Features for automatic discourse analysis of paragraphs

Finding features to detect rhetorical relations between sentences within paragraphs

Master thesis by
Daphne Theijssen

as a partial fulfillment of
The Research Master Language and Communication
Radboud University Nijmegen
Tilburg University

Supervisor
Dr. H. van Halteren

Second reader
Dr. Ir. B. Cranen

December 2007

Department of Linguistics
Radboud University Nijmegen

email: d.theijssen@let.ru.nl

A paragraph is coherent if its sentences are logically related to one another. You can achieve coherence by arranging details according to an organizing principle, by using transitional words and phrases, by using pronouns, by using parallel structure, and by repeating key words and phrases.

From: Kirszner, L.G. and S.R. Mandell (2002). *The Holt Handbook (sixth edition)*. Harcourt College Publishers, p. 103.

Acknowledgements

This thesis could not have been finished successfully without the help of a number of people.
First of all, I would like to thank my parents for giving me the opportunity to study and for supporting me in my choices and struggles. Second, I thank Harm for being of great support during my research by encouraging me in times of distress and by enabling me to spend both days and nights on my thesis.
The research presented in this thesis would not have been possible without my supervisor Dr. Hans van Halteren. From the moment I started the research master Language and Communication, he has played an important role in my educational programme. I wish to thank him for spending his precious time on my career and on this thesis. Also, I would like to thank the second reader of this thesis, Dr. Ir. Bert Cranen, for his critical review and helpful comments.
In the past two and a half years, I have changed my interests from English Language and Culture to Language and Speech Technology. I would like to thank Prof. Dr. Ans van Kemenade for encouraging me to do the research master. Also, I thank Drs. Suzan Verberne for teaching me many useful things that helped me in my education and will certainly be beneficial to my future career as a researcher in the field of language technology and computational linguistics.

Contents

Abstract

Although discourse analysis is considered useful for many applications in the field of language technology, automatic discourse parsing is still problematic. A widely accepted model for discourse analysis is Rhetorical Structure Theory, developed by Mann and Thompson (1988). Soricut and Marcu (2003) have developed the discourse parser SPADE, which automatically detects RST-relations between Elementary Discourse Units (EDUs) within a sentence. An automatic discourse parser that is able to find rhetorical relations at higher levels in the text is not yet available.

This thesis focusses on the rhetorical relations between (Multi-) Sentential Discourse Units ((M-) SDUs) – text spans consisting of one or more sentences – within the same paragraph of an English text. The goal of the research is to establish what information is useful in detecting these relations. To achieve this, potentially relevant features have been derived from literature on existing systems for discourse analysis and from a short study of a subset of the RST Discourse Treebank (Carlson et al. 2003). Next, most features have been made concrete in such a way that they could be extracted automatically. This was not possible for some features, e.g. ellipsis, newspaper style and world knowledge. We developed a metric for syntactic similarity, and introduced the feature NP simplification.

After all feature values were automatically extracted, they were offered to various machine learning algorithms. We have simplified the task of discourse parsing to a decision problem in which we decide whether an (M-) SDU is rhetorically related to either an immediately preceding or following (M-) SDU. This task was presented to machine learning algorithms together with the syntactic, lexical, referential, discourse and surface features in order to determine which features are most useful. Since developing an algorithm ourselves was beyond the scope of this thesis, we decided to experiment with existing implementations and thus without having full knowledge of their suitability given the task and data.

The performance of the classification algorithms was disappointing: Only the models of Naive Bayes (Demsar et al. 2004) and Maximum Entropy (Zhang 2004) reached significant improvement over the baseline of selecting the most common direction (*right*). Causes may be the small data set, the large number of features, the parameter settings, the artificiality of the task and/or the extent to which the algorithms are able to deal with the type of information provided. Assuming that the two algorithms are able to sift the information with some success and that the sifting is expressed in the model parameters, we have developed methods to rank the features according to their relevance on the basis of these parameters. This was also performed for the feature selection algorithms Relief (Kononenko 1994) and CSS (van Halteren, personal communication). From the four rankings based on the separate algorithms, a final ranked list was created. An in-depth study of the suitability of the algorithms and our methods is not included in this thesis, thus we must advise other researchers caution in taking the results described below for

granted.
As mentioned above, we have included five different feature types: surface, syntactic, lexical, reference and discourse. The most relevant surface features concern text characteristics that have also been covered by the other (more sophisticated) feature types. Syntax appears to be useful for the detection of rhetorical relations: the higher the syntactic similarity, the more chance the (M-)SDUs in question are rhetorically related. Lexical cues, which have been used by all researchers in the field of automatic RST annotation, are also beneficial in our task. A high word overlap or word similarity often means there is a rhetorical relation. As expected, reference features are also very useful: the presence of anaphora, personal pronouns, demonstrative pronouns, reference words and missing modifiers are cues that a rhetorical relation is present. Discourse structure also helps in finding rhetorical relations, perhaps due to the rather common newspaper style (with right-skewedness within paragraphs). The presence of direct speech (indicated by quotation marks) also predicts the presence of a rhetorical relation. Our data and method indicate that feature values should be based on the full (M-)SDU as well as the NUCLEUS sentence in the future.
The feature values, the Perl scripts and the feature relevance scores found can be downloaded from `http://lands.let.ru.nl/~daphne`. Since the source data is licensed (RST Discourse Treebank), it is not possible to include them on the website.
3

Samenvatting

Ondanks het feit dat *discourse-analysis* ("redeneringsanalyse") nuttig gevonden wordt voor vele toepassingen op het gebied van taaltechnologie, blijkt dat het automatisch discourse-parseren[1] nog steeds problematisch is. Een algemeen gebruikt model voor discourse-analysis is *Rhetorical Structure Theory* (*RST*, "Rhetorische-OpbouwTheorie"), ontworpen door Mann en Thompson (1988). In 2003 hebben Soricut en Marcu de automatische discourse parser SPADE ontwikkeld, een systeem dat RST-relaties tussen *Elementary Discourse Units* (*EDU's*, "Elementaire Redeneringseenheden") binnen zinnen automatisch weet te vinden. Een automatische discourse-parser die in staat is rhetorische relaties te vinden op hogere niveaus in de tekst is op dit moment niet beschikbaar.

Deze scriptie richt zich daarom op de rhetorische relaties tussen *(Multi-)Sentential Discourse Units* (*(M-)SDU's*, "(Multi-)Zins Redeneringseenheden"). Dit zijn tekstdelen bestaande uit een of meer zinnen binnen dezelfde alinea in een Engelse tekst. Het doel van het onderzoek was te bepalen welke informatie nuttig is voor het detecteren van deze relaties. We hebben in de literatuur over bestaande systemen voor discourse analysis en in een deel van de RST Discourse Treebank (Carlson et al. 2003) gezocht naar *features* ("eigenschappen") van de (M-)SDU's die hierbij mogelijk relevant zijn. Vervolgens hebben we de meeste van deze features concreet gemaakt zodat ze automatisch bepaald konden worden. Dit was niet mogelijk voor features als ellipsis, de schrijfstijl in kranten, en wereldkennis. Voor het feature *syntactische gelijkenis* hebben we zelf een maat ontwikkeld, en we hebben het begrip *NP-vereenvoudiging* geïntroduceerd.

De automatisch geëxtraheerde waarden van de features hebben we aangeboden aan verschillende machineleersystemen. We hebben de discourse-analysis-taak vereenvoudigd tot een beslissingsprobleem waarin we bepalen of een (M-)SDU rhetorisch gerelateerd is aan de voorafgaande of de volgende (M-)SDU. De machineleersystemen analyseren de gegevens om zo de richting van de relatie (naar links of naar rechts) automatisch te kunnen voorspellen. Omdat het ontwikkelen van een dergelijk systeem buiten de onderzoeksvraag van deze scriptie ligt waren we genoodzaakt te experimenteren met bestaande implementaties. Hierdoor hebben we geen volledige kennis over de geschiktheid van de systemen gegeven onze taak (het beslissingsprobleem) en data (de aangeboden tekst).

De prestaties van de hier toegepaste classificatie-algoritmes zijn teleurstellend: Alleen de modellen van Naive Bayes (Demsar et al. 2004) en Maximum Entropy (Zhang 2004) behalen een significante verbetering ten opzichte van de baseline waarin we kiezen voor de meest-voorkomende richting (rechts). Mogelijke oorzaken van de slechte resultaten zijn de kleine hoeveelheid beschikbare tekst, het grote aantal features, de gebruikte parameterinstellingen van de gebruikte systemen, de kunstmatigheid van de taak en de mate waarin de gebruikte systemen om kunnen gaan met de soort informatie die we

[1] *parseren* is het (taalkundig) ontleden van tekst

aanbieden. We nemen aan dat de twee voorgenoemde algoritmes tot op zekere hoogte in staat geweest zijn de informatie te zeven, en dit zeven terug te vinden is in de modelparameters. We hebben methodes bedacht om de features te sorteren naar hun relevantie op basis van deze parameters. Dit is ook gedaan voor de feature-selectie-algoritmes Relief (Kononenko 1994) en CSS (van Halteren, persoonlijke communicatie). Met behulp van de gesorteerde lijsten van de vier verschillende algoritmes is een uiteindelijke sortering gemaakt. Een grondige studie naar de geschiktheid van de algoritmes en onze methoden is niet aanwezig in deze scriptie. We adviseren andere onderzoekers daarom voorzichtig te zijn in het voor waar aannemen van onderstaande resultaten.

We hebben vijf verschillende feature-typen toegepast, namelijk die betrekking hebben op oppervlakkige teksteigenschappen, syntax, lexiale kenmerken, verwijzingen en redeneringsopbouw. De meest relevante oppervlakte-features geven grotendeels eigenschappen weer die ook opgenomen zijn in de andere (complexere) feature-typen. Syntaxis blijkt nuttig voor de detectie van rhetorische relaties: hoe groter de syntactische gelijkenis, des te groter de kans dat de (M-)SDU's in kwestie rhetorisch gerelateerd zijn. Lexicale aanwijzingen, al eerder gebruikt door alle andere onderzoekers op het gebied van automatische RST annotatie, zijn ook bevorderlijk. Een grote woordoverlap of woordgelijkenis betekent vaak dat er een rhetorische relatie is. Zoals verwacht zijn verwijzingsfeatures erg nuttig: de aanwezigheid van anaphoren, persoonlijke voornaamwoorden, aanwijzende voornaamwoorden, verwijswoorden en ontbrekende bepalingen zijn aanwijzingen dat er een rhetorische relatie aanwezig is. Redeneringsopbouw (argumentatie) helpt ook bij het vinden van rhetorische relaties, misschien vanwege de vaak voorspelbare schrijfstijl in kranten (waarin zinnen vaak terugverwijzen naar de eerste zin van die alinea). De aanwezigheid van gesproken taal (aangegeven met aanhalingstekens) voorspelt de aanwezigheid van rhetorische relaties ook. Uit onze gegevens en methode concluderen we dat de features in de toekomst het beste gebaseerd kunnen worden op zowel de hele (M-)SDU als de NUCLEUS-zin erbinnen.

Alle feature-waarden, de Perl scripts en de gevonden feature-relevantie-scores kunnen gedownload worden op `http://lands.let.ru.nl/~daphne`. Het is niet mogelijk om de bronbestanden beschikbaar te stellen vanwege rescricties (RST Discourse Treebank).

Chapter 1

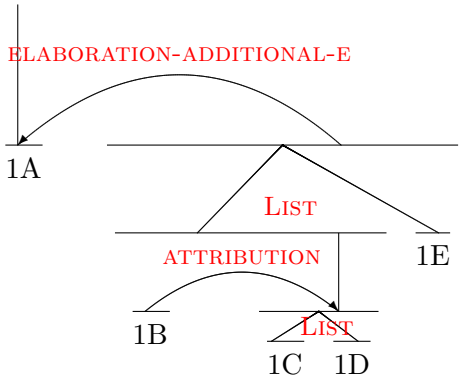
Introduction

In the field of language and speech technology, discourse analysis has gained popularity. It is considered useful for many applications in both language and speech technology, e.g. text, speech and image generation, text and video summarization, argument evaluation, machine translation and essay scoring (Taboada and Mann 2006). Being able to extract the relations automatically is therefore the aim of many researchers in the field. This thesis is dedicated to the automatic extraction of rhetorical relations as well. Since there have already been a number of achievements in this area, this chapter will provide some background information on previous research. The state-of-the-art situation leads to the research question that is central in this thesis. This chapter introduces the thesis objectives and explains the research environment in which the thesis is implemented. Lastly, it summarizes what the structure of the rest of the thesis will be.

1.1 Rhetorical Structure Theory

The model for discourse analysis that is most used by researchers in the field of language technology is Rhetorical Structure Theory (RST), which was developed by Mann and Thompson (1988). One of the motivations for choosing RST as the discourse analysis approach in many language technology applications is that human annotators show considerable consensus, demonstrating that the rules for assigning the rhetorical relations are clearly defined (Bosma 2005). Its popularity has led to the development of an RST Treebank of manually annotated English texts, which is available for training and testing purposes (Carlson et al. 2003). It consists of 385 Wall Street Journal articles from the Penn Treebank (Marcus et al. 1993) with a total of 176,383 words. Also, an annotation tool for manual RST discourse analysis has been created: RSTTool (O'Donnell 2000). This tool is user-friendly and allows saving the annotation in so-called rs3-files, which represent the hierarchy, relations and text of a document. The data and software supplies have again stimulated researchers to choose RST as the model for discourse analysis.
RST is based on the idea that rhetorical relations exist between spans of text, of which one span, called the Nucleus, is more important for the purpose of the author than the other spans, called the Satellites. They are also referred to as nuclearity roles. Sometimes spans are equally vital; the relation is then named multi-nuclear. In the RST Treebank, relations are not necessarily binary. This means that a Nucleus can have more than one Satellite, and that multi-nuclear relations can consist of more than two Nuclei. The smallest text spans that can hold rhetorical relations are named

Figure 1.1: Example RST tree



[Sun Microsystems Inc., a computer maker, announced the effectiveness of its registration statement for $125 million of 6 3/8% convertible subordinated debentures due Oct. 15, 1999.]^{1A} [The company said]^{1B} [the debentures are being issued at an issue price of $849 for each $1,000 principal amount]^{1C} [and are convertible at any time prior to maturity at a conversion price of $25 a share.]^{1D} [The debentures are available through Goldman, Sachs & Co.]^{1E} (wsj$_{0650}$)

Elementary Discourse Units (EDUs). In the RST Treebank, EDUs are clauses or clause-like units, of which the boundaries have been manually determined by interpreting lexical and syntactic clues (Carlson et al. 2003). In total, the RST Treebank contains 21,789 EDUs with an average word length of 8.1 words. An example tree from the RST Treebank is presented in Figure 1.1. In the figure, arcs point at NUCLEI, while straight lines indicate text spans in multi-nuclear relations. Two or more EDUs together can form a new span, which again holds a rhetorical relation with another text span. This way, a hierarchical structure is created for each text. The set of relations researchers choose depends on the goal of their research and the data set used.

1.2 Discourse parsers

The literature shows that a large number of automatic discourse parsers have already been created. Some have been developed for languages other than English, such as German (Reitter and Stede 2003), Thai (Wattanamethanont et al. 2005) or Dutch (Timmerman 2007), some follow other theories of discourse analysis (e.g. Schilder 2002, Polanyi et al. 2004, Webber 2004) and some are only partly automatic (e.g Stede and Heintze 2004). A state-of-the-art and publicly available system for automatic RST parsing of English texts is the one created by Soricut and Marcu (2003), which is Sentence-level PArsing of DiscoursE (SPADE). SPADE produces an RST tree for every sentence in the input, but makes no attempt to find relations between sentences and at higher levels. Other researchers, however, have aimed at extracting rhetorical relations between text spans consisting of at least one sentence, which has resulted in the discourse parser RASTA (Rhetorical Structure Theory Analyzer), developed by Corston-Oliver (1998), and LeThanh's (2004) system DAS (Discourse Analyzing System). Both systems are not generally available. A description of the approaches taken in the existing RST parsers

is provided in Chapter 2.

1.3 Research question

Apparently, a system for automatic discourse (RST) analysis that is suitable for text analysis at all text levels is not available. However, SPADE provides a first step towards it by being able to split sentences into EDUs and provide RST trees for each sentence. A second step would be to find relations between text spans consisting of at least one sentence within the same paragraph, at least when one assumes texts are well-constructed. In a well-constructed text, a sentence consists of EDUs, a paragraph of sentences and a coherent text of paragraphs. The goal of this master thesis is to tackle the problem of finding relations at the paragraph level of well-constructed texts:

> **Research question:**
> *Can we identify features that can be used to predict the presence of rhetorical (RST) relations between (Multi-)Sentential Discourse Units within paragraphs in English?*

In the above research question, we introduce the term *(Multi-)Sentential Discourse Unit*, which can be defined as follows:

> **(Multi-)Sentential Discourse Unit:**
> *A (Multi-)Sentential Discourse Unit or (M-)SDU is a text span with a length of at least one sentence and at most one paragraph, forming a discourse unit in a text.*[1]

The approach we take in establishing which features might be relevant for the detection of rhetorical relations between (M-)SDUs in paragraphs is to manually discover an inventory of possibly relevant features prior to automatically discovering which are indeed useful by applying machine learning algorithms. First of all, we read existing literature on the subject to find which features have already been applied by other researchers. Second, a number of RST trees in the Treebank will be considered to discover other potentially relevant features. Moreover, we will include surface information such as which words are present in the (M-)SDUs, etc., as well as linguistic information such as the part-of-speech tags and the syntactic structure of the (M-)SDUs. The result will be a long list of potentially relevant features, which are described and explained in Chapter 3. The total set of possibly useful features will then be used in various machine learning experiments in order to establish which features indeed appear to be relevant for discourse analysis at the paragraph level. We believe that in this set-up we have reduced the risk of missing relevant features as much as possible.
Since the total set of relation types in the RST model is relatively large compared to the size of the available training corpus (78 relation types for a corpus of 7123 sentences), only a small set of relations is frequent enough for robust assignment. We therefore decide not to include which relation is concerned in order to sidestep these sparseness problems. This decision to detect where (not which) rhetorical relations exist as a first step of discourse analysis has also been made by other researchers (e.g Marcu 1999). Another choice

[1] We should remark that (M-) SDU that cover a full paragraph are not relevant here since they cannot be rhetorically related to another (M-)SDU in the same paragraph.

made is ignoring the distribution of the nuclearity roles. This is because for instance the relations CAUSE, RESULT and CAUSE-RESULT are ambiguous and consequently the distribution is unpredictable. For CAUSE, the distribution is NUCLEUS-SATELLITE, for RESULT, it is SATELLITE-NUCLEUS, while for CAUSE-RESULT, it is NUCLEUS-NUCLEUS. Though the three relations are similar, they undesirably require different labels and nuclearity distributions. In sum, this master thesis employs underspecification in the sense that it focusses on determining whether there is a relation or not.
As described above, this thesis is a first step in finding rhetorical relations between (M-)SDUs in the same paragraph. It does so by establishing which features appear useful in finding them. In future research a system can be developed that makes use of the features found in the present thesis. Such a system would determine between which sentences or larger (M-)SDUs rhetorical relations exist.

1.4 Objectives

To reach the goal of the present master thesis, we try to accomplish the following objectives:

1. To manually identify features that might be relevant for establishing where (i.e. between which (M-) SDUs) rhetorical relations exist (Chapter 3);

2. To automatically extract the values of these features from the RST Treebank (Chapter 4);

3. To apply machine learning algorithms to establish which features are useful in deciding whether a relation holds between two (M-)SDUs (Chapters 5 and 6);

4. To evaluate the useful features and try to explain why they are useful and possibly relate them to the literature (Chapter 7).

