rate of perceptron, which is considered as 1 in structured perceptron. The learning rate in online PA is min-value between a regularization constant and normalized score function value.

2. Best vs. rest Approach: in the preference kernel approach (Shen & Joshi, 2003) [126] the reranking problem is reduced to a binary classification task on pairs. This reduction enables even a standard support vector machine to optimize the problem. We use a component of this task. We define the best scored discourse window (Section 4.3.4.3) as a positive example and the rest are the negatives to the system. We use a standard support vector machine [127] with linear kernel.

3. Preference Kernel Approach: we also investigated the classical approach of preference kernel, as it is introduced by (Shen & Joshi, 2003) [126]. In this method, the reranking problem learning to select the correct candidate h^1 from a candidate set $\{h^1, \cdots, h^k\}$ is reduced to a binary classification problem by creating pairs: positive training instances $\langle h^1, h^2 \rangle, \cdots, \langle h^1, h^k \rangle$ and negative instances $\langle h^2, h^1 \rangle, \cdots, \langle h^k, h^1 \rangle$. The advantage of using this approach is that there are abundant tools for binary machine learning.

If we have a kernel K over the candidate space T, we can construct a *preference* kernel (Shen & Joshi, 2003) [126] P_K over the space of pairs $T \times T$ as follows:

$$\begin{aligned} P_K &= K(h_1^1, h_2^1) + K(h_1^2, h_2^2) \\ &- K(h_1^1, h_2^2) - K(h_1^2, h_2^1) \end{aligned} \tag{9}$$

In our case, we make pair from the n-best hypotheses h_i as $\langle h_i^1, h_i^2 \rangle$ generated by the base model. We used linear kernel to train the reranker.

Thus we create the feature vectors extracted from the candidate sequences using the features described in this Chapter, Section 4.3.3. We then trained linear SVMs using the LIBLINEAR software (Fan et al 2008) [128], using L1 loss and L2 regularization.

4.3.4.1 Notes on Support Vector Machine

Support Vector Machines(SVMs) are one of the binary classifiers based on maximum margin strategy introduced by Vapnik (1995) [127]. Suppose we are given l training examples $(\mathbf{x_i}, \mathbf{y_i})$, $(1 \leq i \leq l)$, where $\mathbf{x_i}$ is a feature vector in n-dimensional feature space, y_i is the class label $\{-1, +1\}$ (positive or negative) of $\mathbf{x_i}$. SVMs find a hyperplane $\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = \mathbf{0}$ which correctly separates training examples and has maximum margin which is the distance between two hyperplanes $\mathbf{w} \cdot \mathbf{x} + \mathbf{b} \geq \mathbf{1}$ and $\mathbf{w} \cdot \mathbf{x} + \mathbf{b} \leq \mathbf{1}$. The optimal hyperplane with maximum margin can be obtained by solving the following quadratic programming.

$$\begin{aligned} min\frac{1}{2}||w|| + C\Sigma_i^l \xi_i \\ \text{s.t.} y_i(w \cdot x_i + b) \geq 1\xi_i \\ \xi_i \geq 0 \end{aligned}$$

where C is the constant (in our experiment, we depend on our SVM reranker), ξ_i is called a slack variable for a non-separable case. Finally, the optimal hyperplane is written as follows.

$$f(\mathbf{x}) = \mathrm{sign}(\boldsymbol{\Sigma}^{\mathbf{l}}_{\mathbf{i}}\alpha_{\mathbf{i}}\mathbf{y_i}\mathbf{K}(\mathbf{x_i},\mathbf{x})+\mathbf{b})$$

where α_i is the Lagrange multiplier corresponding to each constraint, and $K(\mathbf{x}', \mathbf{x}'')$ is called a kernel function, it calculates similarity between two arguments $\mathbf{x}'$ and $\mathbf{x}''$. SVMs estimate the label of an unknown example $\mathbf{x}$ whether sign of $f(\mathbf{x})$ is positive or not. There are two advantages in using SVMs for statistical dependency analysis:

(i) High generalization performance in high dimensional feature spaces.

(ii) Learning with combination of multiple features is possible by virtue of polynomial kernel functions.

SVMs are discriminative classifiers, and not generative probabilistic models like naive Bayes classifiers or maximum entropy models. The absolute value of $f(x)$ in second equation represents the distance of example x from the optimal hyperplane, which is not an appropriate score (or cost) in use of dynamic programing for search of the best answer, though it is widely used in probabilistic models. Therefore we employ a reranking classification using SVMs following Shen & Joshi (2005) [129].

4.3.4.2 Notes on Perceptron Algorithm

We used single-layered perceptron throughout the task. Therefore in discussions we will concentrate on this type of perceptron only. We discuss the two types of perceptron that we used in our works: Online Passive-Aggressive and Structured (Voted) perceptron.

Voted Perceptron (Perceptron Algorithm by Collins[2002])

We also shortly discuss here about online passive-aggressive perceptron for multiclass problem. The online learning is an inductive learning process where the learning proceeds in a sequence of trials. Each trial can be decomposed into three steps. First the algorithm receives an instance. Second the algorithm predicts the label of the instance. Third the algorithm receives the true label of the instance (Littlestone 1988; Vovk et al 2005 [130, 131]). Crammer et al (2006) [124] explained the passive-aggressive algorithm in the

Algorithm 2 Voted Perceptron training [123]

```
Define Score: $F(x, w) = w \cdot h(x)$
Input: Example $x_{i,j}$ with feature vectors $h(x_{i,j})$
Initialization: Set parameter $w^0 = 0$
for $i = 1 \cdots n$ do
  ArgMaxing of Scores: $j = \arg\max_{j=1\cdots n_i} F(x_{ij}, w^{i-1})$
  if (j=1) then
    $w^i = w^{i-1}$
  else
    Updation: $w^i = w^{i-1} + h(x_{i1}) - h(x_{ij})$
  end if
end for
Output: Parameter vectors $w^i$ for $i = 1 \cdots n$
```

Algorithm 3 Voted Perceptron testing

```
Define: $F(x, w) = w \cdot h(x)$
Input: A set of candidates $x_j$ for $j = 1 \cdots m$,
A sequence of parameter vectors $w^i$ for $i = 1 \cdots n$
Initialization: Set $V[j] = 0$ for $j = 1 \cdots m$
for $i = 1 \cdots n$ do
  ArgMaxing of Scores: $j = \arg\max_{k=1\cdots m} F(x_k, w^i)$
  Voting: $V[j] = V[j] + 1$
end for
Output: $x_j$ where $j = \arg\max_k V[k]$
```

online learning scenario. In contrast to the classical or structured voted perceptron algorithm it introduces a concept of margin in classification. On rounds (i.e. no. of epochs for a classical perceptron) where the algorithm attains a margin less than 1 it suffers an instantaneous loss. This loss is defined by the following hinge-loss function l,

$$l(\mathbf{w};(\mathbf{x},y)) \;=\; \begin{cases} 0, & \text{if } y(\mathbf{w}\cdot\mathbf{x}) \geq 1 \\ 1 - y(\mathbf{w}\cdot\mathbf{x}), & \text{otherwise} \end{cases}$$

The algorithm is named as *passive-aggressive* (Crammer et al. 2006) [124]. The resulting algorithm is called *passive* whenever the hinge-loss is zero, that is, $\mathbf{w}_{t+1} = \mathbf{w}_t$ whenever $l_t = 0$. In contrast, on those rounds where the loss is positive, the algorithm *aggressively* forces $\mathbf{w}_{t+1}$ to satisfy the constraint $l(\mathbf{w}_{t+1};(\mathbf{x}_t, y_t)) = 0$ regardless of the step-size required. Here $\mathbf{w}_{t+1}$ is a weight vector at $t+1$-th round (epoch).