1.5 Research environment

This thesis is related to the research project *In Search of the* Why, in which a Question Answering (QA) system is developed for *why*-questions specifically (e.g. Verberne et al. 2007). A QA-system aims to find the answer to a question in a given corpus automatically. Since the origin and development of the Internet, QA has gained popularity due to the restrictions of Information Retrieval (IR). In IR, a search engine supplies a list of relevant websites instead of the specific answer to the question (Ng et al. 2000). Several methods for QA have been developed, and the performance of QA-systems for closed-class (factoid) questions such as *who*- and *what*-questions is promising (Voorhees 2003). Verberne et al. (2007) aim at developing a QA system for *why*-questions specifically because they think that, in contradiction to factoid QA, the treatment of complex questions with explanatory answers such as *why*-questions requires the use of principled linguistic analysis. Since the answer to a *why*-question is a fragment of text that is expected to be rhetorically related to what is questioned, the system exploits RST in order to recognize potential answers in a text.
In their article, Verberne et al. (2007) remark that in a future application of *why*-QA using RST, the system will not have access to a manually annotated corpus as it had

during the presented research. In other words, it will have to function by employing automatically annotated data. For this purpose, a system for automatic RST parsing should be obtained or, if not available, be created. Since it is fully automatic, publicly available and developed for English, SPADE (Soricut and Marcu 2003) is suitable for use in the *why*-QA system. However, the data used in Verberne et al. (2007) demonstrate that more than 50% of the relations between unique text spans addressed by *why*-questions are not relations within sentences, but at higher levels of the text (but almost always within paragraphs). Finding features that help establishing where rhetorical relations exist between (M-)SDUs in the same paragraph, as proposed for this thesis, is therefore beneficial to the project. The choice for the underspecification of rhetorical relations is also suitable in the context of the *why*-QA project. Verberne et al. (2007) have found that the list of RST relations associated with *why*-questions is broad and a pre-established list of relevant relations should therefore not be applied too rigidly. Although the present thesis is related to the *why*-QA system, its goal is to find features for the detection of RST relations between (M-)SDUs in general, not for text (spans) and relations that are useful for retrieval of potential *why*-answers solely. The data and method applied are therefore not concentrated on *why*-QA.

1.6 Organization of the thesis

The organization of this thesis is as follows: First, an overview of the existing systems for automatic RST annotation is provided in Chapter 2. Chapter 3 describes the procedure of finding potentially relevant features and shows which features are selected and why. Next, the methods for making the found features concrete and automatically extracting their values are presented in Chapter 4. Chapter 5 describes how we apply machine learning in order to determine which features are indeed useful for finding rhetorical relations between (M-) SDUs, and Chapter 6 shows the corresponding results. In Chapter 7, the found relevant features are evaluated. The final chapter, Chapter 8, contains the conclusion.

Chapter 2

Existing systems for automatic RST annotation

As already stated in the Introduction (Chapter 1), many systems for automatically creating discourse trees have been developed by various researchers. Since our research goal concerns the rhetorical structure of paragraphs written in English, the focus of this chapter is on automatic discourse parsers that have been developed for English. Moreover, we only include descriptions of systems that also employ Rhetorical Structure Theory (RST) and are fully automatic.
This chapter starts with a description of the state-of-the-art automatic RST parser SPADE, developed by Soricut and Marcu (2003). SPADE is developed for the annotation of English sentences. No attempt is made to find relations between (M-)SDUs. In fact, the existence and availability of this parser is one of the reasons that this thesis is limited to finding relations between (M-)SDUs. A complementary system that detects rhetorical relations between (M-)SDUs is only available for Dutch (Timmerman 2007). Although the focus of this thesis is on English, the approach of Timmerman could still be informative, especially because it also concerns the RST relations between (M-)SDUs. As far as we know, this system is the most recently developed one.
Later in this chapter, systems are described that are not available for use but may nonetheless be interesting for the present thesis. The descriptions start with the Rhetorical Structure Theory Analyzer (RASTA) (Corston-Oliver 1998), which was developed as part of the Microsoft English Grammar (MEG) and is therefore not freely available. Next follow descriptions of two methods by Marcu (1999, 2000). In 2003, Marcu's research on automatic RST annotation resulted in SPADE, which he created with Soricut (Soricut and Marcu 2003) and which is the only system he has made available on his personal website. Lastly, the Discourse Analyzing System (DAS) (LeThanh 2004) is described, which has been lost (personal correspondence).

2.1 SPADE (Soricut and Marcu 2003)

In 2003, Soricut and Marcu published an article describing the procedure of their discourse annotation system SPADE (Sentence-level PArsing of DiscoursE). Moreover, Carlson, Marcu and Okurowski have developed an RST Discourse Treebank consisting of a number of Wall Street Journal texts from the Penn Treebank (Marcus et al. 1993), as mentioned in the Introduction (Chapter 1). Opposed to Marcu's previous research

(which will be described in sections 2.4 and 2.5), the approach taken in this system is fully probabilistic and makes use of the RST Discourse Treebank. The probabilistic model is trained on 347 articles, while 38 articles are reserved for testing purposes. The system creates a binary RST tree for each sentence that is offered to it. Though the RST Discourse Treebank also includes non-binary relations, the choice of creating only binary trees is justified by the fact that 99% of the relations in the corpus are binary. SPADE consists of (1) a discourse segmenter and (2) a discourse parser.
SPADE uses established methods for paragraph and sentence splitting. It then splits sentences into EDUs by determining the probability of the presence of an EDU boundary on the basis of the RST Discourse Treebank, and inserting a boundary marker when it is higher than 0.5. To determine the probability, it checks the syntactic trees to find the highest node with the word in question as its head, and having a sister node at its right. For such nodes, the node itself, its parent and its siblings are considered as features. Since syntactic structure alone is not sufficient to treat certain (ambiguous) constructions, lexical information is employed as well. For example, an EDU boundary can be inserted between *passed* and *without*, but not between *priced* and *at*. The found EDU boundaries are added to the Penn Treebank syntactic trees.
The syntactic trees with their EDU boundaries are offered to the second component in the SPADE system, which is the discourse parser. This component is also probabilistic and consists of (1) a parsing model, which establishes the probability of every possible discourse tree, and (2) a discourse parser that creates the desired tree. The parsing model determines how probable it is that two EDUs or text spans are related with a certain relation. They do this by rewriting the RST trees as a set of triplets *R[i,m,j]*, in which *R* denotes a rhetorical relation between text span A consisting of EDUs *i* until *m*, and a text span B consisting of EDUs *m+1* until *j*. The parsing model determines the probabilities of the structure *[i,m,j]* and the relation *R*, and multiplies them to retrieve a single score. Ideally, the probabilities are taken directly from the RST Discourse Treebank, but this is not possible due to sparseness problems. Therefore, Soricut and Marcu apply a rather complex method (see Soricut and Marcu 2003) to derive certain structures and characteristics that represent the desired relations. Using different projections of this representational set, the probability of possible structures and relations for a given EDU or text span pair is established. Filters are applied to ensure that only information relevant for the given text pair is extracted. Using Maximum Likelihood Estimation, the probabilities for each possible discourse parse are determined on the basis of the aforementioned procedure and the RST training corpus. Furthermore, interpolation is applied to reduce the effect of data sparseness. Once the parsing model has established the range of possible RST trees and their corresponding probabilities, the discourse parser can build the tree. This is performed in a bottom-up fashion using a dynamic programming algorithm[1]. If more than one structure is possible for the same text span, the structure with the highest probability is selected, while the others are discarded.
Soricut and Marcu evaluated their discourse parsing system by calculating the Precision, Recall and the F-score (van Rijsbergen 1979) following the Parseval metric for parser evaluation (Black et al. 1991):

$$\text{Precision} = \frac{\text{number of system-output relations in gold standard}}{\text{total number of system-output relations}}$$

[1]Soricut and Marcu (2003) give no details about the dynamic programming algorithm they have applied

$$\text{Recall} = \frac{\text{number of gold-standard relations in system output}}{\text{total number of gold-standard relations}}$$

$$\text{F-score} = \frac{2 \times Precision \times Recall}{(Precision + Recall)}$$

The numerators in the divisions of the first two formulas are in fact the same since they both show the number of overlapping relations in the system output and the gold standard.
Baselines are the system developed by Marcu (2000) and a method in which right-branching (NUCLEUS-SATELLITE) trees are produced. The inter-annotator agreement of two humans who both annotated a part of the corpus is taken to be the performance ceiling. Different versions of SPADE were tested, namely a version in which the syntactic trees offered are the gold standard (*S+*), and one in which they are automatically derived by the Charniak parser (*S-*) (Charniak 2000). Similarly, the system was offered either the manually inserted EDU boundaries (*B+*) or those found by the SPADE segmenter (*B-*). The obtained F-scores for the detection of relations (regardless of which relation label was selected) are presented in Table 2.1.

Table 2.1: Parseval F-scores obtained by SPADE

System	Parseval F-score
Right-branching	64.0
Marcu (2000)	67.0
SPADE S-B-	70.5
SPADE S+B-	73.0
SPADE S-B+	92.8
SPADE S+B+	**96.2**
inter-ann. agr.	92.8

SPADE outperforms the baselines, even when provided with automatically derived syntactic trees and EDU boundaries. As expected, the best results are obtained by SPADE using gold standard trees and manually inserted EDU boundaries. As Soricut and Marcu (2003) mention in their article, their system reaches near-human performance on the task of detecting rhetorical relations within sentences if provided with manually determined EDU boundaries.

2.2 Timmerman (2007)

In his master thesis, Timmerman (2007) describes a method for automatically determining RST-relations in Dutch medical texts. The system consists of three components: (1) a segmenter, that splits the input text into sentences, (2) a recognizer, which lists all possible relations and from these selects those nodes that form the most probable tree, and (3) a tree-builder that creates an rs3-file on the basis of the list of selected nodes produced by the recognizer. Like other researchers, Timmerman does not include all

RST-relations defined by Mann and Thompson (1988), but only a more clearly defined set consisting of combinations of original relations: ELABORATION, CAUSE, RESULT and CONCESSION.

To prevent problems with the evaluation of the systems caused by erroneous segmentation, Timmerman decided to take sentences as the smallest discourse units. The segmentation (component 1) in sentences is based on punctuation only.

The recognizer (component 2) employs a static list of discourse markers which are assumed to relate to certain RST-relations. Discourse markers used are conjunctive adverbs, pronouns, medical discourse markers and relation markers. Conjunctions are not included because, Timmerman argues, they can demonstrate a relation that is either within or between sentences. Previously described approaches use conjunctions as key cues for RST-relations, and leaving them out of consideration seems very rigid. Due to similar ambiguity problems, adjectives are not employed as discourse markers either. To establish which RST-relation holds between the (M-)SDUs concerned, knowledge must be derived from thesauri such as WordNet (Fellbaum 1998). Moreover, repetition of *key words* (Timmerman 2007) can hint at the existence of an RST-relation, though it does not help in establishing which exactly it is. The approach taken here is to define key words as all nouns in the text, and to compare two adjacent sentences for their key words. When there is no repetition, the first sentence is compared to a third sentence, etc. The selection of key words can lead to errors, since it needs syntactic or at least Part-of-Speech information.

From all possible relations between all (M-)SDUs, a shorter list is created which consists only of those nodes that form the most probable discourse tree. Three assumptions are made: (1) the text is coherent, (2) the most important piece of information is usually provided first (so NUCLEUS-SATELLITE), and (3) the text is from the Dutch medical domain. On the basis of the discourse markers, the thesauri and key word repetition, for each adjacent pair of sentences in combination with each possible relation, the likelihood is indicated, for example: *[3 4 N 100 E*LABORATION *]*. Here, the likelihood of an ELABORATION relation between sentence 3 and 4 of the text, of which sentence 3 is the NUCLEUS, is 100. The discourse markers mentioned above are ranked according to importance, and the order helps determining the likelihood. Moreover, the position of the discourse marker in the sentence plays a role. The likelihood is not only based on the occurrence of discourse markers and key word repetition; it can also be raised when for example more than one relation type is possible for the same sentence pair in the same information order (e.g. NUCLEUS-SATELLITE). This indicates the order is probably correct, and the likelihood should therefore be increased. Each present cue or characteristic that Timmerman assumes to be relevant increases the likelihood (e.g. with 60 if a conjunctive adverb is present), which means there is no fixed range. For sentence pairs that are not adjacent, the likelihood is estimated by calculating the percentage of key words that match. In this case, the likelihood is always in the range between 0 and 100.

Only the nodes that have a likelihood that is higher than a fixed threshold are used in determining which combination of nodes is the most probable tree (i.e. has the highest likelihood). The threshold is slightly higher than the likelihood of the weakest cues to ensure that a single occurrence of a weak cue does not lead to the selection of a relation. Stronger cues are needed for this. The final list of nodes are used in the last component to build the RST-tree.

Following the approach of Marcu (2000), the output discourse tree is compared to a gold

standard in order to evaluate the system performance. The gold standard was created by three different annotators. To check inter-annotator agreement, some of the evaluation texts were annotated by all three annotators, and it appeared that the differences between the annotators are as large as between the eventual gold standard and the system output. For these texts, a 'combined tree' was created, showing all relation structures and types that occur in at least two of the three annotated trees. For a small number of short texts, Timmerman established the proportion of all relations in the gold standard that were also found by the system. Being found here means that the system detected a relation between the same text spans as in the gold standard. This detection of relations reaches above 50% in all evaluated cases. Of these found relations, the relation labels and nuclearity distributions were correctly assigned in 25% to 100% of the cases. Although the thesis includes an elaborate discussion of the evaluation methods and results with numerous examples, the results themselves are not clearly presented. There are no figures concerning the number of texts or relations evaluated. Also, only accuracy was determined while the distributions of the relations and their labels are not mentioned. Precision and recall would have been more informative measures here.

2.3 RASTA (Corston-Oliver 1998)

One of the first systems for automatic discourse annotation following RST was RASTA (Rhetorical Structure Theory Analyzer), which was designed by Corston-Oliver for his PhD (Corston-Oliver 1998). Corston-Oliver states that opposed to existing methods for RST annotation at that time, RASTA not only employs superficial information such as the occurrence of certain cue words (e.g. *because*), but also deep linguistic information. His argument for involving this kind of information is that it can help disambiguating many-to-many relationships between linguistic form and rhetorical relation, especially by combining various linguistic aspects. In total, 13 rhetorical relations have been selected for automatic RST annotation: Asymmetric Contrast, Cause, Circumstance, Concession, Condition, Contrast, Elaboration, Joint, List, Means, Purpose, Result and Sequence. The choice for the set of relations is based on the text genre of the data, which has been taken from Encarta. Corston-Oliver's system consists of three components: (1) a segmenter that segments text into EDUs, (2) a component that determines which relations are possible between every (not necessarily adjacent) EDU pair, and (3) a tree-builder that constructs a discourse tree out of the output of the previous component.

Before being offered to RASTA, the input text is parsed by a parser in the Microsoft English Grammar (MEG, of which RASTA is also a component), which not only produces syntactic trees, but also logical forms that name the syntactic roles in a clause. The parsed text is then presented to the first component of the system (the segmenter). Each syntactic node in the logical form is checked for a number of characteristics, such as whether the head is a verb or the constituent is elliptical. If the node possesses the desired characteristics, the constituent is considered an EDU.

Corston-Oliver employs deep linguistic information for the purpose of determining which relations are possible at what positions, and how probable this is (the second component). For each of the 13 relation types, a number of syntactic constraints have been formulated. For example, the relation Sequence is symmetric, meaning it holds between two Nuclei. One of the constraints of Sequence is, therefore, that the one text

unit may not be syntactically subordinate to the other. Syntactic characteristics taken into consideration are clausal status (e.g. whether a clause is subordinate), anaphora (referring to an entity), deixis (meaning that a word or expression relies completely on the context), referential continuity (elaborating on a topic by using reference words), tense (e.g. past), aspect (e.g. progressive) and polarity (whether there is negation or not). Checking the syntactic criteria radically decreases the number of relations that the system can select, making it more computationally efficient than assigning all relations and removing them from the output later.

Next to the list of syntactic criteria, each relation has its own list of cues, each of which is accompanied by a heuristic score. This score is manually determined on the basis of linguistic intuitions, since no RST Discourse Treebank from which the scores could be derived was available at the time. An example of a cue for SEQUENCE is a sentence conjunction such as *and later then*. The heuristic scores are used since some cues are stronger indicators of a relation than others. The second component of the system creates a list of all possible relations between all EDU pairs that match the syntactic criteria. The heuristic scores based on the present cues are also added, and the list of relations is ordered according to the scores.

The third and final system component, the tree builder, considers the list of possible relations. This is also the component that tries to detect relations between (M-) SDUs. It starts by selecting the relation ranked highest, since that is the most probable according to the heuristic scores. It then moves to the second relation in the set. When one of the EDUs in the relation is already linked to another EDU, the relation is not always possible, in which case the next relation is tried. After considering all possible relations, the whole text is represented in one tree, which is saved. After this or in case a tree could not be constructed, the system tries a new 'route' of going through the list of relations (for details the reader is referred to Corston-Oliver's PhD dissertation, 1998). This is repeated until either the desired maximum of output trees is reached or all possible paths have been followed. The approach for finding relations between (M-)SDUs appears to be fully based on the already hypothesized relations between EDUs. This is possible because all possible EDU pairs have been considered, not only those that are adjacent. In a final stage, the binary output trees are converted into N-ary trees. In such trees NUCLEI can have more than one SATELLITE.

Performance results or analyses are not presented in the dissertation. Corston-Oliver claims that his approach is promising, but there are no figures to support this. The performance of the system can therefore not be compared to other systems.

2.4 Marcu (1999)

In 1999, Marcu presented a decision-based approach to automatic RST annotation that involved a shift-reduce algorithm. This approach differs greatly from RASTA by Corston-Oliver (1998) described above because it is not rule-based. Since the RST Discourse Treebank was not developed yet, a manually annotated data set was created in order to apply machine learning. In total, 90 texts (news stories, scientific texts and editorials) were annotated by at least two annotators, who used a taxonomy of 71 rhetorical relations, grouped in 17 clusters: APPOSITION - PARENTHETICAL, ATTRIBUTION, CONTRAST, BACKGROUND - CIRCUMSTANCE, CAUSE - REASON - EXPLANATION, CONDITION, ELABORATION, EVALUATION - INTERPRETATION, EVIDENCE, EXAMPLE, MAN-

ner - Means, Alternative, Purpose, Temporal, List, Textual and Other. They had an annotation tool at their disposal, and performed three subtasks: (1) finding EDU and parenthetical unit boundaries, (2) creating trees for the EDUs and text spans, and (3) selecting the appropriate relation types for the tree. Marcu's (1999) approach consists of (1) discourse segmentation and (2) a discourse parsing, both addressed probabilistically.

The values for a number of features are determined for each lexeme, enabling a machine learning algorithm to learn where to insert a discourse boundary (component 1). Features used for this purpose are the Part-of-Speech (POS) tags of the two preceding and following lexemes and of the lexeme itself, the possible presence of a discourse marker, the possible presence of an abbreviation, the position of the discourse marker if present, punctuation marks such as commas and dashes, and whether the word in focus is a verb. Secondly, a shift-reduce algorithm is used for a first step in rhetorical parsing. The algorithm has to choose from a number of actions to arrive at the desired relation for each individual EDU pair. The first is a shift action, which means moving to a next EDU in the text. Then there are six reduce actions, covering the three different nuclearity distributions (Nucleus - Satellite, Satellite - Nucleus and Nucleus - Nucleus) and both ways to attach an EDU (binary or as an extra child). Since there are 17 relation clusters, for each of which a reduce action can be applied, the system has to be trained to choose from 17 * 6 + 1 = 103 actions. Each time the system has to make this decision is a separate *case* in the machine learning setting. Features applied are the number of subtrees already created and the remaining number of EDUs, the structures of trees at the same text level (sentence, paragraph, etc.), the two preceding and following words and POS-tags, the word and POS-tag in question, whether there are discourse markers in the last three subtrees or in the new EDU and if present at what position in the EDU they are, the five last shift-reduce actions performed, the semantic similarity between the new EDU and EDUs in the three most recent subtrees, and WordNet-based measures of word similarity. The latter include e.g. synonyms (different words with the same meaning), hypernyms (general words that are used to refer to specific items within this general group), cohyponyms (words that are in the same general group) and antonyms (words that are opposites of each other). By applying the shift-reduce algorithm sequentially, an RST tree can be created for a whole text.

The presented approach is evaluated by calculating the labeled precision and recall for EDU segmentation, the recognition of relations between text spans, determining nuclearity roles and selecting the type of relations. The scores decrease significantly when not the gold standard EDUs, but the automatically selected EDUs are offered to the parser. When trained and tested on Wall Street Journal texts, and without receiving any human input, the system achieves the results in Table 2.2. The scores for the annotators are established by computing the average labeled precision and recall with respect to the discourse trees built by other annotators.

Marcu also trained and tested on other text genres. The results show that the text genre of the training and test set greatly influences the performance of the discourse parser. The conclusions are that the features applied are sufficient for recognizing where (between which EDUs or text spans) relations exist, and to label the nuclearity roles, but appear not optimal for the selection of relation types. A proposed solution is the use of more features and more data. The evaluations applied do not provide any information on how the quality of the shift-reduce algorithm in tracing rhetorical relations within sentences relates to the tracing of relations between (M-)SDUs. Since the results are

Table 2.2: Precision and Recall reached by Marcu's (1999) discourse parser

	annotators		system	
	Recall	Precision	Recall	Precision
EDUs	85.1%	86.8%	18.1%	95.8%
spans	79.9%	80.1%	34.0%	65.8%
nuclearity	67.6%	77.1%	21.6%	54.0%
relations	73.1%	73.3%	13.0%	34.3%

based on individual cases solely, it might well be that they concern relations at lower levels in the text and therefore do not reflect the performance on relations higher in the text.

2.5 Marcu (2000)

A different approach is taken in Marcu (2000), in which he explores the use of surface information and a manually established model. The system outputs binary discourse trees in which the leaves (EDUs) are clauses or clause-like units, between which 54 different relations may hold. It consists of three components: (1) segmentation in EDUs and recognition of discourse markers, (2) determining the relations between text spans, and (3) establishing nuclearity roles and the length of text spans, and building the discourse trees. Marcu uses cohesion, connectives (cue phrases) and a manually constructed mathematical model of valid rhetorical structures. The cohesion measures applied are word co-occurrences and more sophisticated forms of lexical cohesion. Marcu selected 450 different cue phrases from literature by various researchers, and traced them, together with a 300 word window, in the Brown corpus. In total, he selected over 7,600 texts, of which he managed to manually analyze 2,100. All the cue phrases were saved together with a number of cue phrase characteristics taken from the 300 word window, e.g. its orthographic environment (e.g. punctuation), its functional role (either sentential, discourse or pragmatic) and the rhetorical relation(s) connected with the cue phrase. When a cue phrase could indicate more than one relation, all possibilities were enumerated. The list of cue phrase characteristics is used in all three components of the system.

The cue phrases and shallow processing are applied for the purpose of discourse segmentation (the first component). The segmentation in sentences and paragraphs is fully based on punctuation. During the manual analysis of the 2,100 cue phrases in the corpus, Marcu generated a set of eleven actions for automatically inserting EDU boundaries in sentences. They describe different situations, e.g. a cue phrases preceded by a comma. For each cue phrase from the corpus, the associated action is saved, together with information on its position in the EDU (beginning, middle or end). The segmentation in EDUs is based on cue phrases and their corresponding corpus-based rules (actions). While applying these rules, the segmenter checks whether the cue phrase is in fact a discourse marker in the given context. Text that remains after the segmentation on the basis of cue phrases is considered as clause-like units and are thus separate EDUs. Testing on five texts from the same genre (scientific texts), the recall and precision reached

are 51.2% and 95.9% respectively.
The second component consists of the determination of relations between text spans. At each text level (sentence, paragraph and section), hypotheses are formulated that describe relations with other EDUs or text spans at the same level. Relations are over-generated, which is solved in a later stage, when the discourse trees are actually built. Discourse markers are traced by applying regular expressions, with help of the Unix tool lex.
In previous research, Marcu argued that "*rhetorical relations that hold between large textual spans can be explained in terms of similar relations that hold between their most important elementary units*". In Marcu (2000), Rhetorical relations between larger text spans are called *extended relations* and those between EDUs *simple relations*. Marcu warns that his observation should not lead researchers to conclude that they should only focus on finding relations between the NUCLEUS EDUs. For example, when three paragraphs are introduced with the cues *First*, *Second* and *Third*, a LIST or similar relation will be found. The content of the paragraphs, however, might have led to a completely different relation. It thus depends on the context whether a simple relation is acceptable in the discourse tree. Marcu (2000) states that "*a rhetorical structure tree is valid only if each rhetorical relation that holds between two spans is either an extended rhetorical relation or can be explained in terms of a simple rhetorical relation*". He names this the *Compositionality Criterion*.
The use of discourse markers is sufficient for the detection of simple relations at the sentence level in most cases. No connection could be found between discourse markers that signify relations at sentence level and those that signify relations at paragraph or section level. Marcu was able to conclude that discourse markers indicate relations between paragraphs when they are either in the first or in the last sentence of the paragraph.
To be able to find relations in cases where no discourse marker is present, and to find relations at the paragraph and section level, word co-occurrence is measured. Stop words such as *the*, *a* and *and* are left out of consideration, as well as suffixes. When the lexical similarity is above a threshold, the hypothesized relation is ELABORATION or BACKGROUND, otherwise it is JOINT. The threshold is based on the average similarity in the text at the text level concerned (e.g. the paragraph level). This method for finding relations between (M-)SDUs seems rather simplistic, and it is not certain whether it will be helpful in our search for useful features that help establishing where rhetorical relations exist between (M-)SDUs in paragraphs. At the end of this system stage, all hypothesized relations at the three text levels are saved and offered to the tree building component.
For the last component, Marcu designed a mathematical model that describes the form of a correct discourse tree, making use of the compositionality criterion. It consists of a number of rewrite-rules that use the hypothesized relations as input and that build a correct RST tree in a bottom-up fashion. The exact rules and their use are not described here since they are rather complex. During the building process, the system checks the potential of each tree in creation in order to prune unlikely trees at each of the three levels (sentence, paragraph and section). Eventually, a number of correct RST trees is produced by the tree building component. Often, trees that are skewed to the right appear to be the desired tree, which is sensible when one assumes that a writer provides the most important information first. This observation is taken into account in the weighing of the trees; right-skewed trees are assigned a greater weight than others. The

tree with the greatest weight is drawn in a PostScript representation.
In order to evaluate the system for automatic RST parsing, two annotators created a gold standard for five texts from the Scientific American, to which the output tree was compared. For the identification of EDUs, for the hierarchy in text spans, for the nuclearity roles and for the relations the labeled precision and recall have been calculated by lining up both trees (see Table 2.3).

Table 2.3: Precision and Recall reached by Marcu's (2000) discourse parser

	annotators		system	
	Recall	Precision	Recall	Precision
EDUs	87.9%	87.9%	51.2%	95.9%
spans	89.6%	89.6%	63.5%	87.7%
nuclearity	79.4%	88.2%	50.6%	85.1%
relations	83.4%	83.4%	47.0%	78.4%

The precision and recall of the annotators is taken to be an upper bound for the performance of the system. The evaluation shows that for each of the four aspects tested, the recall is very low, while the precision is near the inter-annotator agreement. A confusion matrix shows that relations that are detected quite successfully are Contrast, Condition, Example, Otherwise, Purpose, Concession and Cause. These relations are easier to detect than others due to their close relationship with cue phrases. Unambiguous discourse markers lead to success at every text level (sentence, paragraph and section). The relations Elaboration, Background and Joint are more difficult to find, which is probably caused by the fact that they often concern relations at higher levels in the tree. Within paragraphs and short texts, automatic discourse parsing is effective. This can be explained by the fact that in these fragments, the first EDU most of the times is the Nucleus, which is elaborated on by following EDUs. In more complex paragraphs, a writer has inserted discourse markers. Most problematic are long texts, especially newspaper texts since they often mention all facts in the introduction while elaborating on them in various following paragraphs. The assumption of right-skewedness does not apply in such texts. Since the focus of this master thesis is on finding relations within paragraphs, this has no effect for the present task.

2.6 DAS (LeThanh 2004)

Yet another approach is presented by LeThanh (2004) in her PhD dissertation. The difference with previous approaches is that she had the RST Discourse Treebank (Carlson et al. 2003) at her disposal. The system, also referred to as Discourse Analyzing System (DAS), consists of three main parts: (1) discourse segmentation and finding relations between EDU pairs or triplets, (2) labeling the rhetorical relations between the EDUs, and (3) constructing rhetorical structures at the sentence and text level (i.e. combining the local analyses to form a tree for each sentence and using these to build a discourse tree representing the whole text). The system makes use of 22 relation groups: List, Sequence, Condition, Otherwise, Hypothetical, Antithesis, Contrast, Concession,

Cause, Result, Cause - Result, Purpose, Solutionhood, Circumstance, Manner, Means, Interpretation, Evaluation, Summary, Elaboration, Explanation and Joint.

The segmentation procedure employed in the first component is based on the segmentation principals as proposed by Carlson et al. (2003). From these principles a list of Penn Treebank syntactic chains that cover one EDU is derived. If a string found in the text matches to such a syntactic chain, an EDU boundary is inserted at the beginning and end of the string. During the segmentation process, relations and nuclearity roles are also traced if possible. For example, if a segmentation rule describes a noun phrase and its subordinate clause, a relation is created between the two EDUs, and the noun phrase is regarded the Nucleus and the subordinate clause the Satellite. The types of relations are added in the next system component.

To recognize the rhetorical relations, factors are exploited that have already been used in previous research (cue phrases, syntactic information, time references, reiterative devices, reference words, substitution words and ellipses), but also new factors, namely noun phrase (NP) cues (e.g. *the result*) and verb phrase (VP) cues (e.g. *to mean*). In order to find these new factors, NPs and VPs are stemmed and compared to a list of NP and VP cues. This list also contains information about to which EDU the NP or VP cue belongs, which relations can be indicated by the NP or VP cue, and what the likelihood is of the relation with that NP or VP cue. All aforementioned recognition factors are applied in DAS in the following order: syntactic information - cue phrases - NP cues - VP cues - remaining factors. To reduce the processing time, cue phrases are only checked when there can be no relation found on the basis of syntactic information solely, NP cues only when cue phrases do not suffice, etc. Next, heuristic scores are determined on the basis of the type of recognition factor that is present, and on the likelihood that it points at the relation concerned. For example, for cue phrases, the heuristic score is 100, for NP and VP cues it is 90. The scores are based on human intuitions, but when more RST data is available, they could be based on actual occurrences in the data. A relation is added to the tree if its heuristic score is higher than a manually determined threshold and than all other relations, and it satisfies a number of criteria based on Corston-Oliver (1998). These criteria prevent the selection of undesired relations following the heuristic rules. If the threshold is not passed, but there is a signal that the two text spans concerned have a semantic relation, the relation Elaboration is selected. Otherwise, DAS chooses Joint as relation. Eventually, each EDU is connected in the discourse annotation, including its nuclearity role and relation.

In the last system component, all sentences are formed into discourse trees, and eventually the total tree for the whole text is constructed. At this moment the text consists of EDUs that are connected in pairs or sometimes in triplets, but there are no connections between those EDU-groups yet. LeThanh follows Marcu's (2000) Compositionality Criterion, except in some cases. The criterion states that a relation between two text spans either holds between two larger text units or between the most important EDUs within these spans. According to LeThanh, her data indicates that in cases where one of the two related text spans has the internal nuclearity structure Satellite - Nucleus, the Satellite in this text span influences the type of relation between the larger text spans. In these Satellite - Nucleus instances, DAS looks for cue phrases in the Satellite that have not been used to determine the relation between the Satellite and Nucleus within the text span. If there is such a cue phrase, the relation is derived from this, and if there is not, DAS takes the relation between the Satellite and the Nucleus inside

the first text span.
The sentence-level discourse analyzer has now, for each sentence, constructed a discourse tree on the basis of relations between EDUs. Since syntactic information is not as helpful for discourse analysis between (M-)SDUs as for that within sentences (e.g. which is main clause and which subclause), other sources must be taken into account. The approach chosen for this purpose is similar to that for finding RST relations between EDUs, as described above, though leaving out the syntactic information. As initiated in the discourse analysis at the sentence level, all relations have a heuristic score that indicates how probable the relation is. When providing an adjacent sentence pair with a relation, one can calculate an accumulated score by adding up all scores in the tree so far. The system will try to connect two text spans at the same text level before moving to others. The system adds the most promising relations to the list of subtrees. Paths that have a score that is too low compared to other paths are saved separately so that the already created relations can be taken from them in a later stage, preventing the need to create them once again. If the list of subtrees contains only one tree including all sentences in the text, it is added to a set of trees. The process is repeated for each promising path as long as the desired maximum number of output trees is not reached and the list of possible relations is not empty. In the end, a number of possible discourse trees are presented.
LeThanh (2004) proposes an evaluation method, in which precision, recall and F-score are calculated. Precision here represents the percentage of all system outputs that match human annotations, while the recall shows the percentage of all instances in the corpus that the system analysis matches human annotation. She estimated inter-annotator agreement in order to establish a ceiling for the performance of DAS. To compare different systems, LeThanh divides the F-score obtained by the system by the inter-annotator agreement. The F-score reached by the complete DAS system using 22 relations is 38.1%, which leads to a score of 72.3% when inter-annotator agreement is taken into account. LeThanh also evaluated the output of various stages in the pipelined system. The scores obtained by DAS show that it functions well within sentences, but there are many differences between the system output and the human annotations between (M-)SDUs. No error analysis is presented despite the fact that the discourse analysis at the text level is quite erroneous. LeThanh compared her system for finding relations at all text levels (including those between (M-) SDUs) to the only system known to her that presented accuracy figures on the same task, namely that designed by Marcu (2000). Using the method of taking into account the inter-annotator agreement, LeThanh states that the results for DAS are better than that for Marcu. However, no comparison is made between the number and type of relations applied and no information of statistical significance is provided.

Chapter 3

Potential features

This chapter presents the procedure of finding potentially relevant features for establishing where (between which (Multi-)Sentential Discourse Units) rhetorical relations exist.
First, we select features that are described in the previous chapter and which are useful for answering the research question of this thesis. Features that are employed only for segmenting in EDUs, finding relations between them and similar tasks that are not useful for answering the research question are thus omitted. For this reason, the features used in SPADE (Soricut and Marcu 2003) are not included here.
The second section describes the procedure taken to manually identify potentially relevant features by scrutinizing data from the RST Treebank (Carlson et al. 2003).
In section 3, a summarizing list of all features to be tested in the research is provided.

3.1 In the literature

3.1.1 Corston-Oliver (1998)

RASTA was developed as part of the Microsoft English Grammar (MEG). For this project, Corston-Oliver also developed modules for anaphora resolution and for aspects of handling ellipsis. Anaphora are linguistic references to entities, for example personal pronouns that refer to a proper noun mentioned in the previous sentence. Ellipsis means leaving out a part of the sentence that has already been mentioned in the previous sentence or clause in a similar syntactic structure, e.g. *A: He is a very nice man. B: Yes, he is [a very nice man].* Not surprisingly, **anaphora** and **ellipsis** are used as features by the discourse analyzer. We will have to attempt to obtain similar tools in order to be able to use these features as well. Two phenomena can hinder anaphora resolution: deixis and referential continuity. Deixis is present when the meaning of words or anaphora depends completely on the context, for example the personal pronoun *I* that refers to the person who uses it, and therefore changes in meaning at every turn in the conversation. In referential continuity, an entity is referred to by several anaphoric elements in different sentences or clauses. Finding the antecedent is more complicated in such situations since the list of possible antecedents is enlarged with every step one retraces in the text. Both phenomena are very common in newspaper texts, making them useful to consider in the light of this thesis.
Corston-Oliver (1998) also introduces a number of purely **syntactic aspects** that can help in finding rhetorical relations, or at least reduce the number of possible relations

between two text spans. They are **tense** (e.g. past), **aspect** (e.g. progressive) and **polarity** (whether there is negation or not). Subsequently, the spans are checked for **lexical cues** and **discourse markers** and if present, a heuristic score is added, increasing its potential. Lexical cues will also be included in this thesis.

3.1.2 Marcu (1999)

Marcu (1999) uses a shift-reduce algorithm that finds rhetorical relations. This approach is decision-based and, as in the approach intended for our research, makes use of machine learning algorithms. Most of the features applied by Marcu (1999) depend too severely on the shift-reduce method selected to be generally used, e.g. the previous decision, the number of trees already created, etc.

Some features, however, are suitable for this thesis research. For example, the algorithm has information on which **words** and **POS tags** are in preceding and following text spans. We will also use this surface information in our experiments. More sophisticated features are **WordNet-based measures of word similarity**. These include synonymy, antonymy, meronymy, hyponymy, etc. Also, **lexical similarity (word overlap)** is calculated following the method of Hearst (1997). Word similarity and overlap are indicators of the topic of a text span, and can thus be useful for establishing whether two (M-) SDUs are related. Lastly, the text spans are checked for **discourse markers**. This feature has also been mentioned by Corston-Oliver (1998).

3.1.3 Marcu (2000)

As in 1999, Marcu (2000) uses **lexical similarity** as a feature to discover rhetorical relations. He applies a simple method of counting the number of overlapping words. If it passes a certain threshold, he assumes the text spans concern the same topic, which influences the type or relation and the nuclearity role selected.

The method proposed in Marcu (2000) focusses on **discourse markers**. To fully exploit them, Marcu has traced discourse markers in a number of texts from the Brown corpus, together with a 300 word window. Using this window, he was able to determine various characteristics for each discourse marker, e.g. its orthographic environment (e.g. punctuation), its functional role (either sentential, discourse or pragmatic) and its position within the EDU or text span. Since this corpus of discourse markers is not publicly available and creating a similar corpus is beyond the scope of this thesis, we will have to use other resources to find relevant discourse markers (see below). The use of discourse markers appears sufficient for the detection of simple relations at the sentence level in most cases (Marcu 2000). Marcu was not able to find a relation between discourse markers that signify relations between EDUs and those that signify relations between (M-)SDUs or paragraphs. This could indicate that the use of discourse markers is not as useful for our present research as it is for finding relations within sentences. Still, discourse markers are strong rhetorical cues which we therefore should not ignore.

Marcu (2000) states that one can expect **right-skewedness** in a text, which means that a writer introduces a topic and elaborates on it in the rest of the paragraph, thereby creating a right-skewed discourse tree. The introduction of a new topic in the same paragraph seems highly unlikely except when explicitly marked by discourse markers or otherwise.

3.1.4 LeThanh (2004)

In her system DAS, LeThanh (2004) exploits features that have already been used in previous research (**cue phrases**, **time references**, **WordNet-based measures of word similarity**, **anaphora** and **ellipsis**), but also new factors, namely **noun phrase (NP) cues** (e.g. *the result*) and **verb phrase (VP) cues** (e.g. *to mean*). A list of cue phrases is presented in LeThanh's (2004) PhD-thesis. It also includes information on the possible positions of a cue phrase in a span (beginning, middle, end, all positions), the span to which the cue phrase should belong (left span, right span, any side) and the effective scope of a cue phrase (clause, sentence, paragraph). This list of cue phrases can also be used in this thesis and thus solves the problem mentioned above. Syntactic information is only applied to find relations between EDUs. In addition to the list of cue phrases, LeThanh (2004) has created a list of NP and VP cues that is also included in her PhD-thesis, enabling us to use this feature as well. Finding lexical relations is achieved by checking cohesive devices (WordNet-based measures of word similarity, deixis and ellipses). The DAS component that finds relations between (M-)SDUs applies the aforementioned recognition factors in the following order: cue phrases - NP cues - VP cues - remaining factors. To reduce the processing time, NP cues are only checked when no relation can be found on the basis of cue phrases solely, VP cues only when NP cues do not suffice, etc. In our approach, we use all features and try to establish which are most important. The results can perhaps be used in the future to create an order similar to that of LeThanh (2004).

3.1.5 Timmerman (2007)

In his master thesis, Timmerman (2007) explains which features he used for his system for the automatic RST annotation of Dutch medical texts. He checks the spans for **conjunctive adverbs** and **pronouns**, consults **thesauri such as WordNet** and calculates the **word overlap of key words** (nouns). Problematic is the fact that Timmerman's (2007) system is developed for Dutch, meaning that his list of adverbs, for example, consists only of Dutch words. Since researching the difference between Dutch and English rhetorical structure is not the topic of the present study and we want to avoid complications, we will not use the words proposed by Timmerman. The features that are not language specific (pronouns, WordNet-based measures of word similarity and word overlap) can be applied in our research, but have already been proposed by other researchers.

3.2 In the data

In the previous section, potentially relevant features have been presented that were found in the literature on various existing systems for automatic RST parsing. Since the goal of this thesis is more specific than the goals of the systems mentioned, not all discussed systems have been developed for English and the data sets used in the experiments may differ from ours, it is necessary to try and find new possibly relevant features in our data. This section describes how this has been performed and which features have been discovered.

3.2.1 Data sample

In order to find features in the data to be applied for machine learning in this thesis, a sample of the relations in the RST Treebank (Carlson et al. 2003) has been manually evaluated. Considering all relations between text spans in the RST Treebank is not possible due to time restrictions. Moreover, it is important to know whether the features found are useful in a broader context than in that in which they were discovered.