Algorithm 4 Prediction-Based(PB) Max-Loss(ML) Cost Sensitive Multiclass Online PA [124]

```
**Input:** cost function $\boldsymbol{\rho}(y, y')$
**Initialize:** $\mathbf{w}_1 = (0, \cdots, 0)$
**for** $t = 1, 2, \cdots$ **do**
    receive instance: $\mathbf{x}_t \in \mathbb{R}^n$
    predict: $\hat{y}_t = \arg\max_{y \in \mathcal{Y}}(\mathbf{w}_t \cdot \mathbf{\Phi}(\mathbf{x}_t, y))$
    receive correct label: $y_t \in \mathcal{Y}$
    define: $\tilde{y}_t = \arg\max_{r \in \mathcal{Y}} \mathbf{w}_t \cdot \mathbf{\Phi}(\mathbf{x}_t, r) - \mathbf{w}_t \cdot \mathbf{\Phi}(\mathbf{x}_t, y_t) + \sqrt{\boldsymbol{\rho}(y_t, r)}$
    define:
        $q_t = \begin{cases} \hat{y}_t, & \text{PB} \\ \tilde{y}_t, & \text{ML} \end{cases}$
    suffer loss: $l_t = \mathbf{w}_t \cdot \mathbf{\Phi}(\mathbf{x}_t, q_t) - \mathbf{w}_t \cdot \mathbf{\Phi}(\mathbf{x}_t, y_t) + \sqrt{\boldsymbol{\rho}(y_t, q_t)}$
    set: $\boldsymbol{\tau}_t = \frac{l_t}{\|\mathbf{\Phi}(\mathbf{x}_t, y_t) - \mathbf{\Phi}(\mathbf{x}_t, q_t)\|^2}$
    update: $\mathbf{w}_{t+1} = \mathbf{w}_t + \boldsymbol{\tau}_t(\mathbf{\Phi}(\mathbf{x}_t, y_t) - \mathbf{\Phi}(\mathbf{x}_t, q_t))$
**end for**
```

We assume for the entire task that our data is linearly separable. Now we compare in short the SVM and perceptron classifier:

Comparison between SVM & Perceptron.

One of the main contributions of classical SVM learning theory is a proof of a new bound on the difference of between the training and test error of a linear classifier that maximizes the margin. The significance of this bound is that it depends only on the size of margin or on the number of the support vectors, not on the dimension. The perceptron classifier learns from each of the given samples, i.e. it learns from the whole distribution of the given training points. Therefore the learning function it draws from the training is more noisy than that of SVM.

The second contribution of SVM learning theory is that it provides a method for computing the maximal margin classifier efficiently for some high dimensional mappings (this idea is based on kernel functions); though the linear kernel learns fast and the computation cost of using other complex kernel function like polynomial kernel is very high. In case of perceptron it is very easy to use some other kind of learning strategy like voting or online passive-aggressive learning; with these learning principles a perceptron gives a comparable performance with SVM, which has already been noted in the area of NLP.

The weakness of SVM is that it is not easy to implement and their learning process is slow; instead Perceptron algorithm is very easy to implement and is very fast to train. Especially, the voted perceptron is superior to SVM in terms of learning time, prediction time and memory footprint[19].

4.3.4.3 Experiments

We used the same gold-standard data and its splits as is described in Chapter 2 Section 2.7. We prepare the n-best outputs of sentences from the base system (cf. Section 4.3.1). The training data is prepared from the input of n-best lists of the train split, using a oracle module, which generates k-best oracle lists from the n-best single outputs. We procure k-best lists from oracle using the evaluator module (see section 2.6.1), ordered by the highest to the lowest probability score. Each of the list of the k-best list is a 5-sentence discourse window.

We prepare the test data given the n-best lists of the test split. We obtain k-best list for testing, prepared with the module described in section 4.3.1. We re-integrate the sentences connected with the same discourse connective

[19] Memory footprint refers to the amount of main memory that a program uses or references while running.

id into the 5-sentence discourse window keeping the connective-bearing sentence in the middle. This re-integration done using a priority queue in the style of [119]. Each of the list from the k-best list are ordered by the highest to the lowest score with sum of the log of posterior probabilities of each sentence in the n-best list.

Therefore, in short, the n-best list is the list of sentence-level analyses whereas the k-best list is the list of 5-sentence discourse window-level analyses.

We followed the same approach for evaluation that we illustrated in Chapter 2 under Section 2.6.1.

Baseline: we consider the performance of the single-best output from the base implementation (cf. Section 4.3.1) as the baseline.

4.3.4.4 Classifier Results

	ARG1 Results			ARG2 Results		
Exact	P	R	F	P	R	F
Baseline	69.88	48.51	57.26	83.44	75.14	79.07
Online PA	**66.10**	**53.92**	**59.39**(16)	82.59	76.39	79.37(4)
Struct Per	67.18	52.64	59.03(4)	**82.96**	**76.28**	**79.48**(8)
BestVsRest	66.19	52.83	58.94(8)	81.69	77.14	79.35(4)
Pref-Linear	66.54	53.31	59.20(4)	82.82	76.28	79.42(4)

Table 32: Exact Match Results for four classifiers. Baseline scores in the first row. Used n-best list numbers in parenthesis. The best performances are boldfaced.

We observe that reranking with global features improved the F1 scores for `Arg1` significantly, although for `Arg2` the improvement is insignificant. Since in most of the cases the `Arg2` is syntactically bound with the connective, it is obvious that lexico-syntactically motivated local features help the classification of `Arg2`. On the other hand, the classification of `Arg1` is considerably dependent on non-grammatical, hand-crafted rule generated features. If we compare our reranking classification results of `Arg1` with that without previous sentence feature shown in Table 6 and Table 7 under Chapter 2 [106], then we observe that the global-constrained and structurally-motivated features improved the classification of `Arg1` by more than 10 points.

We also notice from the table for both the argument classification cases that we achieve balanced scores in terms of precision and recall with the structured global features. In fact there is a good improvement of recall without much loss in terms of precision.There is not any significant improvement in case of Arg2 reranking because the problem of the classification mostly resides on boundary detection of Arg2; also, we know that estimation of position of Arg2 is pretty easy task, given the connective is correctly identified.

	ARG1 Results			ARG2 Results		
Exact	*P*	*R*	*F*	*P*	*R*	*F*
Baseline	82.90	61.65	70.72	93.40	84.20	88.56
Online PA	**80.11**	**69.43**	**74.39**(16)	**92.94**	**85.73**	**89.19**(4)
Struct Per	81.18	67.03	73.43(4)	93.20	85.50	89.17(8)
BestVsRest	81.25	66.46	73.11(8)	93.03	85.16	89.1(4)
Pref-linear	80.55	68.49	74.03(4)	93.12	85.56	89.18(4)

Table 33: Partial Match Results for four classifiers. Baseline scores in the first row. Used n-best list numbers in parenthesis. The best performances are boldfaced.

We note an improvement of Arg1 in Table 33, with softer partial evaluation metrics; we also observe the same trend in results for Arg2 classification as in the table 32.

4.3.4.5 Candidate Set Size

We conduct further experiments to study the influence of the candidate set size on the quality of reranked output. In addition, we also attempt to assess the upper-bound of reranker performance, i.e. the oracle performance. We choose the reranker based on online PA among the four classifiers. Since all the four classifiers performed comparably the same way, it is enough to study the performance of one of them on the candidate set size, that will reflect the performance of the other classifiers. We also describe and discuss the results on the exact partial measures only, as we notice from the previous section that the effect of reranking is comparable with the exact measure and softer measures.

In both the Tables (34, 35) we notice that the oracle performance is steadily increasing with 16-best lists. We observe that the performance of classification of both Arg1 and Arg2 increases at the level of 2-best list then it stagnates after 4-best performance. This nature of increment may be related to the simple but high-level feature set used in this task of the discourse parsing; and it can also be some issues involved with local feature set, as we

	Reranked ARG1			Oracle		
k	P	R	F	P	R	F
1	69.88	48.51	57.26	69.88	48.51	57.26
2	67.26	52.34	58.87	81.26	61.70	70.14
4	66.39	53.56	59.29	88.35	71.91	79.29
8	66.11	53.86	59.36	92.47	79.09	85.26
16	66.10	53.92	59.39	93.80	83.77	88.50

Table 34: Oracle and reranker performance as a function of the candidate set size of `Arg1`.

	Reranked ARG2			Oracle		
k	P	R	F	P	R	F
1	83.44	75.14	79.07	83.44	75.14	79.07
2	82.90	75.69	79.13	90.13	82.43	86.11
4	82.59	76.39	79.37	92.27	86.53	89.31
8	82.41	76.44	79.32	92.81	88.13	90.41
16	83.41	76.44	79.32	92.82	88.54	90.63

Table 35: Oracle and reranker performance as a function of the candidate set size of `Arg2`.

observed a huge difference of posterior probabilities between the single-best and the each of the $(n-1)$ lists of a n-best decision by CRF.