The rhetorical relations that are relevant for the present study are those between two (M-) SDUs within the same paragraph. Our data set therefore consists of all these binary relations that can be found in the RST Treebank (Carlson et al. 2003). They can be subdivided in four different types: (1) a relation between two SDUs (two sentences), (2) a relation between an SDU and an M-SDU (a text span consisting of more than one sentence), (3) a relation between an M-SDU and an SDU, and (4) a relation between two M-SDUs. A Perl script has been written to find (M-)SDUs and their relations in the RST Treebank.[1] First, we extracted the text from the discourse trees and used the EDU boundaries and punctuation to establish which EDU boundaries are sentence boundaries as well.[2] Also, we saved the paragraph boundaries (marked <*P*> in the RST Treebank). With the help of the unique numbers of the first and last EDU of each sentence and paragraph, we had to determine which of all possible sentences and sentence groups within paragraphs are (M-)SDUs in the tree, i.e. which sentences and sentence groups are a node in the discourse tree. Sentences and sentence groups that do not match this criterion are not included in our research, because they do not represent the desired discourse structure and are therefore not suitable to learn from. For each found (M-)SDU, we check whether it holds a binary relation with another (M-)SDU in the same paragraph. Only these relations have been counted in order to determine the data subset for the manual detection of potential features.

We guessed that in order to obtain a clear view of the data and thus to find potentially relevant features, at least 200 relations should be manually evaluated (see Table 3.1).

Table 3.1: Types of relation in data sample

	all data	*sample*
SDU, SDU	1914	144
SDU, M-SDU	552	41
M-SDU, SDU	352	26
M-SDU, M-SDU	106	8
Total	2924	219

To enable future testing on the same texts as Soricut and Marcu (2003) did with their system SPADE, we excluded these texts from those considered to find features. Only texts that consist of at least 5 sentences are taken into consideration since short texts are likely to contain only a small number of text spans and therefore also a small number

[1] The script is later extended to also find the machine learning cases, as described in Chapter 5.

[2] We here assume that EDUs are never longer than one sentence, and that a sentence always contains whole EDUs. There were some sentencing problems, e.g. when the first EDU ends in ., *!* or *?*, and the next EDU starts with a name or abbreviation (capital but not a new sentence), e.g. [*Source: (11) Fulton Prebon (12) (U.S.A.) (13)*] [*Inc. (14)*]. These instances have been manually corrected.

of relations. Such short texts are not uncommon in newspapers. Of the 300 texts that meet these restrictions, 30 texts were randomly selected. For each type of relation shown in Table 3.1, relations were randomly selected from the 30 texts. A list of the reference names of the 30 texts, supplemented with information on the number of paragraphs, sentences, (M-)SDUs and relations can be found in Appendix B.1, while B.2 shows the number of sample relations taken from each text.

3.2.2 Procedure

Manually analyzing the data is necessary because previously developed systems for automatic RST annotation apply different data for different purposes. Since we expect that machine learning algorithms will be able to identify relevant features in the large set of surface features we can easily find (words, POS-tags, etc.), and have already derived numerous features from the literature, the main goal here is to establish what gives us – as a human being – the impression a rhetorical relations exists between two (M-)SDUs. At this point we ignore the fact that some information (especially that concerning world knowledge) cannot be obtained automatically. The chosen approach has serious consequences for the treatment of the features found. Since it is rather subjective, features that are present might be overlooked in one relation when other features are more obvious, while they are found in others where features are more scarce. Determining frequencies of occurrence and other statistics is therefore useless.

3.2.3 Features found

References

Other researchers have already argued for the importance of anaphora and similar phenomena, which are all categorized as *references* here. The most obvious reference type is the use of **personal pronouns**, e.g.

> *Mr. Rogers spent half his cash on hand Friday for "our favorite stocks that have fallen apart."(8) He expects to invest the rest if the market weakens further.(9)* (wsj_2381)

This example also includes another form of reference, namely applying adverbs such as *further*. It implies that the fact that the market is weakening has been mentioned previously in the text. Certain adjectives (e.g. *other*, *next* and *additional*) can have a similar function:

> *Mobil alluded to the work-force cuts last week when it took a $40 million charge as part of its third-quarter earnings and attributed it to a restructuring.(7) Mobil officials said that it is unlikely any additional charges related to this move will be taken in future quarters.(8)* (wsj_0688)

In this example, *additional* is a clue that other *charges* have been discussed previously. Apparently, some words can be used as **reference words**.
Other words that are used to refer to previously mentioned entities are **determiners** like *each/every*, *both* and *some/most of*:

> *For the six months ended Sept. 30, Daiwa reported unconsolidated (parent company) net income of 79.03 billion yen ($556.5 million) on revenue of 332.38 billion yen ($2.34 billion).(11) Both figures were record highs.(12)* (wsj_1303)

Field offices at New Orleans; Houston; Denver; Midland, Tex.; Bakersfield, Calif.; Oklahoma City; and Liberal, Kan., will be maintained.(14) But the staffs at some of those locations will be slashed while at others the work force will be increased.(15) (wsj_0688)

Other potentially relevant features are wh-determiners such as *which* or *what*. They are subject to ambiguity because they can be wh-determiners as well as wh-pronouns (often used between two clauses in the same sentence). This means the part-of-speech tag of these words has to be taken into account.

Plunge?(1) What plunge?(2) (wsj_2391)

Another very common reference method is using **demonstrative pronouns**, e.g.

The property claims service division of the American Insurance Services Group estimated insured losses from the earthquake at $960 million.(2) This estimate doesn't include claims under workers' compensation, life, health disability and liability insurance and damage to infrastructure such as bridges, highways and public buildings. (wsj_0648)

In the example, the demonstrative pronoun *this* clearly refers to the estimation mentioned in the previous sentence. Demonstrative pronouns are also used in time adverbials such as *this month*. These instances should not be considered reference words, since they do not refer to a previously mentioned time unit but to the assumedly known time of writing.
Less ambiguous is the use of **definite articles**:

Separately, a $400 million issue of Fannie Mae Remic mortgage securities is being offered in 15 classes by Bear, Stearns & Co.(23) The offering, Series 1989-86, is backed by Fannie Mae 9% securities.(24) (wsj_1312)

If the *offering* in sentence 24 would not have been introduced in the previous sentence, the article would have been indefinite (*an*). The definite article can also be combined with numerals, in a genitive construction or otherwise:

Non-executive directors of Qintex Australia, who must approve payments to the senior executives, balked at the amount.(32) Two of the directors resigned, Mr. Skase said, so the payments haven't yet been approved.(33) (wsj_1372)

Mr. Uhr said that Mr. Petrie or his company have been accumulating Deb Shops stock for several years, each time issuing a similar regulatory statement.(8) He said no discussions currently are taking place between the two companies.(9) (wsj_0656)

The reference methods described above all concern replacing (e.g. with personal pronouns) or emphasizing (e.g. with demonstrative pronouns, definite articles, etc.) a previously mentioned entity. However, the lack of certain elements in a noun phrase can also indicate the item has been discussed earlier in the text. We have found two variants, namely the exclusion of noun modifiers and of head nouns, which we from now on will refer to as **NP simplification**. Examples are provided below.

When consumers have so many choices, brand loyalty is much harder to maintain.(35) The Wall Street Journal's "American Way of Buying" survey found that 53% of today's car buyers tend to switch brands. For the survey, Peter D. Hart Research Associates and the Roper Organization each asked about 2,000 U.S. consumers about their buying habits.(36/37) (wsj_1377)

Grimm counted 16 transactions valued at $1 billion or more in the latest period, twice as many as a year earlier.(4) The largest was the $12 billion merger creating Bristol-Myers Squibb Co.(5) (wsj_0645)

Lexical cues

The use of lexical cues is a very common method for writers to express their rhetorical goals. Not unexpectedly, therefore, we conclude the same from our data analysis. Although lexical cues such as **cue phrases** and **discourse markers** are ambiguous in the sense that they can always signal a relation within a sentence and between sentences (Timmerman 2007), the data demonstrates it is useful in many cases, e.g.

U.S. exports to Canada jumped 11.2% in August from July while U.S. imports from Canada rose only 2.7%.(2) As a result, Canada's trade surplus with the U.S. narrowed to C$656.5 million (US$558 million) in August from C$1.23 billion (US$1.04 billion) in July.(3) (wsj_1988)

What also became evident from the data is that the use of similar or contrasting verbs or nouns in two (M-)SDUs can be a predictor of a rhetorical relation. An example with two similar and two contrasting verbs can be found below:

The FDA has said it presented evidence it uncovered to the company indicating that Bolar substituted the brand-name product for its own to gain government approval to sell generic versions of Macrodantin.(9) Bolar has denied that it switched the brand-name product for its own in such testing.(10) (wsj_2382)

The verb *to substitute* is a synonym of *to switch*, while the relation between *to present evidence* and *to deny* needs a deeper lexical analysis. For humans, however, the similarity is quite obvious and certainly provides us with a cue for the presence of a rhetorical relation. Following Marcu (1999), we will refer to this as **word similarity**.
A different type of lexical cue is the use of **time references**. It could be argued that this is syntactic information (see later on in this section) since it also determines the structure of a sentence. We believe the semantic meaning of time references is more important than its influence on syntactic structure and therefore it is presented here. Moreover, time references can be placed at various places (though at a limited number) in the sentence. In the example below, the structures of both sentences are obviously similar due to the fact that the time reference is located at the beginning of the sentence, but their meaning actually triggers the expectation for a rhetorical relation.

Until recently, Adobe had a lock on the market for image software, but last month Apple, Adobe's biggest customer, and Microsoft rebelled.(22) Now the two firms are collaborating on an alternative to Adobe's approach, and analysts say they are likely to carry IBM, the biggest seller of personal computers, along with them.(23) (wsj_2365)

Continuous punctuation

Within sentences, punctuation is helpful in establishing which clauses or other units are rhetorically related. Since most punctuation is used to end or split sentences, it is not helpful in finding relations between spans consisting of at least one sentence. Exceptions are **quotation marks**, **dashes** and **brackets** since they can be applied over several

sentences. When an opening bracket is in one sentence and the corresponding closing bracket in another, for example, the sentences are probably related. This is also what we found in the data:

> *"Japanese stock salesmen selling American bonds?(26) Maybe it's crazy," he said.(27)* (wsj_1303)

> *("A turban," she specifies, "though it wasn't the time for that 14 years ago.(5) But I loved turbans."(6))* (wsj_1367)

The first example shows a quotation consisting of more than one sentence, which is very common in newspaper articles. Often, however, the quotation is not presented as a whole, but a clause is inserted, as in the second example where *she specifies* has been added. In this example, the second quotation part still consists of more than one sentence, and moreover, both sentences are between (the same) brackets. Direct speech is fairly frequent in the Wall Street Journal texts from the RST Treebank, but constructions as presented in the examples above are rather special. Much more frequent is direct speech that has been distributed over several quotations, or indirect speech where quotations have been (partly) rephrased, for instance:

> *"There's more noise out there, and the consumer may have to work harder to cut through it," says Vincent P. Barabba, executive director of market research and planning at General Motors Corp.(18) "But the reward is that there's less need to make tradeoffs" in choosing one's wheels.(19)* (wsj_1377)

> *"By default," the dollar probably will be able to hold up pretty well in coming days, says Francoise Soares-Kemp, a foreign-exchange adviser at Credit Suisse.(3) "We're close to the bottom" of the near-term ranges, she contends.(4)* (wsj_0693)

Following these examples, we conclude that the presence of quotations in two adjacent (M-)SDUs is sufficient to lead us to expect that they are rhetorically related.

Syntactic structure

Due to the fact that the RST Treebank and consequently our data consists of articles from the Wall Street Journal, (M-)SDUs are often very similar, especially when financial figures are concerned. An example is provided below:

> *In the first nine months, 1,977 transactions were announced, up 15% from 1,716 in the year-earlier period.(6) Transactions in which prices were disclosed totaled $188.1 billion, up 15% from $163.2 billion a year earlier.(7)* (wsj_0645)

Examples such as these seem to show that **syntactic similarity** is an indicator of rhetorical relations. It is difficult to be certain it is the syntactic structure that is beneficial since the syntactically similar (M-)SDUs are often also lexically similar. The following example presents sentences that are syntactically but only to a lesser extent lexically similar:

> *For instance, employment in Denver will be reduced to 105 from 430.(16) But on the West Coast, where profitable oil production is more likely than in the midcontinent region, the Bakersfield, Calif., office staff of 130 will grow by 175 to 305.(17)* (wsj_0688)

Table 3.2: Example of syntactic similarity in wsj_0688

	SDU 16	SDU 17
adverb	-	*But*
PP	*For instance*	*on the West Coast, where profitable oil production is more likely than in the mid-continent region, the Bakersfield, Calif.*
NP subject	*employment in Denver*	*office staff of 130*
modal	*will*	*will*
lexical verb	*be reduced*	*grow*
PP	*to 105*	*by 175*
PP	*from 430*	*to 305*

The syntactic similarity is obvious, as Table 3.2 shows. Of course the presence of the cue phrase *but* is an even more clear clue in this example.
Because we feel syntactic information is helpful in our data, and also because other researchers have argued for its benefit as well, we decide to include syntactic information in our feature set.

Meta-information

A last category of potentially relevant information for the detection of rhetorical relations is that of meta-information. In the discussion of *continuous punctuation* and *syntactic structure* above we have already touched upon the phenomenon that journalists write in a typical **newspaper style**, especially in domain-specific newspapers like the Wall Street Journal. When reading such an article, we use this knowledge to form expectations about the meaning of the text and the intentions of the author. In the example below, for instance, the quotation in the second sentence is not surprising.

> *Wyse Technology, for instance, is considered a candidate to sell its troubled operation.(42) "Wyse has done well establishing a distribution business, but they haven't delivered products that sell," said Kimball Brown, an analyst at Prudential-Bache Securities.(43)* (wsj_2365)

A possible explanation is that the statement that *Wyse Technology* is a candidate for something needs a reason why this is the case. Especially because of the introduction of the firm in the first sentence (indicated by *for instance*), a quotation from a spokesperson or expert is a likely extension in the following sentence. Certainly there are other ways to continue the text in the first sentence, but in situations as these our expectations concerning the rather standard newspaper style are useful in noting the presence of rhetorical relations.
In other relations, there are no referential, lexical or syntactic cues, nor can conclusions be drawn from punctuation or newspaper style. In these instances, **world knowledge** is needed:

> *On the morning of the crash, he had been put on notice that an audit committee was recommending his dismissal because of invoicing irregularities in a company audit.(20) Investigators have been trying to determine whether the crash was an accident, sabotage or suicide.(21)* (wsj_0619)

In the example, we need world knowledge to become suspicious when a person dies in a crash while he has been dismissed that morning. Since human beings have this knowledge, we expect an investigation to discover whether the crash was an accident or happened on purpose.
Apparently, meta-information is sometimes required to understand two (M-)SDUs are rhetorically related. However, this information is not always available and is especially difficult to obtain automatically. An attempt to solve the storing and consulting of world knowledge is far beyond the scope of this thesis. Because of this, meta-information will be ignored in this thesis from now on.

3.3 Final list of features

We now present a list of all features found in the literature and the data that we plan to apply in machine learning experiments. The feature **length of (M-)SDU** is added here. It could well be that relatively short (M-)SDUs are typically related to the previous (M-)SDU, or to the next. The feature can be easily extracted from the data. In order to include the notion of **right-skewedness** we have included the features **position in the text** and **internal discourse structure**. The feature **ellipsis** (Corston-Oliver 1998) will not be included in our research since we have no tool to solve ellipsis at our disposal, and because we have not encountered instances in the data. A possible explanation is that ellipsis is a characteristic of clauses and is therefore more informative within rather than between sentences.
This is the final list of features to be applied (see Appendix C.1 for an overview including the source(s) of the features):

1. Surface features
   (a) words
   (b) POS tags
   (c) (M-)SDU length
2. Syntactic features
   (a) syntactic similiarity
   (b) tense, aspect and polarity
3. Lexical features
   (a) cue phrases/discourse markers
   (b) NP and VP cues
   (c) word overlap
   (d) word similarity
   (e) time references
4. Reference features
   (a) anaphora resolution
   (b) personal pronouns

(c) definite articles
(d) demonstrative pronouns
(e) reference words (e.g. *further* and *additional*)
(f) (Wh-)determiners (e.g. *which/what* and *both*)
(g) NP simplification

5. Discourse features

(a) position in the text
(b) continuous punctuation
(c) internal discourse structure

Our methods to make the features concrete and to extract their values automatically are described in the following chapter.

Chapter 4

Extracting feature values

This chapter contains information on the methods chosen in order to make the features in the final list from Chapter 3 concrete and to automatically extract the feature values so they can be applied in machine learning algorithms. An overview of the applied resources, tools and Perl scripts can be found in Appendix A. The final list of features with their types (discrete/continuous), their numbers, their possible values, etc. can be found in the table in Appendix C.2.

4.1 Surface features

4.1.1 Words

To establish the informative value of individual words to the detection of rhetorical relations, we create a bag-of-words of the lemmas of all words in the (M-)SDUs in question. We define lemmas as word forms from which inflection has been removed but for which derivation has been kept. Our motivation to use lemmas rather than tokens or stems is that we believe that with lemmatization the word forms represent the full meaning of the original words while preventing word differences caused by the syntactic structure of the sentence. For the purpose of lemmatization we employ the CELEX lexicon (Baayen et al. 1995), and use the Part-of-Speech tags from the Penn Treebank.

4.1.2 POS tags

Since the RST Treebank is part of the Wall Street Journal, manually annotated Part-of-Speech tags are available (POS-files). Checking the POS-tags in the part of the Penn Treebank that has been used for the RST Treebank is beyond the scope of this thesis, so we assume the POS-tags are correctly selected by the human annotators of the Penn Treebank project. Encountered inconsistencies in the Penn Treebank have not been changed in the files themselves, but dealt with in an intermediate step in order to prevent mapping problems. Errors in the RST Treebank, however, have been changed in the data because they are so obviously incorrect. [1]

[1] Despite the fact that it has been annotated by humans, we have discovered a number of inconsistencies in both the Penn Treebank and the RST Treebank. They are presented here to help other researchers who are working with the RST Treebank in combination with the annotations from the Penn Treebank. There may be more errors, but these are the ones we encountered in the data we have worked with.

In order to include POS-tags in our feature set, we determine the relative frequency of each of the POS-tags (unigrams) in the (M-)SDU. The tag set of the Penn Treebank can be found in Appendix D.1. Furthermore, we check the presence of the trigrams that occur at least thrice in the paragraphs in our data set. The trigrams consist of three slots which can be filled with either the POS-tag or the word. This means that there are 8 different types of trigrams for each group of three words. We include all within-sentence trigrams. Sentence boundaries are defined as full stops, question marks, exclamation marks or paragraph endings. Full stops in abbreviations can easily be excluded from this group since they are not tagged as punctuation marks.

4.1.3 (M-)SDU length

In order to establish the length of an (M-) SDU, the number of sentences and the number of words in the (M-)SDU is determined.

4.2 Syntactic features

To find the values of the syntactic features, we employ the parsed files (PRD-files) and files with both parses and POS-tags (MRG-files) from the Penn Treebank, release 2 (1994). Since the annotations have been manually checked, we can assume they are correct. However, some errors have been found, especially concerning (the position of) quotation marks.[2] These have been manually changed in the Penn Treebank files. This decreases the risk of drawing conclusions on the basis of noisy input.

4.2.1 Syntactic similarity

Whereas the previously described features are fairly concrete and their values can be easily determined automatically, the feature *syntactic similarity* is more problematic. Our initial intention was to apply metrics for parser evaluation such as Parseval (Black et al. 1991) or Leaf-Ancestor Assessment (Sampson et al. 1989). These metrics, however, appeared not to be able to cope with parses of different sentences, since they need the

In the Penn Treebank:
wsj_1331: *BUNDY'S/NNP* instead of *BUNDY 'S* (missing space)
wsj_1367: *– that/DT turban/NN* instead of *– that turban –* (missing dashes)
wsj_1376: *We've/NN* instead of *We 've* (missing space)
wsj_1387: *it/PRP s/VBZ* instead of *it 's* (missing apostrophe)
In the RST Treebank:
wsj_1974, EDUs 39-40: *5/ 16* instead of *5/16*
wsj_1367, EDUs 183-186: *translatorfor* instead of *translator for*

[2]During the process of extracting the syntactic trees, a number of errors has been found in the annotations of the Penn Treebank (release 2, 1994) again. In the parsed files (PRD-files) and the files with both parses and POS-tags (MRG-files), closing quotation marks are missing in the trees when they occur at the end of the text. This is the case in wsj_0601, wsj_0604, wsj_0617, wsj_0633, wsj_0635, wsj_0654, wsj_0666, wsj_1110, wsj_1136, wsj_1150, wsj_1159, wsj_1178, wsj_1179, wsj_1306, wsj_1316, wsj_1366, wsj_1367, wsj_1368, wsj_1375, wsj_1376, wsj_1379, wsj_1396, wsj_2303, wsj_2321, wsj_2347, wsj_2357, wsj_2359, wsj_2375, wsj_2381, wsj_2386 and wsj_2393. In wsj_0629, closing quotation marks were represented as opening quotation marks. In wsj_0632, wsj_1125, wsj_1128 and wsj_2303, the opening quotation marks of a sentence were incorrectly included in the annotation of the previous sentence. Similarly, the closing quotation marks of sentences in wsj_1105, wsj_1158 and wsj_2386 were incorrectly placed in the following sentence.

words (tree leafs) in their parse comparison. Next, we considered more recent methods for the determination of syntactic similarity, e.g. *document fingerprinting* (Bernstein and Zobel 2005). The problem with document fingerprinting is that it is designed to determine the syntactic similarity between two full texts, not between two text spans that are maximally one paragraph long. The procedure is to determine the number of chunks – with a fixed size of for instance eight words – that two texts have in common. The size of the chunks must be long enough so they reflect the syntactic structure and short enough not to be too infrequent to enable comparison between texts. Since (M-)SDUs are relatively short, it is not possible to meet both restrictions.
Because we were unable to find an existing method for the determination of syntactic similarity between (M-)SDUs, we had to develop a method ourselves. Consider the following example:

> ((S_1 (NP-SBJ-1 (NP *The 77-year-old official*) , ($SBAR_2$ (WHNP-2 *who*)(S_3 (NP-SBJ *T*-2) (VP *oversaw* (NP (NP *the building*) (PP *of* (NP *the Berlin Wall*)))))) ,) (VP *was* (VP *removed* (NP *-1) (PP-TMP *during* (NP (NP *a meeting*) (PP *of* (NP *the 163-member Communist Party Central Committee*)) (PP-LOC *in* (NP *East Berlin*)))))) .)) (wsj_1924:15/17)

Since the SDU in the example consists of more than one clause, it is difficult to find one single syntactic representation. To solve this problem, we compare each clause in this (M-)SDU with each clause in a second one. The clauses in the example above have been numbered for the reader's convenience. In order to represent the syntactic structure of the clauses in a way that enables comparison to other clauses, we search the categories of the syntactic elements at the highest levels, and of all verb phrase complements. The *clause-level categories* that can be found in our example are presented in Table 4.1.

Table 4.1: Clause-level categories for wsj_1924:15/17

S_1:	NP-SBJ-c VP PP-TMP-c
$SBAR_2$:	WHNP-s S
S_3:	VP NP-c

When looking at the categories in detail, one discovers that syntactic elements that only contain *traces* of other elements have been removed. A trace is an empty element in the syntactic tree which marks the initial position of an element before it has been moved.[3] In the Penn Treebank, traces are indicated by the asterisk (*). For example, consider the VP complement *NP *-1* of *removed* in the example sentence. The trace marks the initial position of the NP *The 77-year-old official*, which has been moved to the beginning of the sentence because of the passive voice. At the place where the VP complement of *to remove* is normally expected, only a trace remains. The explanation for our choice to ignore traces is that we want to compare the surface syntax of two (M-)SDUs, not the underlying syntactic structure.

[3]In his famous work, Chomsky (1957) introduced the idea that language consists of a deep structure and a surface structure. The deep structure represents the underlying grammatical structure, while the surface structure is how we produce or hear it. In the transformation from the deeper structure to the surface form, syntactic elements can be moved. At the initial position only a trace (an empty element) remains.

The letters *c* and *s* in lower case in Table 4.1 indicate the NP or PP concerned is either *complex* or *simple*. Complex NPs (either in a PP or solely) consist of more than one NP, while simple NPs do not (see Table 4.2 for all possible forms of simple NPs and PPs).

Table 4.2: Possible forms of simple NPs and PPs

NP-s	(NP *noun phrase*)
	(WHNP *noun phrase*)
	(NP (NP *noun phrase*))
	(NP (WHNP *noun phrase*))
	(WHNP (NP *noun phrase*))
	(WHNP (WHNP *noun phrase*))
PP-s	(PP (NP-s))
	(WHPP (NP-s))

If a certain phrase is in a parenthetical environment (e.g. between brackets), the category of the phrase is preceded by PRN- in the category sequence of the clause.
We now use the category sequence of each clause and apply the Levenshtein minimal edit distance to compare it to clauses in the adjacent (M-)SDU. Insertion and deletion of categories have a cost of 1, while the cost of category substitution depends on the categories involved and has a value between 0 and 1. In the case of substitution, a distinction is made between the *head category*, a possible *PRN level* and, if present, *details* and *complexity* (definitions are presented below). The substitution costs of all possible head category pairs have been manually set on the basis of our own intuitions about similarity. Most substitutions have a cost of 1, though for identical head categories no substitution is required and the corresponding cost is therefore 0. The head category pairs for which the substitution is established as less than 1 – and of which the head categories are not identical – can be found in Appendix D.2.
For equivalent categories, which we here define as those having no substitution cost, more information is compared. If one category is in a parenthetical unit while the other is not, the cost is increased with 0.25. For NPs and PPs, the cost is increased with 0.25 as well if the complexity differs. If the detail(s) of the one category do(es) not match those/that in the other, if present, 0.1 is added to the cost. In the example used above, details are *SBJ* (subject) in *NP-SBJ-*c_1 and *TMP* (temporal) in *PP-TMP-*c_1. The absence or presence of details has no influence on the complexity of NPs and PPs.
After the Levenshtein minimal edit distance of two clauses has been determined with help of the category-dependent substitution costs, it is normalized by dividing it by the number of categories in the clause with the longest category sequence. We then change it to a positive score by subtracting it from 1:

$$syntsim_{c1,c2} = 1 - \frac{levdist_{c1,c2}}{cat_{max}}$$

in which $syntsim_{c1,c2}$ is the syntactic similarity between clause 1 and clause 2, $levdist_{c1,c2}$ is the Levenshtein distance between the two clauses and cat_{max} is the number of categories in the clause with the most categories. Since it is more easy to find similar clauses

in two long (M-)SDUs, we normalize the final score by dividing the score by the number of clauses in the (M-)SDU with the most clauses:

$$syntsim_{M1,M2} = \frac{syntsim_{best}}{cl_{max}}$$

in which $syntsim_{M1,M2}$ is the syntactic similarity between (M-)SDU 1 and (M-)SDU 2, $syntsim_{best}$ is the highest syntactic similarity possible between two clause pairs from the two (M-)SDUs, and cl_{max} is the number of clauses in the (M-)SDU with the most clauses. Now consider the example mentioned above and imagine three possible adjacent (M-)SDUs with which the syntactic similarity should be determined (see Table 4.3).

Table 4.3: Compare clauses in wsj_1924:15/17 with those in three fictitious (M-)SDUs

1924:15/17	*example1*	*example2*	*example3*
NP-SBJ-c VP PP-TMP-c	NP-SBJ-c VP PP-TMP-c	NP-SBJ-c VP PP-TMP-c	VP NP-c
WHNP-s S	WHNP-s S	VP PP-c	VP NP-c
VP NP-c	VP NP-c		VP NP-c

Obviously, the syntactic structure of *example1* is exactly the same as our original SDU *1924:15/17*. The highest score is thus 1. However, when this number is normalized by dividing it by the number of clauses in the (M-)SDU with the most clauses – which is 3 in both cases – the end score reached would be only 0.33. Would both (M-)SDUs consist of only one clause, the end score would be 1. The same end score is found for *example2*, while one clause is missing in this (M-)SDU. Equally undesirable is the fact that the similarity between *1924:15/17* and *example3* would be considered equal to that with *example1*, while there is only one overlapping clause instead of three. These observations lead us to conclude that the presented method does not reflect the syntactic similarity. We solve this problem in a rather simplistic way. For both (M-)SDUs under consideration, we focus on all clause pairs that reach the highest score. In Tables 4.4, 4.5 and 4.6, all found scores are presented.

Table 4.4: Number of best scores reached in *1924:15/17* and *example1*

		example1		
		NP-SBJ-c VP PP-TMP-c	WHNP-s S	VP NP-c
1924:15/17	NP-SBJ-c VP PP-TMP-c	1	0.25	0.33
	WHNP-s S	0.25	1	0
	VP NP-c	0.33	0	1

The best scores reached are 1 in all three cases. In Table 4.4 (*example1*), the highest score is present in three different clauses in *1924:15/17* and in three different clauses in *example1*. Table 4.5 (*example2*) shows a different distribution because the highest score is found in only one unique clause in both *1924:15/17* and *example2*. In the final example (Table 4.6), the highest score can be found thrice, but all concern the same clause in *1924:15/17*. We now count the number of pairs that do not overlap – are not in

Table 4.5: Number of best scores reached in *1924:15/17* and *example2*

		example2	
		NP-SBJ-c VP PP-TMP-c	VP PP-c
1924:15/17	NP-SBJ-c VP PP-TMP-c	1	0.63
	WHNP-s S	0.25	0
	VP NP-c	0.33	0.5

Table 4.6: Number of best scores reached in *1924:15/17* and *example3*

		example3		
		VP NP-c	VP NP-c	VP NP-c
1924:15/17	NP-SBJ-c VP PP-TMP-c	0.33	0.33	0.33
	WHNP-s S	0	0	0
	VP NP-c	1	1	1

the same column or row in the table – and multiply the highest score with this number before dividing it by the maximum number of clauses. For *example1*, the final score is (3*1)/3 = 1, for *example2* and *example3* it is (1*1)/3 = 0.33. This final normalized score is the figure that represents the syntactic similarity of two (M-)SDUs following our method. The higher the score, the higher the syntactic similarity.

4.2.2 Tense, aspect and polarity

In English grammar, *tense* and *aspect* can take various forms conveying various meanings. Tense can be *present*, *past*, *future* or *future-in-past*, and aspect *simple*, *progressive*, *perfect* and *perfect-progressive*. The Penn Treebank provides no explicit information on the tense and aspect of a verb phrase. We solve this by applying a very simplistic alternative that makes use of the POS-tags for verbs (see Table 4.7).

Table 4.7: Description of the POS-tags for verbs

MD	Modal
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present

In order to use information on tense and polarity, we extracted the POS-tags of all verbs in an (M-)SDU, and counted the number of modals (*MD*), infinitives (*VB*), gerunds (*VBG*), past forms (*VBN*) and present forms (*VPB* and *VBZ*). The numbers of finite verbs (modals, past and present forms) are divided by the total number of finite verbs in

the (M-)SDU. The numbers of infinitives and gerunds are divided by the total number of verbs of any form. Although we realize this is perhaps not the most optimal representation of tense and aspect, we provide the machine learning algorithms with these normalized counts.

Polarity (the presence of negation) is automatically determined by checking the POS-tags of the (M-)SDU for the adverbs (*RB*) *not*, *n't*[4] and *neither*, and the conjunctions (*CC*) *neither* and *nor*. Instances in which *neither* is a determiner are not included, since they have no influence on the syntactic meaning of the sentence, only on the lexical meaning of the NP in which it is present. When an (M-) SDU contains both *neither* and *nor*, they are considered one single negative element. In order to obtain a relative score again, the number of negative elements is divided by the number of present and past verb forms. Since each finite verb can only be negated once – at least in standard English texts such as the Wall Street Journal articles; double negation is possible in some variants of English – we believe this relative score is a good measure of polarity.

4.3 Lexical features

4.3.1 Cue phrases/Discourse markers

The previous chapter has described how many existing systems for automatic RST annotation use the presence or absence of certain cue phrases or discourse markers. It also concluded that the list of cue phrases developed by LeThanh (2004) can be employed for this purpose. The list presents cue phrases for each separate relation and includes scores indicating the certainty that the presence of the cue phrases indicates this specific relation. Since we are not interested in the type of relation, the scores are not considered in our approach. Also, information is given on the effective scope of the cue phrase (clause, sentence or paragraph). Only the relations with a scope of at least one sentence (sentence or paragraph) are included in our eventual list of cue phrases. Lastly, LeThanh (2004) has included information on the desired position of the cue phrase in the (M-)SDU (beginning, middle, end or any position) and in the relation (in the left, the right or any (M-)SDU). We also applied these constraints in our research. The final list of cue phrases applied here can be found in Appendix D.3.[5]

4.3.2 NP and VP cues

In her PhD dissertation, LeThanh (2004) introduces noun phrase (NP) cues (e.g. *the result*) and verb phrase (VP) cues (e.g. *to mean*). All NP and VP cues are listed in LeThanh's PhD dissertation, again including scores that indicated the certainty that it signals a certain relation. The scores are not included in the present research since determining the type of relation is not part of the thesis. For the list of NP and VP cues applied here, the reader is referred to Appendix D.4.[6]

[4] In the Penn Treebank, contracted negation (*n't*) is removed from the verb to which it is attached, and labeled separately with POS tag *RB*.

[5] Of the 207 cue phrases, only 21 were found in our data.

[6] Of the 41 NP cues, 20 were found in our data. Of the 56 VP cues, 43 were found.

4.3.3 Word overlap

To determine the word overlap between two (M-)SDUs, we follow Marcu's (2000) simple method of counting the relative number of overlapping words. We apply this to the tokens, the lemmas and the stems. The CELEX lexicon (Baayen et al. 1995) is employed for the purpose of finding lemmas and stems. In some cases, finding the ultimate stem is impossible due to circular references. The lemma *fysio*, for instance, has a stem *fysiotherapist*, which has a stem *fysiotherapy* which again has a stem *fysio*. In these cases, the stem that is encountered for the second time is saved (*fysio* in this example). For lemmas consisting of more than one stem (e.g. in compounds), we select the first stem.

4.3.4 Word similarity

Next to the word overlap as discussed above, we want to include more sophisticated measures of overlap by taking word similarity into account. We apply two different exiting methods for this goal: the Dependency Based Thesaurus developed by Lin (1998), and Banerjee and Pedersen's (2003) Extended Gloss Overlap.
Lin (1998) applied his broad-coverage dependency-based parser to find dependency trees in a 64-million-word corpus consisting of newspaper articles, including the Wall Street Journal. He then considered all *dependency triples*, which are defined as two adjacent words and their dependency relation. All triples are grouped by their first word. All words that occur at least 100 times in the corpus are included in the thesaurus. The *Dependency-Based Similarity* is calculated by considering the probability distributions and mutual information of the words in the triples concerned. For the exact method the reader is referred to Lin (1998). The result is an automatically constructed and publicly available thesaurus, which for each word lists up to 200 most similar words and their similarities. It enables us to look up words in two (M-)SDUs and determine the corresponding similarity. If a word is not one of the 200 most similar words, we set the similarity for this word pair to 0. For each word pair possible, the similarity is established. The total score is averaged over all word pairs of which at least one word is present in the Thesaurus. The Dependency-Based Thesaurus includes verbs, nouns, adverbs and adjectives. The final score is our first measure of word similarity.
The metric described by Banerjee and Pedersen (2003) employs the manually constructed thesaurus WordNet (Fellbaum 1998). WordNet consists of so-called *synsets* that include all synonyms of a certain concept, together with a gloss and possibly examples of its use. Moreover, WordNet shows relations such as *hypernyms*, *hyponyms*, *meronyms*, *holonyms* and *troponyms*, but also *attribute*, *similar-to* and *also-see* relations. The glosses can be used to calculate the *gloss overlap* (as introduced by Lesk (1998)), being the number of shared words in the glosses of words. While Lesk applied his metric for the purpose of word sense disambiguation by considering the glosses of a word and surrounding words, Banerjee and Pedersen determine the gloss overlap of the synsets of two words. They also include the synsets related to them by one of the aforementioned relations, resulting in the *Extended Gloss Overlap* measure. The metric has been implemented in Pedersen et al.'s (2004) freely available software package *WordNet::Similarity*. With the help of this tool, we established the Extended Gloss Overlap for words in two (M-)SDUs. Due to the fact that the tool appears only to be working for verbs and nouns, the measure is limited to these categories. The total overlap of all word pairs in both (M-)SDUs in question is averaged resulting in our second measure of word similarity.

4.3.5 Time references

Time references are explicitly coded in the syntactic trees in the Penn Treebank. For each (M-)SDU in the triple, we check whether any of the phrases in Table 4.8 are present. Since we are interested in the semantic relevance of time references, we only check whether a time reference is present, regardless of its form.

Table 4.8: Time references in the Penn Treebank

ADVP-TMP	Adverb Phrase
PP-TMP	Prepositional Phrase
NP-TMP	Noun Phrase
SBAR-TMP	Clause with (possibly empty) subordinating conjunction
NAC-TMP	Not a Constituent (shows scope of certain prenominal modifiers within an NP)

4.4 Reference features

4.4.1 Anaphora resolution

Reference information is ideally found by applying an *anaphora resolver*. Anaphora refer back to a noun phrase mentioned earlier in the text (the so-called *antecedent*). As Poesio and Alexandrov-Kabadjov (2004) state in their article, many language technological applications need a module for anaphora resolution and therefore a number of such systems has been developed, but none of them can be easily obtained and used by other researchers. This is in contrast with other natural language processing systems such as parsers and Part-of-Speech taggers. Poesio and Alexandrov-Kabadjov (2004) developed an *off-the-shelf* anaphora resolution module which they named *GuiTAR* (*General Tool for Anaphora Resolution*). GuiTAR consists of several steps that can either be followed through or replaced by alternative steps. The anaphora resolver in GuiTAR needs syntactic information. The user can provide the system with raw text input, which is then automatically parsed, e.g. with the Charniak parser (Charniak 2000). Since the Penn Treebank syntactic trees are similar to Charniak output, we skip the first step and replaced it by the MRG-files (which are tagged parses) from the Penn Treebank.[7] Using the tags, a second module determines the Minimum Anaphoric Syntax (MAS), representing the minimal information needed by the anaphora resolver. With help of the tags in the syntactic parses, the module marks all nominal expressions by adding the code *ne* and a unique identification number. Moreover, information on the category (e.g. *the-NP*), person, gender and number is coded, all in an XML-format. On the basis of this input, the anaphora resolution module in GuiTAR builds a discourse model. It employs various existing methods for the resolution of pronouns (Mitkov et al. 2002) and definite descriptions (Vieira and Poesio 2000). Proper nouns and demonstratives are not included, at least not in the version we have obtained. The output of GuiTAR is the same as the input except that at the end of the text, a summary is given that indicates

[7] We encountered the same problems as with the PRD-files in the Penn Treebank, and solved them in a similar way.

which nominal expressions refer to which nominal expressions earlier in the text. When the antecedent of a nominal expression also has an antecedent, GuiTAR selects the antecedent that is present earliest in the text. It therefore attempts to solve referential continuity. Since it includes an algorithm for pronoun resolution, it is expected to be able to solve deixis as well.
Anaphoric relations between two (M-)SDUs can be traced thanks to the unique identification numbers of the nominal expressions. We offer the Penn Treebank syntactic trees of the full Wall Street Journal text in question to the MAS-module and subsequently to GuiTAR. In the Penn Treebank files, we add codes in order to be able to find the (M-)SDU boundaries back later on. This has no influence on the performance of GuiTAR since it ignores elements without a tag. We assume two (M-)SDUs are anaphorically related when the antecedent of a nominal expression in the second (M-)SDU is present in the first (M-)SDU, or when both (M-)SDUs contain a nominal expression and they have the same antecedent.
Because GuiTAR does not treat all types of anaphora and because an automatic system is bound to produce erroneous output, as a precaution we also include more simple and less error-prone features. They are described below.

4.4.2 Personal pronouns

We count the number of personal pronouns in each (M-)SDU and divide it by the total number of words in the (M-)SDU to established a relative number. They can be found easily because they have a separate POS tag in the Penn Treebank (*PRP*) Since it is most likely for a pronoun in the first clause of the (M-)SDU to refer to a previous (M-)SDU, we also check whether a personal pronoun is either present or absent in the subject or any of the VP complements of the first clause.

4.4.3 Definite articles

For the definite article (*the*), we determine the values of the same features as for personal pronouns: the relative number of definite articles and the presence or absence of it in the subject and the VP complements of the first clause of the (M-)SDU.

4.4.4 Demonstrative pronouns

Demonstrative pronouns (*this*, *that*, *these* and *those*) are treated in the same way as personal pronouns and definite articles.