4.3.4.6 Reranked Intersentential ARG1

We also attempt to observe the effect on inter-sentential classification of **`Arg1`**, with the results obtained with online PA perceptron. As expected, the change we notice the effects in the Table 36 is a fraction of potential improvement. We compare the inter-sentential versus overall classification results of **`Arg1`** and we notice that the increment in inter-sentential **`Arg1`** classification considerably contributes to overall **`Arg1`** classification.

		P	R	F1
Baseline	Exact	52.87	27.80	36.44
	Partial	68.93	41.06	51.48
	Overlap	79.62	41.88	54.88
Best Reranked ARG1	Exact	50.41	30.04	37.56
	Partial	66.51	44.95	53.78
	Overlap	76.13	44.54	56.23

Table 36: Inter-sentential Reranked `Arg1` Results.

4.3.5 Impact of Features

We study the impact of global features on the performance on **`Arg1`** reranker with the development set (cf. Section 4.3.4.3). We are leaving behind the feature performance of the **`Arg2`**, as the improvement by the reranker for this case is not significant.

Table 37 shows the results of an analysis with incremental greedy-search based feature selection. All the performance steps are evaluated with a k of 16.

The first row of the Table 37 contains the *log posterior* only ($GF0$). This results to the best result achieved by best gold-labeled system, illustrated in Table 6, Chapter 2 [106]. Beside this, we also verified with the test set, that if we run the reranker with this feature only, then it results to the baseline performance.

Then the *undergeneration* feature ($GF2$) is chosen through greedy search among the other features. It gives us, jointly with the log posterior, a significant improvement over the baseline. The impact is predictable as $GF2$ addresses the basic problem that has driven us to the current task.

The addition of the *overgeneration* ($GF1$) feature also increased the performance, though non-significantly; this feature is important for the reranker because this is meant for fixing a predominant overgeneration problem in the n-best lists.

We observe that the $F1$ measure increases significantly after adding the next important feature *Arg1 after the connective sentence* ($GF4$); in this case the recall increases more in comparison to the increment in the precision.

In the next step, the feature *Argument overlapping with connective* ($GF5$) is added. This decreases the $F1$ score a bit, though it increases the precision lowering the recall.

We reach to the second-best performance of the `Arg1` reranker after adding the feature *Argument begins with -I tag* ($GF6$).

The addition of the feature: *Argument begins with -E tag* ($GF7$) does not improve the performance much. It is possible that there was no such mistake by CRF inside the test data.

The scores with partial and overlap matches show the same trend so we leave the discussion with them in order to avoid the redundancy.

Additionally, we also analyze the individual effect of each of features from the set ($GF1, GF2, GF4, GF5, GF6, GF7$), jointly with the feature $GF0$, though only the undergeneration feature increased the performance over the baseline.

In summary, all the features in this feature set are contributing towards an improvement of performance. The *intrinsic* $GF0$ is contributing to achieve the baseline performance; on the top of this feature the contribution by the undergeneration ($GF2$) feature is the most significant one. The remaining features also contribute some other ways: while the addition of some achieve more precision in results ($GF1, GF5, GF7$), the addition of the others ($GF4, GF6$) have a balancing effect between the recall and the precision.

4.3.6 Reranked Arg1 Results for Fully Automatic System

We also attempt to rerank the parser in fully automatic settings. We found the best result with 16-best list using Online PA classifier. We present the reranking results for `Arg1` in Table 38 with the baseline results in full automatic settings (for reference see Chapter 3 Table 16). We note that following

System	*P*	*R*	*F*1
GF0 (Posterior Only)	73.12	50.36	59.64
GF0+GF2	69.62	55.34	61.67
GF0+GF2+GF1	69.92	55.21	61.70
GF0+GF2+GF1+GF4	70.12	56.05	62.30
GF0+GF2+GF1+GF4+GF5	72.36	53.72	61.66
GF0+GF2+GF1+GF4+GF5+GF6	71.10	55.28	62.20
GF0++GF2+GF1+GF4+GF5+GF6+GF7	71.84	54.82	62.19

Table 37: Exact Match Results through Incremental Feature Selection.

this way the imbalance between precision and recall is satisfactorily minimized, compared to the baseline single-best system.

In both the cases, i.e. the partial, and especially for the overlap match scores, the performances are improved substantially, if compared to the single best baseline; the exact match score has not increased significantly. Perhaps this effect is due to the noise in data in automatic settings.

		P	R	F1
Baseline	Exact	65.54	34.92	45.56
	Partial	77.17	44.69	56.60
	Overlap	85.36	45.48	59.34
Reranked Auto ARG1	Exact	54.72	39.65	45.94
	Partial	66.86	52.02	58.52
	Overlap	74.04	53.65	62.17

Table 38: Best Reranked Full Automatic `Arg1` Results with 16-Best List.

4.4 Conclusion

We have presented a constraint-based hand-crafted method that improves a shallow discourse parser based on chunking with conditional random fields. The method converts a severely undergenerating output into one where precision and recall are balanced, and where the requirements imposed by the annotations guidelines are fulfilled. The recall improvements are particularly visible when we use evaluation protocols with reduced strictness in boundary checking.

The method we have presented here is simple to implement, but it would also be interesting to see how well it compares to other global approaches. In particular, it would be very straightforward to replace our weighted constraint system by a reranker trained using standard machine learning techniques. Even in that case, the constraint system could serve as a filter to reduce the hypothesis set size for the reranker. However, the development of useful features for a reranker is an open problem.

In our automated system with global constraints we note a significant improvement over the best performing model of discourse parser on the PDTB corpus. This is mostly contributed by the better performance in `Arg1` classification.

We also find that global features have greater impact on `Arg1` classification than that of `Arg2`. We investigate that the performance of `Arg1` improved by more than 10 points in terms of F1 measure using the global (see Section 4.3.3) and structure related features (cf. Section 4.3.1). This happens perhaps due to the fact `Arg2` is syntactically bound to the connective, whereas `Arg1` is not. `Arg2` depends more on local features (cf. Section 4.3.1) than global one.

The motivation of the work is to perform a balanced classification for both `Arg1` and `Arg2`, implementing the constrained-system with global features. This enables to increase a huge recall without losing much in terms of precision.

It is also observed that while the performances of oracle of `Arg1` and `Arg2` are increasing steadily, the performances of both the rerankers stagnate at or before the point of 16-best lists; this is perhaps due to our effective, simple and small feature set.

Finally, we compute the re-ranked results with full automatic settings for `Arg1` only; we do not observe any improvement for the re-ranked `Arg2` in full automatic settings, as in the case with gold-standard settings.

5 Conclusions

In this thesis-work, we

presented the main contribution of this thesis, that is the end-to-end discourse parser that takes raw texts as input and outputs explicit discourse connectives and their respective arguments with the boundaries.

presented a discourse parsing method. The method is based on cascaded structured prediction, for classifying the arguments of a discourse connective with minimal error propagation. First we established the method for gold labeled data then we implemented this method in full automatic settings.

compared the performance of (gold-labeled) cascaded model against non-cascaded structured prediction algorithms using standard sequence labeling, showing that it is competitive with existing techniques.

investigated the performance of cascaded model under various scenarios: with gold-labels, full automatic and two other mixed cases (cf. Chapter 3) in order to understand the source of errors in results. We also presented lightweight cases with reduced time-complexity in gold-labeled and full automatic settings; we also reranked the n-best decisions by the single-best model in order to achieve better performance.

devised useful shallow features for the lightweight cases.

developed a number of useful data-driven non-grammatical structural (global) features to optimize the parser performance, without any extra computational overhead.

5.1 Main Challanges

There are several challenges and limitations of this work. The most relevant ones are the following:

- *Ambiguities involving connectives:* We carry forward the ambiguities involving the connectives in PDTB (cf. discussion in Pitler et al 2008 [71]), which is inherited from the language itself. Our intention is to avoid that ambiguity as much as possible using the top-layer senses.

- *Parser-Coverage:* This parser gives coverage for the explicit relations only, whereas we know from published statistics (Pitler et al 2008 [71]) that there is an equal amount of *contingency* implicit relations in the texts we used.