4.4.5 Reference words

In the manual analysis of the data, we found that certain words can have a referential meaning, e.g. *further*, *other*, *additional*, etc. The full list of reference words applied can be found in Appendix D.5. The list is based on the words found in the data selection used for the search for features, and is supplemented with synonyms taken from the thesaurus of Microsoft Word 2003[8]. For each of the reference words in the list, we determine whether it is present or absent in the second (M-)SDU of a relation.[9]

[8]The English thesaurus of Microsoft Word 2003 was developed for Microsoft by Bloomsbury Publishing.

[9]Of the 31 reference adverbs and adjectives in the list, 28 were found in our data.

4.4.6 (Wh-)Determiners

Besides definite articles and demonstrative pronouns, more determiners can have a referential role, namely *all*, *both*, *each*, *either*, *every*, *some*, *what* and *which*. Again we establish the relative number of these (wh-)determiners in the whole (M-) SDU and check whether they are present or absent in the subject and the VP complements of its first clause.

4.4.7 NP simplification

In Chapter 3, we defined two types of NP simplification: the exclusion of noun modifiers and that of head nouns. In the syntactic trees of the Penn Treebank, we trace all NPs in subjects and in VP complements. We then check whether two NPs in two adjacent (M-)SDUs have the same last word, which is often the head word of the NP. Next, all words in the first NP are compared to the words in the second NP. Determiners (tagged *DT* in the Penn Treebank) are excluded, leaving only the words that modify the head noun. When the first NP contains a modifier that is not present in the second, we say NP simplification is present in this (M-)SDU pair.
When a head word is lacking in a noun phrase, it is not possible to map it to noun phrases in an adjacent (M-) SDU. Therefore, we check whether there are occurrences of NPs (in subjects or VP complements) of which the first word is a definite article, and the last word either a numeral, a number, a superlative or a comparative. For the numerals, we include all adjectives (Penn Treebank tag *JJ*) which are either *first*, *second* or *third* or end in *-th*. Numbers are tagged *CD*, superlatives *JJS* and comparatives *JJR*, and are therefore easy to detect. An NP such as *(NP (DT the) (JJS best))* (wsj_0664) would thus be considered to contain NP simplification. The presence of this type is checked for the second (M-)SDU in each pair, because it is expected to refer back to a previous NP.

4.5 Discourse features

4.5.1 Position in the text

Since we want to include information on its position in the text, we save the sentence number in the paragraph and the paragraph number in the text. These numbers represent the order of the text.

4.5.2 Continuous punctuation

The presence of continuous punctuation can be easily determined. For brackets, we check whether the opening and closing brackets are in different (M-)SDUs. If dashes or quotation marks are present in two adjacent (M-)SDUs, we conclude that they possess continuous punctuation.[10]

4.5.3 Internal discourse structure

Ideally, an author should write a text that has a right-skewed discourse tree: he/she starts with the most important information (the Nucleus) and elaborates on it in

[10] In our data, no instance of continuous dashes could be found.

the rest of the paragraph (the SATELLITES)[11]. It might therefore be useful to include information on the internal discourse structure of M-SDUs (i.e. consisting of more than one sentence). In the M-SDU, we look at the highest node and save the number of sentences and the nuclearity of both branches. The result is a simple representation like *1N-2S*. When an (M-)SDU consists of only one sentence, or the top node does not split neatly into sentences, we give it the label *na* (*not applicable*).

For an overview of the features and their characteristics (their type, number, values, etc.), see Appendix C.2.

4.6 The full (M-)SDU or the Nucleus sentence?

As mentioned in Chapter 2, Marcu (2000) argues that relations between larger text spans (extended relations) can often be explained by the relations between their NUCLEUS EDUs (simple relations). For our research, it is important to know on which part of the (M-)SDUs the features should be based. Obviously, considering only the NUCLEUS sentence is more computationally efficient than regarding the whole (M-)SDU. Since the smallest units in our approach are sentences (SDUs), we need to check whether the simple relations are also suitable when they are not based on the NUCLEUS EDUs but on the NUCLEUS sentences.
We therefore determine the values of each of the features mentioned above not only on the whole (M-)SDU but also on the NUCLEUS sentence in it. If it concerns an SDU, the only sentence present is automatically the NUCLEUS sentence. For M-SDUs, the internal rhetorical relations are checked until one NUCLEUS sentence is found. In case the NUCLEUS is not a whole sentence, the sentence to which this NUCLEUS belongs is selected. M-SDUs consisting of multi-Nuclear relations are not included in this part of the research, since it makes selecting one NUCLEUS sentence impossible. The consequence is that only a subset of the data can be used to determine the values of the features for the NUCLEUS sentence.

[11] Although we ignore the nuclearity distribution in this thesis, we include it here because it provides us with information on the structure of the discourse tree.

Chapter 5

Applying machine learning

Now we have established an inventory of potentially relevant information for automatic detection of rhetorical relations between (M-)SDUs, and have developed measures to make it concrete and to extract it automatically, it can be offered to machine learning algorithms. This chapter first describes how we simplify automatic discourse analysis to a decision problem. Next, we describe the data and our methods for discovering which features are relevant for the given task.

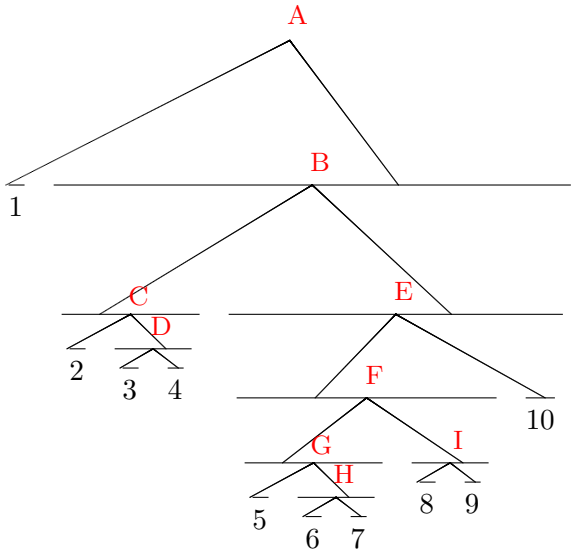
5.1 Simplifying the problem of discourse analysis

To accomplish objective 3, which consists of determining which of the features offered to machine learning algorithms are indeed relevant, we need to simplify the problem of RST discourse analysis. Following Soricut and Marcu (2003), relations are always considered binary in our approach. This is justified because non-binary relations are highly infrequent in the RST Treebank (99% of the relations is binary). Moreover, including non-binary relations in this thesis would undesirably complicate the machine learning problem. An example of a binary RST discourse tree is presented in Figure 5.1. Numbers represent sentences and letters indicate tree nodes. The relations and nuclearity roles are not included in the figure.

When looking at three adjacent text spans, a relation can hold only between two of the spans, since each span can only have one relation with another span. The span in the middle therefore is either connected to the left or the right span. Lets assume the fictitious text that is represented by this tree consists of two paragraphs and a title; sentence 1 is the title, the first paragraph contains sentences 2-4 (node C) and the second sentences 5-10 (node E). We will focus on the second paragraph. Now consider spans 5-7 (node G), 8-9 (node I) and 10. Only between 5-7 (G) and 8-9 (I) a rhetorical relation exists (node F), while there is no relation between spans 8-9 (I) and 10. In other words, branch 8-9 (I) is connected to a text span to its left, not to its right. This decision problem, i.e. the choice between *left* and *right* attachment, is offered to the machine learning algorithms.

In order to collect data representing cases in which such decisions can be made, we need to extract all relations between (M-)SDUs and all adjacent (preceding or following) (M-)SDUs in the same paragraph. This information is taken from the RST Treebank (Carlson et al. 2003). First of all, we extract all text spans in a discourse tree by following the nodes (see Table 5.1 for a list of all text spans in the second paragraph in Figure

Figure 5.1: Simple representation of a fictitious RST tree



5.1). Next, all relations in each paragraph in the text are found in the discourse tree (see Table 5.2).

Table 5.1: Text spans in the second paragraph of example 5.1

M-SDU	*represented in tree:*	*SDU*	*represented in tree:*
5-10	Node E	5	Leaf 5
5-9	Node F	6	Leaf 6
5-7	Node G	7	Leaf 7
6-7	Node H	8	Leaf 8
8-9	Node I	9	Leaf 9
		10	Leaf 10

Table 5.2: All relations in the second paragraph of example 5.1

left (M-)SDU	*right (M-)SDU*
5-9 (F)	10
5-7 (G)	8-9 (I)
5	6-7 (H)
6	7
8	9

For each relation between two (M-)SDUs in the RST Treebank, we then consider all possible adjacent text spans within the same paragraph. For example, for node F in Figure 2 we look at branches 5-7 (G) and 8-9 (I) and all possible preceding and following

text spans. Since the paragraph in focus starts with sentence 5, there are no relevant preceding (M-)SDUs. There is a following (M-)SDU that starts with 10 (which is only 10). All relations and the corresponding preceding and following (M-)SDUs in the second paragraph of Figure 5.1 are presented in Table 5.3.

Table 5.3: Finding triples in the second paragraph of example 5.1

previous	*left*	*right*	*following*
-	5-9	10	-
-	5-7	8-9	10
-	5	6-7	8-9 8
5	6	7	8-9 8
5-7 6-7 7	8	9	10

The combination of the two (M-)SDUs in a rhetorical relation and one preceding *or* following (M-)SDU will from now on be referred to as a *triple*. Each found triple is a *case* in the machine learning problem. The actual relation in a triple is either on the *left* or on the *right*, which is the *class* the machine learning algorithm should select. Table 5.4 shows the triples and correct directions for the second paragraph in the example. The machine learning algorithms will try to predict the class for each triple by applying the features chosen in the first objective.

Table 5.4: Triples in the second paragraph of example 5.1

5-7	8-9	10	left
5	6-7	8-9	left
5	6-7	8	left
5	6	7	right
6	7	8-9	left
6	7	8	left
5-7	8	9	right
6-7	8	9	right
7	8	9	right
8	9	10	left

It is clear that many of the (M-)SDUs in the triples in Table 5.1 overlap. The next section shows how we deal with this.

5.2 Data

Obviously, the triples need to be automatically extracted from the RST Treebank. We extended the Perl script written for the manual detection of potential features (Chapter 3) in such a way that it is able to find the suitable triples as well. As previously mentioned, only relations between sentences and sentence groups that are a node in the RST tree are considered (M-)SDUs, and only binary relations within paragraphs are included in our research. In total, 2136 triples could be extracted. We name this data set *all2136_MSDU* (or short: *all2136*) because it contains all 2136 triples and the feature values are based on the whole (M-)SDUs in it. In order to establish whether the features should indeed be based on the full (M-)SDU, or the information in the NUCLEUS sentence suffices, two extra data sets are created:

1. *sub1866_Nucl*: a subset of 1866 cases for which the feature values are based on the NUCLEUS sentence
2. *sub1866_MSDU*: the same subset of 1866 cases but with features based on the whole (M-) SDU

In these data sets only 1866 triples are present, because it was not possible to select a single NUCLEUS sentence in the other 270 cases. The latter include instances where the M-SDU contains a multi-nuclear relation or the NUCLEUS is not a sentence (e.g. one and a half sentence). SDUs are always present in this data set because they consist of only one sentence, which is therefore always the sentence in focus (the NUCLEUS). The number of triples and their corresponding classes in *all2136* and *sub1866* can be found in Table 5.5. The triples occur in 942 different paragraphs (918 in *sub1866*).

Table 5.5: Number of triples in *all2136* and *sub1866*

	all2136	*sub1866*
left	940	833
right	1196	1033
Total	2136	1866

For each of the triples, the feature values have been automatically derived as proposed in the previous chapter, but manually annotated syntactic trees, POS tags and discourse structures have been employed. Some feature values needed to be found for all three (M-)SDUs (e.g. the presence of NP cues), which is indicated by a feature prefix $l\_$ (*left*), $m\_$ (*middle*) or $r\_$ (*right*). Others needed to be established for the two (M-)SDU pairs that potentially form a relation (e.g. word overlap), indicated by the prefix $L\_$ (*Left*) or $R\_$ (*Right*).
In machine learning, a subset of the data is offered to algorithms that try to discover patterns (the *training set*). To evaluate how well the learned structures generalize to unseen data, the found model can be applied to new data (the *test set*). Due to the rather small number of triples that can be found in the RST Treebank, we decide to apply ten-fold cross-validation on all cases. This means that we divide the data into ten separate *partitions*, of which we train on nine and test on one. This is repeated ten times and the number of correctly classified instances is added up each time. The

total number of correctly classified cases is then divided by the total number of cases (2136). It would not be fair to place some cases of a Wall Street Journal text in the train data and other cases of the same text in the test data, because many triples concern the same relation but with different preceding or following (M-)SDUs (as in Table 5.4, where three triples contain the relation between SDU 8 and 9). Therefore we could not use Orange's build-in ten-fold cross-validation but had to manually split the data into partitions with equal numbers of triples and of texts. A consequence of this approach is that the partitions are not stratified (i.e. the distribution of the classes *left* and *right* differs). The division in partitions for *all2136* and *sub1866* can be found in Appendix E.
The number of *surface* features is so large (over 18,000) that it is too computationally expensive to use all of them in the machine learning algorithms. For this reason we select the 1000 most useful surface features in the training set for each partition with the help of Relief (Kononenko 1994, see description in Section 5.3.1). In this thesis all feature sets that should include *all* surface features only contain these *1000 best* surface features.
Now the data can be offered to machine learning algorithms in order to establish which features are most useful.

5.3 Finding relevant features

In order to find structure in the large set of features we have found, we need machine learning algorithms. Since creating such algorithms is far beyond the scope of this thesis, we deceided to experiment with existing implementations. Since it is not always entirely clear to what extent these systems are suited to handle our type of data, interpreting the results should thus be done with caution.
Obtaining the implementations is much less problematic. Various machine learning interfaces that include algorithms and pre- or post-processing options have been designed, one of which is Orange (Demsar et al. 2004). The tool is freely available on the internet and comes with a graphic shell. It is also possible to write Python scripts that start the machine learning experiments, which is especially useful when one wants to apply different algorithms, data and feature sets as is the case in the present research. For these reasons, we employ the Orange package.
This section describes how the relevant features are discovered in three different ways: (1) by applying algorithms that establish the benefit of individual features (*feature relevance*), (2) by considering the models developed by various classification algorithms, and (3) by comparing the classification scores reached when offering the different feature types (surface, syntactic, lexical, reference and discourse) separately and by leaving them out.

5.3.1 Feature relevance

We apply two algorithms that try to establish which features are most beneficial for the task of choosing between *left* and *right*. They are *Relief* (Kira and Rendell 1992) and *Cluster Separation Score* (*CSS*, van Halteren personal communication).
Relief randomly selects a data instance and considers two types of nearest neighbours, namely one of the same class (the *nearest hit*) and one of a different class (the *nearest miss*). The feature relevance is not based on the distribution of all values of that particular feature, as is common in other methods that try to establish the power of certain

features such as *information gain*, but by considering the distribution of the feature values in three similar data instances (the randomly selected case, the nearest hit and the nearest miss). Because all features and their values are considered in order to find the nearest hit and miss, it could be argued that Relief does not assume features are independent in the way other approaches do. The algorithm of Relief is the following:

```
set all weights W[A] := 0.0;
for i := 1 to m do
    begin
        randomly select an instance R;
        find nearest hit H and nearest miss M;
        for A := 1 to all_attributes do
            W[A] := W[A] - diff(A,R,H)/m + diff(A,R,M)/m;
    end;
```

In this algorithm, m indicates the number of data instances to be randomly considered [1]. *Diff{Attribute,Instance1,Instance2}* calculates the difference between the values of the feature *Attribute* for two instances, one of which is the randomly selected case and one is the nearest hit or miss. For discrete features, the difference is either 1 (the values are the same) or 0 (the values differ). The difference of continuous features is the actual difference but normalized to a range of 0-1. According to Relief, a feature with great predictive power is thus a feature that has equal (discrete) or similar (continuous) values in the same class, but different values in other classes. The found weight ($W[A]$) is what we from now on will refer to as the *relevance score according to Relief.*
Kononenko (1994) has extended original Relief, for example by enabling the treatment of machine learning tasks concerning more than two classes and by extending the algorithm to k-nearest neighbours search. Orange includes an implementation of Kononenko's Relief with a default number of nearest neighbours (k) of 5. Relief can be employed to decrease the number of features to be offered to the machine learning algorithms by demanding a minimum Relief score or selecting the top scores. Furthermore, the scores can be saved and printed, which provides us with a measure of feature relevance.
In addition to the feature selection mechanism Relief, we apply a formula developed by van Halteren (personal communication): *Cluster Separation Score* (*CSS*). For each feature A, the separation score is determined in the following manner:

$$CSS_A = \frac{\bar{x}_{left} - \bar{x}_{right}}{\sigma_{left} + \sigma_{right}}$$

The mean value ($\bar{x}$) and the standard deviation (σ)[2] are determined for both classes *left* and *right*. The resulting *relevance score according to CSS* is an indication of the extent to which the feature is able to distinguish the cases with class *left* from those with class *right*.
Because CSS determines the average value and the standard deviation, it expects the feature values to be continuous. This is highly problematic in our data set since only a

[1] At this stage, we decided to apply the default settings of Orange. Later in the research, we discovered that the default number of examples considered is only 50. In Chapter 6, we find that for instance the surface features receive low relevance scores for Relief. The low number of examples considered could explain this, since low-frequent features are then easily missed.

[2] Although the feature values are not necessarily normally distributed, CSS considers the found standard deviation acceptable as an approximation of the actual deviation.

fraction of the features is continuous. The great majority consists of discrete (nominal) features that should thus be converted into numerical features to enable the use of CSS in this thesis. All discrete features except *internal discourse structure* concern the absence or presence of a word, a phrase, or something else. They are adapted by changing the values *absent* and *present* to *0* and *1*. The question arises whether the standard deviation is meaningful in this context, and can function as the basis of a feature ranking. When a fraction L of the feature values in cases with class *left* are 1, and a fraction R of the values in cases with class *right* are 1, the corresponding CSS can be approximated by (given the large number of cases):

$$CSS = \frac{L - R}{\sqrt{L \times (1 - L)} + \sqrt{R \times (1 - R)}}$$

This means that the CSS increases when (1) the difference between L and R increases, and (2) when L and/or R become closer to 0 or 1. These two are exactly the elements that express how typical the presence or absence of a feature is for the class, and thus show how relevant the feature is for predicting the class. There may be many other formulas that function equally well, but probably differ in the weight assigned to the two elements. Despite the fact that CSS was originally designed for Gaussian data, we expect it to be suitable as a ranking instrument for 0/1-features as well.
The feature for discourse structure is more problematic, since it has values such as *na* (*not applicable*) and *N1-S2*, denoting the internal discourse structure of an (M-) SDU. In order to make it accessible for CSS, it is split into two features, one for the nuclearity, and one for the number of sentences. The nuclearity is given either value -1, 0 or 1 when only the left (internal) (M-)SDU is a NUCLEUS, when both (internal) (M-)SDUs are NUCLEI and when only the right (internal) (M-)SDU is a NUCLEUS, respectively. The number of sentences in both internal (M-) SDUs is represented by a relative score, namely the log of the number of sentences in the left internal (M-)SDU divided by the number sentences in the right internal (M-) SDU. The new representations of internal discourse structure are not ideal but necessary for the application of CSS.
We provide both CSS and the Orange implementation of Relief with all feature values for the cases in each of the 10 training sets in *all2136_MSDU*. Both methods provide relevance scores for each feature in each of the 10 training sets. In order to retrieve a single score for each feature following each of the two algorithms, we calculate the average of the scores in the separate partitions.
Despite the fact that continuous and discrete features are quite different in form and in the way we treat them, we have not distinguished them in establishing the relevance scores. The effect of inter-type ranking should thus be considered in future research. The 50 most relevant features in both algorithms are presented in the following chapter.