- *Cross-Domain Parsing:* we can use the method of this thesis for cross domain parsing, though we did not compute the cross domain performance of current model that is trained with news domain data.

- *Discourse parsing for spoken languages:* we understand that discourse parsing for spoken language will be harder. We depend on syntactic parser output to generate discourse structure. This implies also that it will be difficult to implement our pipeline for those languages for which there is no such kind of resource.

6 Appendix

6.1 Introduction

Coherence refers to the *meaning* relation between textual units. A coherence relation explains how the meaning of different textual unit can combine to build a discourse meaning for a larger unit. There exists other kind of meaning units in parallel to discourse units, those form a kind of meaning structure inside language. There are other kind of semantic structure inside language viz. predicate-argument-structure.

The Berkeley FrameNet project is created to aim a documentation of the range of semantic and syntactic combinatory possibilities of each word in each of its senses, through computer-assisted annotation of example sentences and automatic tabulation and display of the annotation results. This combinatory possibilities are created on the basis of verb valency or valences. We know that valence refers to the number of arguments controlled by a verbal predicate. It is related, but not identically, to the verb transitivity, that counts only object arguments of the verbal predicate. On the other hand, valence includes all arguments including the subject of the verb.

That project created an on-line FrameNet lexical database for English, based on frame semantics, which is also supported by corpus evidence; starting from the sentences of the corpus it has started to take the initiative of "continuous-text-annotation" (see below for details), which resembles with a discourse meaning structure in a document. One simple example from that database is as follows:

$[_{Cook}$ Matilde] **fried** $[_{Food}$ the catfish] $[_{Heating-Instrument}$ in a heavy iron skillet].

where "fried" is the frame-evoking "lexical word" that contains the arguments with frame-elements like *Cook*, *Food* and *Heating instrument.* These lexical word also called as discourse roles [45].

We find an example that in GNOME corpus [132, 133] GF (the grammatical function of the NP) is an attribute of Named Entities – this is basically a property generally taken to play an important role in determining the salience of the discourse entity it realizes (Grosz 1995)[107]. The instructions for annotation of this attribute are derived from those used in the FRAMENET project [134]. So it is clear that it is possible to use successfully both the structures (viz. discourse and frame-semantic).

Currently we are interested in the problem of extracting meaning structures from spoken utterances in human communication. In Spoken Language Understanding (SLU) systems, parsing of meaning structures is carried over the word hypotheses generated by the Automatic Speech Recognizer (ASR). This approach suffers from high word error rates and ad-hoc conceptual representations. In contrast, in this work we aim at discovering meaning components from direct measurements of acoustic and non-verbal linguistic features. The meaning structures are taken from the frame semantics model proposed in FrameNet, a consistent and extendibles semantic structure resource covering a large set of domains. We give a quantitative analysis of meaning structures in terms of speech features across human–human dialogs from the manually annotated LUNA corpus. We show that the acoustic correlations between pitch, formant trajectories, intensity and harmonicity and meaning features are statistically significant over the whole corpus as well as relevant in classifying the target words evoked by a semantic frame.

6.2 The FrameNet Semantic Structures in Conversational Speech

We carried out our experiments using the LUNA spoken dialog corpus, which was developed in the context of the LUNA research project for next-generation spoken dialog interfaces ([135]) and was manually annotated with a multi-layered approach, including attribute-value information, Predicate-Argument-Structure (PAS) and dialog acts ([136]). This corpus includes human–human (HH) dyadic conversations of Italian speakers engaged in a problem-solving task in the domain of software/hardware troubleshooting, whereas the human–machine (HM) dialogs were acquired with a Wizard of Oz approach (WOZ) for the problem specification task only. In the corpus preparation phase, we extracted for each token the lemma, the turn id, the time-stamp and also the PAS label when available. PAS annotation was carried out applying the FrameNet paradigm as described in [134]. This annotation model covers a set of prototypical situations called *frames*, the frame-evoking words called *lexical units* or *target words* and the roles or participants involved in these situations, called *frame elements* (FEs). The latter are typically the syntactic dependents of the lexical units. All lexical units belonging to the same frame have similar semantics and valence (for details about the annotation scheme, see [136]). We adopted where possible the frame and frame element labels which were originally defined for the English FrameNet project. Some new frame definitions were introduced only in case of missing elements in the off-the-shelf resource.

An example annotation of two dialog turns is reported in Fig. **??**. For each Italian utterance transcription, we annotate the target words, the frame and its frame elements. The target words (in bold) *dire* and *chiamo* are assigned to a frame label in capitals, resp. TELLING and BEING_NAMED. This means that *dire* evokes the prototypical situation that is defined as TELLING in the FrameNet database, while *chiamo* evokes a situation called BEING_NAMED. Given that a frame is also characterized by some frame elements or semantic roles, *Addressee* and *Message* are the FEs expressed in the first utterance for TELLING, and *Entity* and *Name* in the second one for BEING_NAMED.

In this work, we focus primarily on the annotation of *target words*, and in particular on the criteria for identifying target words in a turn. In the early stages of the Berkeley FrameNet project[20], one frame per sentence was annotated, so just one target word was chosen in every sentence. More recently, the Berkeley group has started also another annotation effort, called *continuous-text annotation*, in which all possible valence-bearing words in a sentence are annotated as target words. In LUNA, we adopted an intermediate approach, following the idea that all *semantically relevant* target words with a syntactic subcategorization pattern have to be identified and annotated, possibly skipping the utterances with empty or fragmentary semantics (e.g. disfluencies). As expected, most of the targets annotated with our approach are verbs (almost 71% of the occurrences, while 14% are nouns and the rest adjectives and adverbs). In the FrameNet database, instead, the occurrences of verbal targets w.r.t. other PoS are more evenly distributed (44% verbs, 39% nouns, 16% adjectives), since the annotated sentences were selected in order to be representative of different *frames*, thus they are more balanced.

In order to assess the relation of the different annotation levels in the LUNA corpus, we performed the alignment of the multiple layers, viz. annotation of tokens, turns and *PAS* for 125 HH dialogs, mapping each token with turn ID and timestamp as well as with target / non-target) label. A summary of the corpus statistics is reported in Table 39.

[20] http://framenet.icsi.berkeley.edu/

Table 39: *LUNA corpus statistics for the training, development and test sets.*

	Train	Devel	Test
No. of dialogs used	94	11	20
No. of utterances used	4748	506	1131
Average no. of utterances per dialog	50.51	46.00	56.55
Total no. of tokens	34123	3479	7912
Average utterance length (in tokens)	7.19	6.88	7.00
Average dialog length (in mins)	3.21	3.23	3.39
No. of unique tokens	3307	872	1388
No. of lemmas	2312	688	1017
No. of PoS tags	24	17	15
No. of unique frames	204	107	135

6.3 Acoustic Correlates of Meaning Structure

We are interested in the problem of extracting meaning structures from spoken utterances in human communication. In Spoken Language Understanding (SLU) systems, parsing of meaning structures is carried over the word hypotheses generated by the Automatic Speech Recognizer (ASR)[135]. The automatic transcripts generated by the ASR are parsed and syntactic/semantic chunks are extracted. Such parsing models are either handcrafted (e.g. semantic grammars) or statistically trained from annotated corpora with ad-hoc and application specific concept labels. This computational model has had success in applications such as spoken dialog systems but may be limited by semantic coverage or high word error rates, in the case of unconstrained conversational systems. In this work we aim at discovering meaning components from direct measurements of acoustic and non-verbal linguistic features. Such components include the *most* semantically important word as well as its dependents within the semantic structures associated to a spoken utterance.

This approach to speech understanding is motivated by relevant research in speech and language processing, phonetics and language acquisition. In language acquisition, the most important questions are how to acquire words, their meaning while interacting in a physical and social context. In[137, 138] , meaning is grounded into machine actions and no semantic structure *bias* is assumed or exploited. In[139], meaning is directly learned from phone sequence distributions and visual features in the context of infant-directed speech. In computational linguistics the role of prosodic features to predict phrase structures has been well studied for cue phrases[140] and classification of intonational phrase boundaries[141]. Prosodic patterns have been also used as features for detecting and classifying dialog acts in conversational speech[142]. More recently the use of prosodic information has been applied to speech summarization[143]. From an acoustic point of view, prosody has been shown to manifest in variation of pitch, loudness, segment durations and specific manner of articulation.

Theories have been proposed to explain the way prosody can convey meaning through variation of these parameters. In[144, **?**] three *codes* characterize prosodic patterns. The *frequency* (or size), *effort* and *production* code. Following citeohala94 there is a link between paralinguistic assertion and lowering ones pitch. This link has profound origins as larger species, which are often perceived to have dominant or dangerous behavioral pattern, normally have larger vocal apparatus and produce lower pitch.

In section 2 we describe the meaning structures we aim at grounding into acoustic and linguistic features of the spoken utterances. Here we also describe how such model has been used to annotate the human-human dialog corpus, with a summary statistics of data split, used further in classification experiments. In section 3 we thoroughly exploit the acoustic features in order to use those in acoustic prediction. The target word classification with lexical features is described in section 4. Then a combined classification measurements with oracle accuracy is presented in section 5. We finally conclude in section 6.

6.3.1 Acoustic features

Audio recordings of the LUNA spoken dialog corpus were recorded as a mixed duplex channel with 8 KHz mono 16-bit pulse-coded modulation (PCM). The recordings were segmented into speech dialog turns and transcribed by human annotators. Afterwards, these turns were annotated, as described in the previous section. For the purposes of the experiment, all words in the recordings were labeled either as “Non-target word”, “Target word” or “Frame element”.

We have passed each individual turn through the forced alignment procedure with the Italian language ASR trained with the Sphinx-3 toolkit on the LUNA corpus.

We have extracted measurements of speech pitch (F_0), formant trajectories ($F_x, x = 1, 2, 3$), intensity (I_{tot}) and harmonicity (I_{hnr}) with the standard algorithms provided in the PRAAT toolkit citeboersma01. We have performed all measurements over a signal window of 40 ms with the frame rate of 100 Hz. The latter two measurements were combined to obtain an estimation of the intensity of a harmonic component (I_{harm}) of the speech signal. Employment of I_{harm} was motivated by the possible effect of acoustic interferences like environmental noise or other non speech-like sounds. Intensity of a harmonic component is also believed to better correspond to the intensity of phonation and paralinguistic stress. It is less distorted with wide-band energy bursts of plosives or fricatives. We have used the following formula for the depicted combination:

$$I_{harm} = I_{tot} + I_{hnr} - 10 \log_{10}(10^{I_{hnr}/10} + 1). \tag{10}$$

Note that when I_{hnr} is high then $I_{harm} \approx I_{tot}$. However, when I_{hnr} is sufficiently negative, then I_{harm} can also become negative.

The segmentation resulting from the forced alignment was then used to extract token-specific estimates. These absolute values were then compared to an average value of the given measurement throughout a whole turn. The measurements performed that way allow for a direct verification of the "code"-theories. A statistical analysis of the relative features has revealed that there exists a statistically significant difference between the mean values of the segmental relative features depending on which role a given segment possesses.

The average deviation of the maximal intensity of the harmonic component attained within a given segment from the average speech harmonic component intensity of the whole turn is higher for target words as opposed to frame elements and non-target words (see Fig. **??** for further detail). This observation is predicted by the effort code. It is notable that a maximal–to–mean intensity margin of the harmonic component has a clearance of 8 db between the target and all other words. An identically measured margin for an entire signal intensity I_{tot} does not exceed 1.5 db. This fact confirms our conjecture that the harmonic component intensity represents the paralinguistic stress pattern in a better way.

The average deviation of the minimal pitch frequency of the voiced interval attained within a given segment from the average pitch frequency of the whole turn is lower for target words as opposed to others (see Fig. **??** for further detail). The observation is in agreement with the frequency code.

The pitch dynamic range of target words is approximately 15 Hz larger in comparison to the average dynamic range of all of the words in that turn. This observation is in agreement with the effort code. The measurement is statistically significant with $p = 0.05$.

The average deviation of the mean duration of the voiced interval within a given segment from the average duration of the voiced intervals in the whole turn is larger for target words as opposed to frame elements and non-target words. At this point we did not reach a conclusion if voicing duration represents an independent feature or is a byproduct of the generally increased intensity of the harmonic speech component. The PRAAT performs pitch measurements through the use of a correlation statistics. Allegedly, it is able to uncover and track more intense harmonic structures from larger distances. We have additionally performed an analysis of the duration of the individual phonemes as it was recorded during the forced alignment. But we have not found a consistent pattern, that depends on the role of the segment under consideration.

The average inter-frame formant frequency difference (F_2 and F_3) is larger in comparison to the utterance mean for target words as opposed to frame elements and non-target words (see Fig. **??** for further detail). The formant dynamics is in agreement with the effort code. However, we expect the formant features to be more informative if this formant dynamics gets conditioned on the particular phoneme that is being uttered.

6.3.2 Target Word Classification with lexical features

The classification experiment involved identifying each token of a presegmented utterance as either a target word or a non-target word i.e. a binary classification task. This task was done using lexico-syntactic features: *(a) Part-of-Speech (PoS) tags* automatically added with the Chaos parser citebasili-02. *(b) Lemma information*, annotated with TreeTaggerciteschmidt. Human PAS annotation is primarily dependent on these two types of information for target / non-target word classification. Two further features comprise *(c) lowercased token* and its *(d) previous token*, including utterance boundary information.

The experiment was carried out using BoosTexterciteschapire00 classifier, that uses AdaBoost algorithm, which is initialized with a set of weak hypotheses by calling these weak classifiers in a series of iterations, and finally combining the weak hypotheses into a single rule.

For this task, baseline was established by using only tokens as feature;

the number of iterations was optimized for minimum false rejection rate, determined by best performance over the development data. To evaluate the classification performance, test output was scored using precision, recall, and $F1$-measure. Table 40 shows all results produced as baseline, single features and features in combination.

Table 40: *Results with Baseline, Single and Combined Features*

Features	Precision	Recall	F1-measure
Baseline (token)	0.759	0.648	0.699
PoS	0.655	0.825	0.730
Lemma	0.764	0.747	0.755
Token+PoS	0.787	0.800	0.793
Token+lemma+PoS	0.797	0.803	0.800
Token+Prev_tok+PoS	0.765	0.857	0.808
Token+Prev_tok+lemma+PoS	0.782	0.841	0.810

We observe that combined features: lowercased tokens, lemmas and PoS tags already achieve a better performance compared to baseline, since all these in combination convey a good amount of information about target words. The "previous token" feature adds a context to the combined classifier (i.e. the lowercased token, its lemma and PoS tag). Thus, the result is further improved using all four features in combination.

6.3.3 Combination of Lexical and Acoustic features

The effectiveness of the acoustic measurements in predicting target word classification task has been evaluated in combination with the lexical features. A multilayer perceptron (MLP) and a support vector machine (SVM) were used as classifiers for the acoustic features driven classifier. The MLP had one hidden layer with only 200 neurons. The output layer had two neurons corresponding to the two class labels that are being trained in the supervised way - "a target word" and "not a target word". The SVM was using a linear kernel. The feature vector contained all of the features depicted above.

The resulting classifiers on the test set were able to attain a performance level being $F1 \approx 0.3747$ for MLP and $F1 \approx 0.3397$ for SVM. The chance performance on the same test set has $F1 \approx 0.2972$. Thus, it is possible in principle to use acoustic information to infer semantics of a given segment. As is illustrated by Fig. **??**, the histograms of distributions of features are almost overlapping. It is a large number of experiments that allows us to record a statistically significant shift in the mean values of the features. We expect that better employment of a turn context may improve recognition performance.

Another possibility is to integrate results of acoustic classification over multiple instances of the same word. This leads to a potential application of the acoustic classifier in automated language acquisition. The words, and in general the whole linguistic contexts, which are consistently being marked as the frame-generating targets by the acoustic classifier may be incorporated into a model of the linguistic classifier, thus enabling an autonomous acquisition of the linguistic model from the spoken data only.

Table 41: *Performances of the best lexical and acoustic feature based classifiers and their oracle performances on the target word classification task*
.