5.3.2 Classification algorithms

As mentioned at the beginning of this section, we depend on existing implementations of classification algorithms. Those applied in the present thesis are only briefly introduced here. Assuming the conversion of nominal data to ordinal data is warranted, we see no reason to doubt their suitability for the present task.
We will apply five machine learning algorithms: *Naive Bayes*, *k-Nearest Neighbours* (*kNN*), *Support Vector Machines* (*SVM*), *Decision Trees* and *Maximum Entropy*. The

first four are present in the Orange software and we therefore employ those implementations. Since establishing the optimal parameters for each algorithm and each partition in the data would be an undesirably long process and there is probably not enough data to do it properly, we decide to apply the algorithms with the default settings of Orange. For Maximum Entropy we use the implementation of Zhang (2004).
Naive Bayes is a probabilistic classifier that employs Bayes' theorem. It assumes all features are independent and is therefore *naive*. The kNN algorithm classifies cases on the basis of the k closest training examples in the feature space. It stores all training data and uses it only at the actual classification. We use the default implementation in Orange, in which k is the square root of the number of cases. SVM tries to establish hyperplanes in a kernel-defined transformation of the multi-dimensional space of all features and their values in the training set. The hyperplanes are then used to classify the test cases. Orange includes the libsvm-8.1 library for SVM (Chang and Lin 2001), and uses *RBF* (*Radial Basis Function*) as the default kernel. The Decision Trees algorithm creates trees that are easy to understand for humans. The highest node in the tree shows the feature that is most helpful in splitting up the data. Each branch again splits into new branches according to the values of the next-best feature, etc. Maximum Entropy finds a formula that describes the probability of the classes best. All values of each feature are considered separate features and they are assigned weights. The formula is then used to classify the cases on the basis of their feature values. In Zhang's implementation of Maximum Entropy, the weights are found by employing the *Limited-Memory Variable Metric* (Benson and More 2001), which Malouf (2002) demonstrates performs best in Natural Language Processing tasks with large numbers of features.
kNN, SVM and Decision Trees are able to deal with continuous features, but Naive Bayes is not. We therefore make them discrete with the 'discretization'-function in Orange. The range of each continuous feature in the training set is divided into seven equal-frequency intervals, which are given a unique name on the basis of the range they represent. The 'discretization' of each training set is applied to the corresponding test set. Maximum Entropy is able to handle continuous features, but it treats every feature-value combination as a separate feature, thus considering every present value of a continuous feature separately. With the large number of features we present it, the algorithm will design a model that is very difficult to analyze. We therefore offer Maximum Entropy the same discrete versions of the features as Naive Bayes. Again we ignore the fact that the algorithms may treat continuous features (or features made discrete in seven intervals) and discrete features differently. It seems advisable to investigate this issue in future research.
Since the machine learning task concerns choosing between only two classes (*left* and *right*), and the distribution of both classes is known, the machine learning results are best represented by the *accuracy*, being the number of correctly classified cases in the test set divided by the total number of cases in the test set. The accuracies are compared with a baseline of selecting the most frequent class, which is *right* (56.0%). Our baseline thus assumes right-skewedness.
If the algorithms perform significantly better than the baseline, their models are analyzed to find relevance scores for each feature in a way similar to Relief and CSS (if possible). Again a list of the 50 best features according to the algorithm is provided. We then create a ranked list of the features that are present in the 'top 50' of each feature selection algorithm and machine learning model of sections 5.3.1 and 5.3.2. They are ranked on the basis of the ranks in the separate lists.

5.3.3 Feature types

In order to establish which of the feature types (surface, syntactic, lexical, reference and discourse) are beneficial in the current machine learning task, we create 11 different feature sets for *all2136_MSDU* (Table 5.6). The feature sets are offered to all five machine learning algorithms: Naive Bayes, SVM, kNN, Decision Trees and Maximum Entropy.

Table 5.6: Different feature sets

all_features	all feature groups are included
surface_only	only the surface features are included (Section 4.1)
syntactic_only	only the syntactic features are included (Section 4.2)
lexical_only	only the lexical features are included (Section 4.3)
reference_only	only the reference features are included (Section 4.4)
discourse_only	only the discourse features are included (Section 4.5)
no_surface	all feature groups are included except the surface features
no_syntactic	all feature groups are included except the syntactic features
no_lexical	all feature groups are included except the lexical features
no_reference	all feature groups are included except the reference features
no_discourse	all feature groups are included except the discourse features

The results found for the different data sets provide us with information on which types of features are most informative when provided in isolation, and which types are beneficial even when all other feature types have already been applied.

The results of the experiments described in this chapter are presented in Chapter 6.

Chapter 6

Machine learning results

Objective 3 of this thesis is to apply machine learning algorithms in order to establish which features are useful in deciding whether a relation holds between two (M-)SDUs. This chapter shows the results of machine learning algorithms and lists which features appear to be relevant. We would like to repeat that we decided to experiment with existing, ready-to-use, machine learning tools since developing a model ourselves is beyond the scope of this thesis.

First of all, we considered the data set *all2136*. In total, 8664 different features have been used: 20 syntactic, 718 lexical, 84 reference, 14 discourse and 7828 different surface features (the 1,000 best for each training set according to Relief). Only 806 are *active* in more than one partition. With features that are active we mean discrete features that (also) have values other than *a* (*absent*) or *na* (*not applicable*) and continuous features that (also) have values other than *0*. Surface features should also be in the list of the best 1000 features (according to Relief) of more than one training-test split.

We have applied two algorithms that try to establish which features are most beneficial for the task of choosing between *left* and *right*. They are *Relief* (Kira and Rendell 1992) and *Cluster Separation Score* (*CSS*, van Halteren personal communication). For both methods, we list the 50 most relevant features.[1]

Moreover, the results of the machine learning algorithms *Naive Bayes*, *k-Nearest Neighbours*, *Support Vector Machines*, *Decision Trees* (all in Orange, Demsar et al. 2004) and *Maximum Entropy* (Zhang 2004) are presented and evaluated. Where possible, we have attempted to deduce information from the machine learning models.[1]

We then show and discuss the results of the separate feature sets in which the feature type (surface, syntactic, lexical, reference and discourse) are considered.

Lastly, the accuracies reached for *sub1866* are presented and discussed.

6.1 Feature relevance algorithms

The 50 most relevant scores according to Relief (Kononenko 1994) can be found in Appendix F.1. All features in this top 50 are present in more than one partition. The top 10 is presented in Table 6.1. If the position in the triple (second column) is written with a capital letter, the feature is determined on an (M-)SDU pair, otherwise it is based on one of the three single (M-)SDUs.

[1]The full lists can be downloaded from `http://lands.let.ru.nl/~daphne`.

Table 6.1: Relief: Top 10 of features

Feature	*Position in triple*	*Feature type*	*Relief score*
pers. pronoun in first clause	Right	reference	0.0401
time reference	middle	lexical	0.0291
to	right	surface	0.0233
and	left	surface	0.0219
past tense	middle	syntactic	0.0210
past tense	right	syntactic	0.0205
def. article in first clause	Right	reference	0.0200
present tense	right	syntactic	0.0200
def. article in first clause	Left	reference	0.0199
present tense	middle	syntactic	0.0196

Similarly, the 50 best scores of all features according to CSS (van Halteren, personal communication) can be found in Appendix F.2. It includes an extra column which indicates which class is best selected when the value of a binary feature (e.g. cue phrases) is *present* or the value of a continuous feature (e.g. token overlap) is high.

Opposed to Relief, this top 50 also contains features that are present in only one partition. Since it is undesirable to draw conclusions on features that occur in only one partition of the data, we create a new list which includes only those features that are present in more than one partition (Appendix F.3). A Top 10 of this list is shown in Table 6.2.

Table 6.2: CSS: Top 10 of features present in more than 1 partition

Feature	*Position in triple*	*Feature type*	*CSS score*	*direction*
pers. pronoun in first clause	Right	reference	0.1469	right
nr of pers. pronouns	Right	reference	0.1331	right
cont. quotation marks	Right	discourse	0.1256	right
word similarity (Lin)	Right	lexical	0.1181	right
PRP (pers. pronoun)	right	surface	0.1168	right
word similarity (Lin)	Left	lexical	0.1125	left
token overlap	Right	lexical	0.1084	right
syntactic similarity	Left	syntactic	0.0973	left
stem overlap	Right	lexical	0.0949	right
lemma overlap	Right	lexical	0.0919	right

Only the first feature in both lists (the presence of a personal pronoun in the right relation in the triple) is present in both top 10s. Apparently the algorithms select different features. In Chapter 7 the features are discussed in detail.

6.2 Classification algorithms

The accuracies reached by the various machine learning algorithms are shown in Table 6.3. The algorithms were provided with all features for data set *all2136_MSDU*. The classification accuracy of Maximum Entropy is based on the results reached with the optimal *Gaussian prior*[2].

Table 6.3: Accuracies reached for *all2136_MSDU*

The significance (compared to the baseline) is indicated by the asterisks:
* p<0.05, ** p<0.01, *** p<0.001

baseline	Naive Bayes	kNN	SVM	DecTrees	MaxEnt
56.0%	60.0%***	51.1%	56.9%	53.1%	60.9%***

One of the main conclusions we can draw from the results is that despite our efforts to include good representations of all potentially relevant information, the accuracies reached by the machine learning algorithms are rather low. Only Naive Bayes and Maximum Entropy performed significantly better than the baseline when all features with their values based on the whole (M-)SDUs were provided. The rather large number of features (1836) and the low number of cases (2136) led us to suspect that the other three algorithms might be suffering from overtraining. Technically speaking, however, k-Nearest Numbers cannot *suffer* from overtraining, since it overtrains per definition, without that being a problem (van den Bosch, personal communication). Support Vector Machines can overtrain, because it is possible that the error decreases during the training process, but at a certain moment cannot decrease in the test set, and may even rise again (van den Bosch, personal communication). This is also possible for Maximum Entropy, though it seems less problematic than for SVM since the accuracy reached is much better.
To check whether the algorithms are indeed affected by the large number of features and the low number of cases, we offered fewer features to them by employing Relief for feature selection, and selecting the best features for each partition. The results can be found in Table 6.4 and Figure 6.1.
As expected, the performance of kNN, SVM and Decision Trees increases when less features are offered, but only SVM ever performs significantly better than the baseline, when provided with the best 100 features[3]. Still, its performance is significantly lower than that of Maximum Entropy with all features[4]. The performance of Naive Bayes and Maximum Entropy decreases when less features are offered, but they perform significantly better than the baseline even when provided with the best 1000[5] or 100[6] features. Maximum Entropy classifies significantly better even with only the best 50[7] or

[2]A feature value that occurs only with a certain class will be considered a good predictor. However, it could be that such a clear distribution is not caused by the predictive quality of the feature but by the variability of the data. The Gaussian prior determines the boundary between variance (coincidence) and information. This is very important in small data sets such as ours, since it helps preventing overtraining.
[3]chi-square 6.17, p<0.05
[4]chi-square: 4.53, p<0.005
[5]chi-square: Naive Bayes 11.26, Maximum Entropy 19.77, p<0.001 for both
[6]chi-square: Naive Bayes 9.31, Maximum Entropy 12.47, p<0.001 for both
[7]chi-square: 13.73, p<0.001

Table 6.4: Accuracies reached for *all2136_MSDU* with features selected by Relief

The significance (compared to the baseline) is indicated by the asterisks:
* p<0.05, ** p<0.01, *** p<0.001

features	baseline	Naive Bayes	kNN	SVM	DecTrees	MaxEnt
all 1836	56.0%	60.0%***	51.1%	56.9%	53.1%	60.9%***
1000 best	56.0%	59.6%***	53.6%	55.9%	51.8%	60.7%***
100 best	56.0%	59.3%**	53.6%	58.7%*	52.6%	59.8%***
50 best	56.0%	57.6%	53.8%	56.9%	55.1%	60.0%***
20 best	56.0%	56.6%	52.1%	55.9%	49.9%	59.9%***
10 best	56.0%	54.9%	51.3%	55.5%	52.3%	57.3%

20[8] features. The other algorithms perform better when offered less features, but worse again when the number becomes too small (for SVM 50, for kNN and Decision Trees 20).
Another explanation for the disappointing results could be that the default settings in Orange are not optimal for the given task and data. The default k in kNN, for example, is the square root of the number of cases in the training set. We expect that a lower k could increase the accuracy reached and thereby the suitability of the system and its model. Experimentally establishing the optimal parameters, however, is beyond the scope of this thesis, but it should certainly be considered in the future. Similarly, a different kernel could be tried for SVM.
Our intuitions are that the disappointing results are not only caused by the small data set, the large number of features, or the parameter settings, but also by the artificiality of the task. By ignoring the relation types and nuclearity, and by rephrasing discourse analysis as the choice between left and right, we have created a task that is never performed in real life. The distance between our method and real discourse analysis is great. A consequence is that we have not been able to determine a ceiling to the accuracy, which could have helped in establishing whether the results are indeed disappointing or perhaps acceptable given the difficulty of the task.
An important consequence of the low accuracies reached by the classification algorithms is that analyzing the models is speculative. There is a great risk that the models found by the algorithms are coincidental rather than marking structures in the data. The two algorithms that do show a significant improvement over the baseline are apparently able to sift the information from the sets of features with some success. Assuming that this sifting is expressed in the model parameters, we attempt to extract an indication of feature importance by

- singling out the N highest weighted features (if something is very important, it must have made it to the top)
- the overall ranking (the ranking may not be exact, but a feature ranked on average above another feature by the four measurements is likely to be more important)

Still, seeing that the top lists differ extensively between the four measurements, we must

[8] chi-square: 13.09, p<0.001

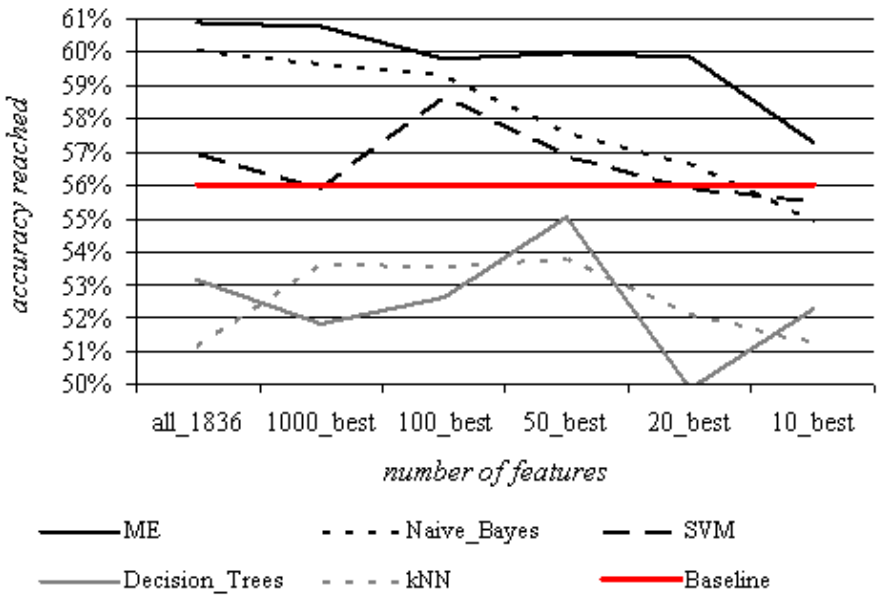


Figure 6.1: Accuracies reached for *all2136_MSDU* with features selected by Relief

advise caution in taking these judgements for granted. Future research, investigating the value of our measurements, seems advisable.
In the next section, we describe how we have established feature relevance scores on the basis of the models of Naive Bayes and Maximum Entropy. As for the systems that are not able to improve over the baseline, they are obviously unable to sift the information and any ranking is not likely to provide a useful measurement of feature importance.

6.3 Finding feature relevance scores for Naive Bayes and Maximum Entropy

The model of Naive Bayes can be summarized as follows:

$$p(C, F_1, \ldots, F_n) = p(C) \prod_{i=1}^{n} p(F_i|C)$$

in which p is the probability, C the class (*left* or *right*), F_i the feature and n the number of features. To find a relevance score for the features following the model of Naive Bayes, we establish the probability of each feature given the class (being $p(F_i|C)$ in the formula). We have approached this by considering both *left* and *right* and counting the number of times a certain feature value occurs with that class, and divide it by the total number of cases with the class:

$$p(FV|C) = \frac{nr(FV|C)}{nr(C)}$$

in which nr means *number of* and FV is the feature value. We then loop through all cases and divide the probability of the feature value given the correct class by the

probability of the feature value given the incorrect class, and take the log. The result is the contribution of the feature value for that particular case:

$$contr(FV|case) = \log \frac{p(FV|C_{found})}{p(FV|C_{opposite})}$$

in which *case* is the triple (machine learning case) that is considered at the moment, C_{found} the class of this case (e.g. *left* and $C_{opposite}$ the opposite class (*right*). We then average the attributions over all cases to achieve a single relevance score for the feature:

$$relevance\ score\ F\ (NB) = \frac{\sum_{case=1}^{c} contr_{FV|case}}{c}$$

in which c is the number of cases.
In Appendix F.4, we present the 50 most relevant features according to the above described relevance score derived from Naive Bayes. All of the features in the top 50 are present in more than one partition. The Top 10 of the features can be found in Table 6.5.

Table 6.5: Naive Bayes: Top 10 of features

Feature	*Position in triple*	*Feature type*	*NB score*
nr pers. pronouns	Right	reference	0.0476
PRP (pers. pronoun)	right	surface	0.0447
pers. pronoun in first clause	Right	reference	0.0428
word similarity (Lin)	Right	lexical	0.0421
word similarity (Lin)	Left	lexical	0.0339
cont. quotation marks	Right	discourse	0.0317
token overlap	Right	lexical	0.0307
internal discourse structure	Right	discourse	0.0257
NNP (proper noun)	right	surface	0.0255
internal discourse structure	Left	discourse	0.0243

Maximum Entropy considers each feature with each value separately and therefore establishes a weight for each feature-value combination. The following function finds an average weight ($\bar{w}(F)$) for the feature F in training set t:

$$\bar{w}(F)_t = \frac{\sum_{V=1}^{v} w(F_V)_t \times nr(F_V)_t}{c_t}$$

in which V is the feature value, v the number of different feature values, $w(F_V)_t$ the weight of the feature-value combination which can be found in the model parameters, $nr(F_V)_t$ the number of occurrences of the feature-value combination in this training set and c_t the number of cases in the training set. The resulting weighted average weight is then averaged over the 10 training sets:

$$relevance\ score\ F\ (ME) = \frac{\sum_{t=1}^{10} \bar{w}(F)_t}{10}$$

This final figure is the relevance score for the feature in question. Features that are not present in the training set of a training-test split are assigned a score 0 for that split. The top 50 of the highest ME relevance scores consists only of features that are present in more than one partition. For this reason only one list can be found in Appendix F.5. The model enables us to establish the class (direction) expected by most features (as was the case with CSS). Table 6.6 shows the Top 10.

Table 6.6: Maximum Entropy: Top 10 of features

Feature	*Position in triple*	*Feature type*	*ME score*	*direction*
cont. quotation marks	Right	discourse	0.3007	right
missing modifier	Right	reference	0.2395	right
pers. pronoun in first clause	Right	reference	0.2179	right
word similarity (Lin)	Right	lexical	0.2107	right
cont. quotation marks	Left	discourse	0.1735	left
added	Right	reference	0.1706	right
other	Right	reference	0.1599	right
nr of dem. pronouns	Right	reference	0.1568	right
nr of determiners	Left	reference	0.1509	left
include (VP)	left	lexical	0.1503	left

The above methods are developed in order to approximate the relevance of the features we have offered to Naive Bayes and Maximum Entropy. Whether our approach is indeed a reflection of the models of the algorithms is not entirely certain. Nonetheless we believe we have established transparent and sensible relevance scores for our features. We repeat that other researchers should be careful in drawing conclusions on the basis of our findings.

6.4 Reaching a final list of relevant features

We have found four different lists with presumably the most relevant features, namely according to Relief, CSS, Naive Bayes and Maximum Entropy. Despite the problems concerning the small data set, the rather artificial task, the obliqueness of the machine learning algorithms and the possibly limited applicability of the algorithms to our type of data, we want to derive a final list of relevant features on the basis of the four algorithms mentioned. We approach this in a rather crude way.
For all four algorithms we have established lists showing the relevance of the 806 features that are active (i.e. having values other than *absent*, *na* or *0*) in more than one partition. In each of these lists, we have assigned scores to the features on the basis of their ranks. The best feature received 1 point, the second 2, etc. Equally ranked features received equal ranking scores. We added up the ranking scores in each of the four methods. Appendix F.6 shows the 50 best features following this ranking procedure. The Top 10 is presented in Table 6.7.
Obviously, a final list as that created would ideally have been based on a combination of the algorithms used, not on the outcome of the separate systems. Combining the

Table 6.7: Ranking in all 4 algorithms: Top 10 of features present in more than 1 partition

Rank	*Feature*	*Position in triple*	*Feature type*
1	pers. pronoun in first clause	Right	reference
2	def. article in first clause	Right	reference
3	cont. quotation marks	Right	discourse
4	past tense	left	syntactic
5	token overlap	Right	lexical
6	PRP (pers. pronoun)	right	surface
7	time reference	right	lexical
8	missing modifier	Right	reference
9	present tense	left	syntactic
10	lemma overlap	Right	lexical

algorithms and analyzing the results, however, is not included in this thesis and thus left for future research.