Classifier	Prec.	Recall	F1
Lexical Features	0.782	0.841	0.810
Acoustic Features	0.247	0.774	0.375
Oracle Combination	0.935	0.913	0.924
Baseline Linguistic Classifier	0.759	0.648	0.699
Oracle Comb. (+ best acoustic)	0.926	0.811	0.865

As shown in Table 41, the figure of merit of combined systems could be as high as $92,4\%$. The combination of both acoustic and linguistic classifiers has the potential to be very accurate.

6.4 Conclusion

In the experiments with large amounts of spoken data we have observed a statistically significant deviation of the means of objectively measured segment parameters depending on the meaning of that segment. As such, our observation confirms our initial conjecture regarding the acoustic features grounding of the semantic elements within an utterance. Our findings support the theories of speech prosody and the effects of frequency and effort codes. An other important result is the correlation between acoustic measurements and a semantic representation of meaning that is linguistically motivated and consistent across domains. The preliminary classification experiments of the fine semantic structure elements are very encouraging and motivate the combination of acoustic and lexical features.

References

[1] A. Louis, A. Joshi, and A. Nenkova, "Discourse indicators for content selection in summarization," in *Proceedings of SIGDIAL 2010*, 2010.

[2] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber, "The Penn Discourse Treebank 2.0," in *Proceedings of the 6^{th} International Conference on Languages Resources and Evaluations (LREC 2008)*, Marrakech, Morocco, 2008.

[3] B. Grosz and C. Sidner, "Attention, intentions, and the structure of discourse," *Computational Linguistics*, 1986.

[4] M. Foucault, "L'archéologie du savoir," *Éditions Gallimard.*, 1969.

[5] B. Webber, M. Egg, and V. Kordoni, "Discourse Structure and Language Technology," *Natural Language Engineering*, vol. 1, no. 1, pp. 1–54, 2011.

[6] B. Wellner, "Sequence models and ranking methods for discourse parsing," Ph.D. dissertation, Brandeis University, 2009.

[7] D. Jurafsky and J. H. Martin, *Speech and Language Processing.* Pearson Education Inc., 2009.

[8] J. Hobbs, "On the coherence and structure of discourse," Stanford University, Tech. Rep., 1985.

[9] M. Halliday and R. Hasan, *Cohesion in English.* Longman, 1976.

[10] J. H. Martin, *English Text: System and Structure.* Benjamin, Amsterdam, 1992.

[11] J. Grimes, *The Thread of Discourse.* The Hague: Mouton, 1975.

[12] D. Litman and J. Allen, *Discourse processing and commonsense plans*, ser. Intentions in Communication, P. Cohen, J. Morgan, and M. Pollack, Eds. MIT Press, 1990.

[13] J. D. Moore and C. L. Paris, "Planning texts for advisory dialogs: capturing intentional and rhetorical information," *Computational Linguistics*, vol. 19, no. 4, 1993.

[14] W. Mann and S. Thompson, "Rhetorical structure theory: Toward a functional theory of text organization," *Text*, vol. 8, no. 3, pp. 243–281, 1988.

[15] J. van Eijk and H. Kamp, *Representing discourse in context*, ser. Handbook of Logic and Linguistics, J. van Benthem and A. ter Meulen, Eds. Elsevier, 1997.

[16] M. Henzinger, B.-W. Chang, B. Milch, and S. Brin, "Query-free news search," in *WWW 2003*, ser. 12th international conference on World Wide Web, 2003.

[17] M. A. Hearst, "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Computational Linguistics*, vol. 23, no. 1, 1997.

[18] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *Dekang Lin and Dekai Wu, editors*, ser. EMNLP, July 2004.

[19] A. C. Graesser, D. S. McNamara, M. M. Louwerse, Z. Qiang *et al.*, "Coh-metrix:analysis of text on cohesion and language," *CAI Behavior Research Methods, Instruments, & Computers*, vol. 36, no. 2, pp. 193–202, 2004.

[20] F. Y. Y. Choi, "Advances in domain independent linear text segmentation," in *NAACL 2000*, ser. 1st North American chapter of the Association for Computational Linguistics conference, 2000.

[21] D. J. Litman and R. J. Passonneau., "Empirical evidence for intention-based discourse segmentation." in *ACL Workshop*, ser. ACL Wrkshp on lntentionality and Structure in Discourse, 1993.

[22] C. Sporleder and M. Lapata, "Discourse chunking and its application to sentence compression," in *Human Language Technology Conference, Conference on Empirical Methods in Natural Language Processing*, 2005.

[23] E. Pitler and A. Nenkova, "Using syntax to disambiguate explicit discourse connectives in text," in *Proceedings of the 47^{th} Annual Meeting of the Association for Computational Linguistics and the 4^{th} International Joint Conference on Natural Language Processing*, 2009.

[24] D. Marcu and A. Echihabi, “An unsupervised approach to recognizing discourse relations,” in *the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002)*, 2002, pp. July 7–12.

[25] C. Sporleder and A. Lascarides, “Using automatically labelled examples to classify rhetorical relations: An assessment,” *Natural Language Engineering*, vol. 14, no. 03, pp. 369–416, July 2008.

[26] R. Soricut and D. Marcu, “Sentence level discourse parsing using syntactic and lexical information,” in *the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, May 27-June 1 2003.

[27] D. Marcu, “The rhetorical parsing of unrestricted texts: A surface-based approach,” *Computational Linguistics*, vol. 26, no. 3, pp. 395–448, 2000.

[28] J. Baldridge and A. Lascarides, “Probabilistic head-driven parsing for discourse structure,” in *the Ninth Conference on Computational Natural Language Learning (CoNNL)*, 2005.

[29] M. Collins, “Head-driven statistical models for natural language parsing,” *Computational Linguistics*, 2003.

[30] N. Asher and A. Lascarides, *Logics of Conversation.* Cambridge University Press, 2003.

[31] J. Baldridge, N. Asher, and J. Hunter, “Annotation for and robust parsing of discourse structure on unrestricted texts,” *Zeitschrift fur Sprachwissenschaft*, vol. 26, pp. 213–239, 2007.

[32] R. McDonald and F. Pereira, “Online learning of approximate dependency parsing algorithms,” in *European Association for Computational Linguistics (EACL)*, 2006.

[33] R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar, “Structured models for fine-to-coarse sentiment analysis,” in *Association for Computational Linguistics (ACL)*, 2007.

[34] C. Zirn, M. Niepert, H. Stuckenschmidt, and M. Strube, “Fine-grained sentiment analysis with structural features,” in *Proceedings of 5th*

International Joint Conference on Natural Language Processing. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, November 2011, pp. 336–344. [Online]. Available: http://www.aclweb.org/anthology/I11-1038

[35] P. D. Turney, “Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews,” in *, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL’02)*, 2002, pp. 417–424.

[36] S. Somasundaran, G. Namata, J. Wiebe, and L. Getoor, “Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification,” in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, 2009, pp. 170–179.

[37] H. H. Clark and S. E. Brennan, *Perspectives on socially shared cognition*, ser. Grounding in communication, L. B. Resnick, J. M. Levine, and T. J. S. D., Eds. American Psychological Association, 1991.

[38] E. Karagjosova, “The meaning and function of german modal particles,” Ph.D. dissertation, Saarabrücken Dissertations in Computational Linguistics and Language Technology, 2004.

[39] M. Zimmermann, *Discourse particles*, ser. Handbook of Semantics. Berlin: Mouton de Gruyter, 2009.

[40] R. Stalnaker, “Common ground,” *Linguistics and Philosophy*, vol. 25, pp. 701–721, 2002.

[41] M. Egg, “A unified account of the semantics of discourse particles,” in *Proceedings of SIGDIAL 2010*, ser. 11th Annual Meeting of the Special Interest Group in Discourse and Dialogue, 2010.

[42] W. Abraham, *Discourse Particles: Descriptive and Theoretical Investigations on the Logical, Syntactic and Pragmatic Properties of Discourse Particles in German*, ser. Pragmatics & Beyond New Series. John Benjamin Publishing, 1991.

[43] K. Ono, K. Sumita, and S. Miike, "Abstract generation based on rhetorical structure extraction," in *Proceedings of COLING*, 1994, pp. 344–348.

[44] D. Marcu, "To build text summaries of high qual- ity, nuclearity is not sufficient," in *Working Notes of the the AAAI-98 Spring Symposium on Intelligent Text Summarization*, 1998.

[45] J. Chai and R. Jin, "Discourse structure for context question answering," in *Proceedings of the Workshop at HLT-NAACL*, ser. HLT-NAACL 2004 Workshop on Pragmatics in Question Answering, 2004.

[46] M. Sun and J. Chai, "Discourse processing for context question answering based on linguistic knowledge," *. Knowledge Based Systems*, vol. 20, no. 6, pp. 511–526, 2007.

[47] M. Kirschner, R. Bernardi, M. Baroni, and L. T. Dinh, "Analyzing interactive qa dialogues using logistic regression models," *Lecture Notes in Computer Science*, vol. 5883/2009, pp. 334–344, 2009.

[48] H. Zong, Z. Yu, J. Guo, Y. Xian, and J. Li, "An answer extraction method based on discourse structure and rank learning," in *Natural Language Processing andKnowledge Engineering (NLP-KE), 2011 7th International Conference on*, 2011.

[49] E. Reiter and R. Dale, "Building applied natural language generation system," *Natural Language Engineering*, vol. 3, no. 1, pp. 57–88, 1997.

[50] H. Hendriks, *Information Packaging: From Cards to Boxes*, ser. Information sharing: Reference and Presupposition in Language Generation and Interpretation, K. van Deemter and R. Kibble, Eds. CSLI, 2002.

[51] R. Mitkov, "Introduction: Special issue on anaphora resolution in machine translation and multilingual nlp," *Machine translation*, vol. 14, pp. 159–161, 1999.

[52] D. Marcu, L. Carlson, and M. Watanabe, "The automatic translation of discourse structures," in *Proceedings of the 1st conference of the North American chapter of the ACL*, 2000, pp. 9–17.

[53] C. Hardmeier and M. Federico, “Modelling pronominal anaphora in statistical machine translation,” in *International Workshop on Spoken Language Translation*, Dec 2010.

[54] R. L. Nagard and P. Koehn, “Aiding pronoun translation with coreference resolution,” in *Proc. 5th Joint Workshop on Statistical Machine Translation and Metrics (MATR)*, 2010.

[55] G. Foster, P. Isabelle, and R. Kuhn, “Translating structured documents,” in *Proceedings of AMTA 2010*, 2010.

[56] D. Hardt and J. Elming, “Incremental re-training for post-editing smt,” in *Proceedings of AMTA*, 2010.

[57] *Enriching Automated Essay Scoring Using Discourse Marking*, ser. Discourse Relations and Discourse Markers, 1998.

[58] K. Ohtsuka and W. F. Brewer, “Discourse organization in the comprehension of temporal order in narrative texts,” *Discourse Processes*, vol. 15, pp. 317–336, 1992.

[59] F. Bex, *Arguments, stories and criminal evidence: A formal hybrid theory*. Dordrecht: Springer, 2011.

[60] F. Bex and B. Verheij, “Solving a murder case by asking critical questions: An approach to fact-finding in terms of argumentation and story schemes,” *International Journal on Reasoning*, 2011.

[61] L. Carlson, D. Marcu, and M. E. Okurowski, “Building a discourse-tagged corpus in the framework of rhetorical structure theory,” in *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*, ser. SIGDIAL ’01. Stroudsburg, PA, USA: Association for Computational Linguistics, 2001.

[62] M. Marcus, B. Santorini, and M. Marcinkiewicz, “Building a Large Annotated Corpus of English: the Penn Treebank,” *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[63] R. Prasad, A. Joshi, and B. Webber, “Exploiting scope for shallow discourse parsing,” in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10)*, Valletta, Malta, 2010.

[64] E. Miltsakaki, R. Prasad, A. Joshi, and B. Webber, "Annotating discourse connectives and their arguments," in *Proceedings of the HLT/NAACL Workshop on Frontiers in Corpus Annotation*, 2004, pp. 9–16.

[65] "Tsujii laboratory. genia project home page," 2003, last accessed: Feb, 2012. [Online]. Available: www-tsujii.is.s.u-tokyo.ac.jp/GENIA

[66] R. Prasad, S. McRoy, N. Frid, A. Joshi, and H. Yu, "The biomedical discourse relation bank," *BMC Bioinformatics*, 2011.

[67] B. Wellner and J. Pustejovsky, "Automatically identifying the arguments of discourse connectives," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic, 2007, pp. 92–101.

[68] V. Punyakanok, D. Roth, and W. Yih, "The importance of syntactic parsing and inference in semantic role labeling," *Computational Linguistics*, vol. 34, no. 2, pp. 257–287, 2008.

[69] R. Elwell and J. Baldridge, "Discourse connective argument identification with connective specific rankers," in *Proceedings of ICSC-2008*, Santa Clara, United States, 2008.

[70] N. Dinesh, A. Lee, E. Miltsakaki, R. Prasad, A. Joshi, and B. Webber, "Attribution and the (non-)alignment of syntactic and discourse arguments of connectives," in *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, Ann Arbor, Michigan, June 2005, pp. 29–36.

[71] E. Pitler, M. Raghupathy, H. Mehta, A. Nenkova, A. Lee, and A. Joshi, "Easily identifiable discourse relations," in *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, United Kingdom, 2008, pp. 87–90.

[72] Z. Lin, M.-Y. Kan, and H. T. Ng, "Recognizing implicit discourse relations in the Penn Discourse Treebank," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, 2009, pp. 343–351.

[73] E. Pitler, A. Louis, and A. Nenkova, "Automatic sense prediction for implicit discourse relations in text," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, Suntec, Singapore, 2009, pp. 683–691.

[74] S. Tonelli, G. Riccardi, R. Prasad, and A. Joshi, "Annotation of discourse relations for conversational spoken dialogs," in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, 2010.

[75] G. Minnen, J. Carroll, and D. Pearce, "Applied morphological processing of English," *Natural Language Engineering*, 2001.

[76] H. Yamada and Y. Matsumoto, "Statistical dependency analysis with support vector machines," in *Proceedings of 8th International Workshop on Parsing Technologies*, 2003.

[77] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *18th International Conf. on Machine Learning.* Morgan Kaufmann, 2001.

[78] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of HLT/NAACL*, 2003, pp. 213–220.

[79] A. McCallum, D. Freitag, and F. Pereira, "Maximum entropy markov models for information extraction and segmentation," in *the International Conference on Machine Learning (ICML)*, 2000.

[80] M. Collins, "Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms," in *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

[81] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2003.

[82] D. McAllester, M. Collins, and F. Pereira, "Case-factor diagrams for structured probabilistic modeling," in *the Converence on Uncer- tainty in Artificial Intelligence (UAI)*, 2004.

[83] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *JOURNAL OF MACHINE LEARNING RESEARCH*, vol. 6, pp. 1453–1484, 2005.

[84] H. Daume III, "Practical structured learning techniques for natural language processing," Ph.D. dissertation, University of Southern California, 2006.

[85] E. Alpaydin, *Introduction to Machine Learning.* MIT Press, 2004.

[86] J. Goodman, "Global thresholding and multiple-pass parsing." in *the Conference on Empirical Methods in Natural Language Processing (EMNLP 1997)*, 1997.

[87] X. Carreras, M. Collins, and T. Koo, "Tag, dynamic programming, and the perceptron for efficient, feature-rich parsing," 2008.

[88] E. Charniak and M. Johnson, "Coarse-to-fine n-best parsing and maxent discriminative reranking," in *Proceedings of the 43rd Annual Meeting of the ACL*, 2005.

[89] S. Petrov, "Coarse-to-fine natural language processing," Ph.D. dissertation, University of California at Bekeley, Berkeley, CA, USA, 2009. [Online]. Available: http://www.petrovi.de/data/dissertation.pdf

[90] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[91] D. Weiss and B. Taskar, "Structured prediction cascades," in *AISTATS 2010*, 2010.

[92] R. Caruana and D. Freitag, "Greedy attribute selection," in *Proceedings of the Eleventh International Conference on Machine Learning*, 1994.

[93] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach.* Prentice Hall, 2003.

[94] J. Hjorth, *Computer Intensive Statistical Methods.* London: Chapman and Hall, 1993.

[95] E. W. Weisstein, "“significant digits.” from mathworld–a wolfram web resource." [Online]. Available: http://mathworld.wolfram.com/SignificantDigits.html

[96] S. Ghosh, S. Tonelli, G. Riccardi, and R. Johansson, "End-to-end discourse parser evaluation," in *Proceedings of the Fifth IEEE International Conference on Semantic Computing (ICSC 2011)*, Palo Alto, United States, 2011.

[97] D. Klein and C. D. Manning, "Fast exact inference with a factored model for natural language parsing," *Advances in Neural Information Processing Systems 15 (NIPS 2002), Cambridge, MA: MIT Press*, 2003.

[98] J. Hopcroft, R. Motowani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley, 2000.

[99] T. A. Booth and R. A. Thompson, "Applying probability measures to abstract languages," *IEEE Transactions on Computers*, 1973.

[100] C. D. Manning and H. Sch utze, *Foundations of Statistical Natural Language Processing.* MIT Press. Cambridge, MA, 1999.

[101] D. M. Magerman, "Statistical decision-tree models for parsing," in *ACL 2005*, ser. 33rd Annual Meeting of the Association for Computational Linguistics,, 1995.

[102] M. Collins, "A new statistical parser based on bigram lexical dependencies," in *ACL 1996*, ser. 34th Annual Meeting of the ACL, 1996.

[103] ——, "Head-driven statistical models for natural language parsing," Ph.D. dissertation, University of Pennsylvania, Philadelphia, 1999.

[104] E. Charniak, "Statistical parsing with a context-free grammar and word statistics," in *NCAI*, ser. Fourteenth National Conference on Artificial Intelligence, 1997.

[105] E. Black, *et al.*, "A procedure for quantitatively comparing the syntactic coverage of english grammars," in *DARPA SNLW*, ser. the February 1991 DARPA Speech and Natural Language Workshop, 1991.

[106] S. Ghosh, R. Johansson, G. Riccardi, and S. Tonelli, "Shallow discourse parsing with conditional random fields," in *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*, Chiang Mai, Thailand, 2011.

[107] B. Grosz, A. K. Joshi, and S. Weinstein, "Centering: A framework for modeling the local coherence of discourse," *Computational Linguistics*, vol. 21, no. 2, 1995.

[108] M. Lapata, "Probabilistic text structuring: Experiments with sentence ordering." in *Proceedings of the 41st Meeting of the Association of Computational Linguistics*, 2003, pp. 545–552.

[109] R. Barzilay, M. Lapata *et al.*, "Modeling local coherence: an entity-based approach," in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL05)*, 2005.

[110] B. Grosz and J. Hircshberg, "Some intonational characteristics of discourse structure," in *Proceedings of the International Conference on Spoken Language Processing, Vol. 1*, Ohala *et al.*, Eds., vol. 1, 1992, pp. 429–432.

[111] R. Barzilay, L. Lee *et al.*, "Catching the drift: Probabilistic content models, with applications to generation and summarization," in *Proc. of NAACL-HLT, 2004*, 2004.

[112] R. Soricut, D. Marcu *et al.*, "Stochastic coherence modeling, parameter estimation and decoding for text planning applications," in *Proceedings of ACL-2006 (Poster)*, 2006, pp. 803–810.

[113] K. Toutanova, A. Haghighi, C. D. Manning *et al.*, "A global joint model for semantic role labeling," *Computational Linguistics*, 2008.

[114] M. Collins, "Discriminative reranking for natural language parsing," in *Computational Linguistics*. Morgan Kaufmann, 2000, pp. 175–182.

[115] H. Daume III, D. Marcu *et al.*, "Np bracketing by maximum entropy tagging and svm reranking," in *Proceedings of EMNLP'04*, 2004.

[116] M. Collins, "Ranking algorithms for named-entity extraction: Boosting and the voted perceptron," in *Proceedings of ACL 2002*, 2002.

[117] R. Johansson and A. Moschitti, “Syntactic and semantic structure for opinion expression detection,” in *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, 2010, pp. 67–76.

[118] M. Dinarelli, A. Moschitti, and G. Riccardi, “Re-ranking models for spoken language understanding,” in *Conference of the European Chapter of the Association of Computational Linguistics*, Athens,Greece, Apr. 2009, pp. 202–210.

[119] L. Huang and D. Chiang, “Better k-best parsing,” in *Proceedings of the 9th International Workshop on Parsing Technologies (IWPT 2005)*, Vancouver, Canada, 2005, pp. 53–64.

[120] The PDTB Group, “The Penn Discourse Treebank 2.0. Annotation Manual,” Institute for Research in Cognitive Science, University of Pennsylvania, Tech. Rep., 2008.

[121] S. Ghosh, R. Johansson, G. Riccardi, and S. Tonelli, “Improving the recall of a discourse parser by constraint-based postprocessing,” in *Proceedings of International Conference on Languages Resources and Evaluations (LREC 2012)*, 2012.

[122] L. Huang, “Forest reranking: Discriminative parsing with non-local features,” in *In Proc. of ACL*, 2008.

[123] M. Collins and N. Duffy, “New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron,” in *Proceeding of ACL 2002*, 2002.

[124] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Schwartz, and Y. Singer, “Online passive-aggressive algorithms,” *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.

[125] R. E. Schapire and Y. Freund, “Large margin classification using the perceptron algorithm,” *Machine Learning Journal*, vol. 37, no. 3, pp. 277–296, 1999.

[126] L. Shen and A. Joshi, “An svm based voting algorithm with application to parse reranking,” in *CoNLL 2003*, 2003.

[127] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

[128] R.-E. Fan, C.-J. Lin, K.-W. Chang, X.-R. Wang *et al.*, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, 2008.

[129] L. Shen and A. Joshi, "Ranking and reranking with perceptron," *Machine Learning Journal*, 2005.

[130] N. Littlestone, "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Machine Learning Journal*, pp. 285–318, 1988.

[131] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic learning in a random world.* Springer, 2005.

[132] M. Poesio, "Annotating a corpus to develop and evaluate discourse entity realization algorithms," in *LREC 2000*, 2000.

[133] ——, *The GNOME Annotation Manual, Fourth Edition.* [Online]. Available: http://www.hcrc.ed.ac.uk/\~{}gnome

[134] C. J. F. C. F. Baker and J. B. Lowe, "The Berkeley FrameNet Project," in *ACL*, ser. Proc. of ACL/Coling, 1998.

[135] F. B. D. H.-T. M. M. G. R. R. De Mori and G. Tür, "Spoken language understanding: A survey," *IEEE Signal Processing*, vol. 25, no. 3, 2008.

[136] S. Q. S. T. A. M. M. Dinarelli and G. Riccardi, "Annotating Spoken Dialogs: from Speech Segments to Dialog Acts and Frame Semantics," in *Workshop on Semantic Represent. of Spoken Lang.*, ser. Proc. of 2^{nd} Workshop on Semantic Represent. of Spoken Lang., 2009.

[137] A. Gorin, S. Levinson, and A. Sankar, "An experiment in spoken language acquisition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, January 1994.

[138] A. Gorin, D. Petrovska-Delacretaz, G. Riccardi, and J. Wright, "Learning spoken language without transcription," in *Proceedings of IEEE ASRU Workshop*, ser. Proceedings of IEEE ASRU Workshop, 1999.

[139] D. Roy and A. P. Pentland, “Learning words from sights and sounds: a computational model,” *Cognitive Science: A Multidisciplinary Journal*, vol. 26, 2002.

[140] J. Hirschberg and D. Litman, “Empirical studies on the disambiguation of cue phrases,” *Computational Linguistics*, vol. 19, September 1993.

[141] M. Wang and J. Hirschberg, “Automatic classification of intonational phrase boundaries,” *Computer Speech and Language*, vol. 6, 1992.

[142] A. Stolcke and et al., “Dialogue act modeling for automatic tagging and recognition of conversational speech,” *Computational Linguistics*, vol. 26, Septemeber 2000.

[143] S. Maskey and J. Hirschberg, “Summarizing speech without text using hidden markov models,” in *HLT-NAACL’06*, ser. Proc. of HLT-NAACL’2006, June 2006.

[144] C. Gussenhoven, “Intonation and interpretation: phonetics and phonology,” in *Proc. of Speech Prosody*, ser. Proc. of Speech Prosody, 2002.