6.5 Feature types

The previous chapter introduced a number of feature sets that we created in order to establish the benefit of the feature types (surface, syntactic, lexical, reference and discourse) when offered in isolation. Moreover, the features of each type have been removed in order to discover which types are useful when added after the other four features types have already been applied. The results are in Table 6.8.

Table 6.8: Accuracies reached for *all2136_MSDU* with features selected by type

The significance (compared to the baseline) is indicated by the asterisks:
* p<0.05, ** p<0.01, *** p<0.001

features	baseline	Naive Bayes	kNN	SVM	DecTrees	MaxEnt
all_features	56,0%	60.0%***	51.1%	56.9%	53.1%	60.9%***
no_surface	56,0%	59.7%***	53.5%	56.1%	55.2%	60.3%***
no_syntactic	56,0%	60.7%***	51.5%	56.8%	54.3%	61.1%***
no_lexical	56,0%	59.5%**	50.9%	57.6%	53.4%	60.1%***
no_reference	56,0%	58.6%*	52.4%	57.0%	52.7%	60.8%***
no_discourse	56,0%	60.5%***	50.7%	53.4%	53.4%	61.3%***
surface_only	56,0%	58.1%*	52.8%	56.7%	52.2%	57.4%
syntactic_only	56,0%	54.0%	54.3%	56.1%	55.2%	54.5%
lexical_only	56,0%	55.3%	53.7%	54.4%	53.4%	57.0%
reference_only	56,0%	59.4%**	56.8%	56.9%	54.8%	59.7%***
discourse_only	56,0%	55.7%	53.6%	55.7%	54.1%	55.9%

Again, the low accuracies seem to indicate that either the task is too difficult (meaning the features may be useful but not sufficient) or the systems applied are not suitable for

the given task. Because of this problem a detailed analysis is not possible.
Of the feature sets that differ significantly from the baseline, none differs significantly from the *all_features* feature set. This means that surface features (for Naive Bayes) and reference features (for Naive Bayes and Maximum Entropy) solely are equally informative as all features together.
No single feature type is able to improve classification performance if all the other feature types were already used, indicating that the different feature types have a large overlap in information content. For example, the presence of personal pronouns is also a surface feature (POS tag PRP).

6.6 The full (M-)SDU or the Nucleus sentence?

For data sets *sub1866_MSDU* and *sub1866_Nucl*, we have extracted the feature values for the same feature set as with *all2136*, and have offered them to the same machine learning algorithms. The baseline of selecting the most common class (*right*) is 55.4% in these subsets.
The accuracies reached by the algorithms when provided with the two *sub1836* data sets have been presented in Table 6.9.

Table 6.9: Accuracies reached for *sub1866_MSDU* and *sub1866_Nucl* with all features

The significance (compared to the baseline) is indicated by the asterisks:
* p<0.05, ** p<0.01, *** p<0.001

data set	baseline	Naive Bayes	kNN	SVM	DecTrees	MaxEnt
sub1866_MSDU	55.4%	60.2%***	53.4%	56.8%	53.1%	63.3%***
sub1866_Nucl	55.4%	60.5%***	53.9%	60.1%***	51.6%	62.2%***

As could be expected, we encounter the same problems as stated earlier in the chapter: the results are too low to give firm ground that they are meaningful and not a mere coincidence. Nonetheless, the results are mentioned and shortly analyzed below.
The classifiers kNN and Decision Trees perform worse than the baseline. Naive Bayes and Maximum Entropy both classify significantly better than the baseline on *sub1836_MSDU*[9] as well as *sub1836_Nucl*[10], but there are no significant differences between them. This can mean two things. First, it could be that it does not matter whether the features are based on the whole (M-)SDU or only on the Nucleus sentence. Second, it could be that for some cases, the features should be based on the Nucleus sentence, while for others, it is better to derive them from the full (M-)SDU. A detailed error analysis could clarify this issue and should therefore be performed in future research.
Different are the classification accuracies of SVM. When the features values are based on the whole (M-)SDU, it performs only slightly (but not significantly) better than the baseline (56.8% compared to 55.4%). With the features based on the Nucleus sentence the accuracy increases to 60.1%[11]. Following SVM, we would conclude that the features are best based on the Nucleus sentences. Still, it could be that the cases that are

[9] chi-square: 17.96, p<0.001 for Naive Bayes, chi-square: 47.50, p<0.001 for Maximum Entropy
[10] chi-square: 19.99, p<0.001 for Naive Bayes, chi-square: 34.98, p<0.001 for Maximum Entropy
[11] chi-square 16.79, p<0.001

incorrectly classified when the features are based on the NUCLEUS sentences would be classified correctly if they were based on the full (M-)SDU.[12]
It may seem that the results in Table 6.9 are better than those in Table 6.3, but the results are not comparable because the cases in *sub1866_MSDU* and *sub1866_Nucl* differ from those in *all2136_MSDU*.
Although the results of Naive Bayes, Maximum Entropy and SVM give us interesting information about the role of the NUCLEUS sentences in our data and task, it is still not clear on which part of the (M-)SDUs the features should be based. Finding relations on the basis of the NUCLEUS sentences leads to similar or even better results than on the basis of the full (M-)SDUs and is therefore promising. This is desirable because it is more computationally efficient than including the full (M-)SDUs. However, for some cases the feature values might be more informative when based on the whole (M-)SDU. In the future, feature values are thus best extracted for both the NUCLEUS sentence and the whole (M-)SDU in order to prevent the loss of any relevant information.

[12] An error analysis was not possible due to time restrictions.

Chapter 7

Evaluation of the useful features

The previous chapters have described how we have experimented with various existing algorithms in order to establish which of the features found in Chapter 3 appear to be relevant for the present task.
The results have led us to be very critical towards our methodology and to be cautious in drawing conclusions. Of the five classification algorithms we have employed, only two were able to construct models that performed better than the baseline. These were the Orange implementation of Naive Bayes (Demsar et al. 2004) and Zhang's (2004) software for Maximum Entropy. The disappointing results may be caused by the small data set, the large number of features, the parameter settings, the artificiality of the task and/or the extent to which the algorithms are able to deal with the type of information provided. As it seems, the two aforementioned systems have been able to sift the information from the feature sets with some success. Assuming that this sifting is reflected in the model parameters, we have developed methods to derive a single relevance score for each feature on the basis the models of each system. We must advise caution in copying our methods and results since the models of the existing implementations used are not fully transparent, and the suitability of our methods for reaching single relevance scores is uncertain.
In addition to the two classification algorithms mentioned, Relief (Kononenko 1994) and CSS (van Halteren, personal communication) have been used. They are feature scoring algorithms that rank scores according to their degree of informativeness for the given classes *left* and *right*. Again, the suitability of the algorithms is not entirely clear. Relief has not been designed for language processing specifically, and CSS was designed to deal with numerical features. We therefore advise the reader again to be cautious.
Following the estimated relevance scores, the features have been ranked for each of the four algorithms. The rankings have again lead to a final ranked list. Ideally, the final ranking should be based on a combination of the separate systems rather than on their individual outcomes. Seeing that the tops of the ranking lists differ extensively between the four measurements, one should not take the rankings for granted. Also, the fact that the algorithms treat discrete and continuous features differently is ignored in the ranking. Future research, investigating the value of our measurements and the effect of inter-type (discrete/continous) ranking, seems advisable.
Clearly, drawing conclusions from the found feature rankings is only legitimate when done with care. We will still attempt to achieve objective 4, namely to evaluate the useful features, to try to explain why they are useful and possibly to relate them to the literature. This chapter discusses the relevant features found for each of the feature

types. The best 50 features for each machine learning algorithm are discussed, followed by a short analysis of the features in the overall top 50[1]. Readers are again warned that the evaluations may be based on results that are coincidental or obtained in a suboptimal way.

7.1 Surface features

The relevance of surface features depends heavily on the algorithms applied and on the decision to either include all features or only those that occur in more than one partition. For example, of the 50 most important features following Relief, 28 are surface features, compared to only 8 in the model of Maximum Entropy.
All surface features found by Relief but one concern words: *a*, *and*, *as*, *be*, *but*, *for*, *have*, *i*, *in*, *it*, *mr.*, *of*, *'s*, *that*, *the* and *to*. Also, the Part-of-Speech tag *PRP* (personal pronoun) is present in the list. Most surface features are thus grammatical words. This is rather unexpected because at first sight they seem too common to be helpful. The words *but*, *it* and *the* have been mentioned by other researchers as relevant features, and we have also found them in the data sample. For this reason, they have been included in other feature types (lexical and reference). The fact that Relief selected these features from the pile of 1000 surface features may show they have predictive power.
The surface features found by CSS are the words *a*, *equipment*, *everyone*, *it*, *little*, *of*, *the*, *to* and *tractor*. The benefit of some of the words, e.g. *equipment* and *tractor*, is difficult to explain and is therefore expected to be a result of the small data set. Also, the machine learning cases (triples) often overlap because they consider the same rhetorical relation but with a different preceding or following (M-)SDU. When considering only those features that are present in more than one partition, they are indeed absent. The feature *everyone* is also only found in one partition. Both lists include the POS tags *PRP* (personal pronoun) and *NNP* (proper noun). Proper names of persons and companies are very common in financial newspaper articles such as those in the Wall Street Journal. The fact that they are useful in finding rhetorical relations could mean that the paragraphs are written in a certain (newspaper) style. The length of the (M-)SDU in words is also found in the list of features present in more than one partition.
In Naive Bayes, surface features solely are (almost) equally effective (58.1%) as all features together (60.0%). Opposed to the other three algorithms, Naive Bayes includes trigrams (consisting of words and POS tags) in the list of the 50 best features. In total, 19 of the 37 surface features in this list are trigrams. Five of them describe various forms of proper nouns, a feature also found by the other algorithms. Some trigrams in the list are clearly beneficial, e.g. *On_DT_other*, which is part of the cue *On the other hand*[2]. None of the trigrams is present in more than one partition. This is also caused by the fact that to be found in a partition, it should not only be present but should also be one of the 1000 best features according to Relief because of our pre-processing step of reducing the number of features. Naive Bayes also considers many POS tags relevant: CC (coordinating conjunction), CD (cardinal number), DT (determiner), JJR (adjective, comparative), NNP (proper noun), PRP (personal pronoun), VB (verb, base form), VBG (verb, gerund or present participle), VBN (past participle), VBZ (verb, 3rd person singular present) and WP (wh-pronoun). The presence of the tags for certain verb

[1] The top50s can be found in Appendix F
[2] There are no other occurrences of *On DT other* found in the data than those for *On the other hand*.

forms indicates that syntax is important for the detection of rhetorical relations. The benefit of coordinating conjunctions is clear, but one would expect that the type of coordinating conjunction is much more informative than the mere presence. The presence of comparatives and cardinal numbers could be be related to what we have previously defined as NP simplification. The presence or absence of these modifiers can be a way of referring back to a previously used noun phrase. However, they are part of the 1000 best surface features in only one partition. This could mean the feature is less relevant than we would expect. Also, the tags VB and VBN are not in the list with only the features present in more than one partition. New in this list are the POS tags IN (preposition or subordinating conjunction), JJ (adjective) and RB (adverb). Again, the benefit of these tags is clear but one would expect that the type of conjunction, adverb or adjective would be of more importance. According to Naive Bayes, relevant words are *of*, *the* and *wait*, of which the latter is not in the list of features present in more than one partition. As with CSS, the length of the (M-) SDU is also considered useful.

As mentioned at the beginning of this section, the 50 most useful features following the model of Maximum Entropy includes only 8 surface features, being the words *as*, *farmer*, *little*, *new* and *now*, and the POS tags VBZ and NNP (the latter for the right and the middle (M-)SDU in a triple). The presence of the word *farmer* in this top 50 can be explained by the fact that it occurs in (overlapping) cases in 12 texts in our data set, spread over various training-test splits. Maximum Entropy needs more than the surface features to perform optimally (the accuracy reached is 57.4% with only surface features, 60.9% with all features types).

Until now no attention has been paid to the fact that the features always occur in pairs or triples in the feature set; they are determined for each of the three (M-) SDUs or for each of the two (M-)SDU pairs in the machine learning triples. We will now look at the surface features in the *overall top 50* of the most relevant features based on the ranking in the lists derived from all four algorithms, and try to draw conclusions on the features in relation to the position in the triple. Moreover, we will check whether the feature hints at the direction (*left* or *right*) we would expect by looking at CSS and Maximum Entropy. The expected direction means that a feature is either a cue for the presence of the one relation or for the absence of the other.

In Table 7.1, the words that are present in the overall top 50 can be found, together with their position in the triple. If present in the top 50 of CSS and/or Maximum Entropy, a direction is indicated. This direction is that which is best chosen when the feature value is *p* (*present*). The table leads to rather surprising conclusions. For example, when the right-most (M-)SDU contains the word *the*, we expected it to be a reference word, thus indicating the relation should be on the right. However, CSS points at a relation at the left side of the triple. This can perhaps be caused by the relatively high frequency of *the* in (M-)SDUs. Another reason is mentioned in Section 7.4. For *as*, we could expect a relation at the left when it is present in the middle (M-)SDU. It is then used as a cue for a relation with a preceding (M-)SDU. Maximum Entropy shows us the opposite: when *as* occurs in the middle (M-)SDU, the relation the middle (M-)SDU, the relation is most probably on the right. A possible explanation is that *as* is used in various phrases (e.g. *as previously mentioned*, *as long as*, etc.) that have very different meanings. The direction could thus be a result of our data set rather than discourse structure in general. The word *to* causes similar questions. One could expect it functions as a cue (e.g. *to achieve this*), but it could also be part of *have to* or other constructions. As previously mentioned, the undesirable presence of the word *farmer* is a result of the topics of the

texts in our data set. The word *little* in the middle (M-)SDU seems to refer back to the previous (M-)SDU, because the relation is expected on the left by both algorithms. It could thus be similar to the determiner *some (of)* (Section 3.2.3). The word *it* in the list shows the pattern we expected. Pronouns are normally used to refer back to an earlier mentioned entity, and would thus always signal a rhetorical relation with a previous (M-)SDU. The table shows that when *it* occurs in the right-most (M-)SDU, the relation is expected on the right of the triple.

Table 7.1: Surface features in overall top 50: words

words	position	direction CSS	direction ME
a	middle	Right	-
as	middle	-	Right
farmer	right	-	Left
it[a]	right	Right	-
little	middle	Left	Left
the[b]	right	Left	-
to	middle	Right	-

[a]the features for *left* and *middle* are present in the top 100
[b]the feature for *middle* is present in the top 100

The POS tags in the list are presented in Table 7.2, together with the direction that can be expected with high feature values (thus large relative frequencies of the POS tags). In the last line, *unclear* means that the direction varies per frequency range (the seven intervals in the discrete versions of the continuous features) and no general line could be detected. As also remarked earlier in this section, syntax appears to be important, but it is difficult to explain why certain verb forms are useful in certain (M-)SDUs. Personal pronouns are of course very useful cues for rhetorical relations. The expected direction is confirmed by CSS. Proper nouns are important in financial newspaper texts. The table shows that when a proper noun is present in the right-most (M-)SDU, the relation is most likely on the left of the triple. Perhaps a person or company introduces a new topic, making it less probable to have a relation with a previous (M-)SDU. Future research should be directed at the benefit of combinations of features, so we can draw more clear conclusions on what happens here.

Table 7.2: Surface features in overall top 50: POS tags

POS tags	position	direction CSS	direction ME
PRP (personal pronoun)	middle	-	-
PRP (personal pronoun)	right	Right	-
NNP (proper noun)[a]	right	Left	Left
VBZ (verb, 3rd person sing. pres.)[b]	left	-	Unclear

[a]the feature for *middle* is present in the top 100
[b]the feature for *right* is present in the top 100

7.2 Syntactic features

The presence of certain surface features has already shown that syntax is relevant in the detection of rhetorical relations. It is therefore not surprising that all four algorithms have syntactic features in their top 50s. We believe the tense, aspect, polarity and modality of an (M-)SDU is important when it is compared with adjacent (M-)SDUs. Matching verb tenses or forms can be a hint that rhetorical relations are present. Differing verb forms might be equally informative. Since we do not consider the interaction between features in this thesis, no conclusions can be drawn on the matter. This section thus only shortly describes the findings without attempting to explain them.
Relief considers present, past tense and modality important features. The feature syntactic similarity, being the method we developed ourselves, is not present in the top 50 of Relief.
The two top 50s for CSS (one with all features and one only with those features that are present in more than one partition) both contain the same syntactic features: past tense, present tense, gerund forms and syntactic similarity.
For the model of Naive Bayes, the two top 50s differ greatly in syntactic features. Because the list with all features contains such a large number of surface features, only two syntactic features are in the top 50: past tense and syntactic similarity. The number of syntactic features is much larger in the top 50 with only the features present in more than one partition: past tense, present tense, gerund form, infinitive form, polarity (negation) and syntactic similarity. Applying only syntactic (54.0%) features leads to results that do not even exceed the baseline (56.0%).
Maximum Entropy includes past tense, present tense, modality, infinitive form and syntactic similarity in the top 50. Again, with only syntactic features, Maximum Entropy performs worse (54.5%) than the baseline (56.0%).
Apparently the tense of an (M-)SDU is very useful in finding rhetorical relations, since it is an important feature in all four algorithms. Both present and past tense are therefore present in the overall top 50 (see Table 7.3). Also included are modals, gerunds and infinitives.

Table 7.3: Syntactic features in overall top 50: tense, aspect and polarity

verb	position	direction CSS	direction ME
past tense[a]	left	left	unclear
present tense	left	Right	-
present tense	right	-	Right
modal[b]	right	-	-
gerund[c]	middle	Right	-
infinitive[d]	left	-	-
infinitive	middle	-	Unclear

[a] the feature for *right* is present in the top 100
[b] the features for *left* and *middle* are present in the top 100
[c] the feature for *left* is present in the top 100
[d] the feature for *right* is present in the top 100

Table 7.4 shows that syntactic similarity was only in the top 50 for the left pair in the machine learning triple. CSS expects a rhetorical relation on the left when the syntactic

similarity between the left-most and the middle (M-) SDU is high. This is what the literature and our data also suggested. For Maximum Entropy, the direction depends on the similarity range: the expected class varies per interval (in the discrete version of syntactic similarity), and no general line could be found.

Table 7.4: Syntactic features in overall top 50: syntactic similarity

feature	position	direction CSS	direction ME
syntactic similarity	Left	Left	Unclear

7.3 Lexical features

Lexical information has been used by all systems described in Chapter 2, and has also come forward in the data sample we considered.
In the top 50 of Relief, no cue phrases are present. This is rather surprising since it is the only feature that has been applied by all researchers described in this thesis. Also, neither word overlap nor word similarity are part of the group of best features. Instead, only time references and the verb *to make* are relevant enough to be in the top 50. The benefit of time references is clear because they denote the (often chronological) structure of the text. Perhaps the feature is even more useful when considered for all (M-)SDUs simultaneously; two adjacent (M-)SDUs with a time reference are probably rhetorically related.
According to CSS, many lexical features are important, including word overlap (for tokens, lemmas as well as stems) and word similarity (from Lin's (1998) Dependency Thesaurus). Time references are also considered useful. Moreover, two of the cue phrases are present: *as a result* and *in addition*. LeThanh's (2004) NP and VP cues are also relevant: *condition*, *effect*, *goal*, *requirement*, *situation* and *speculation* for NPs, and *to assume*, *to create*, *to mean* and *to result from* for VPs. Opposed to all other features mentioned, only the last (*to result from*) is present in only one partition. In the list with only the features that are present in more than one partition, two other VP cues can be found, namely *to bring* and *can be*. Some of the NP and VP cues are self-evident, e.g. *effect*, while others are less easy to link to rhetorical relations, e.g. *to create*. A larger data set could have provided more evidence for the benefit of such cues, while at this moment their relevance could be a mere coincidence.
As is the case with Relief, the model of Naive Bayes assign little relevance to cue phrases, NP cues and VP cues, at least too little to put them in the list of the 50 best features. Only the word similarity based on Lin's thesaurus and the word overlap for tokens and lemmas are present in the top 50 of all features. In the list with features present in more than one partition, more lexical features can be found, though none of them are cue phrases or NP or VP cues. New in this list are the word overlap in stems, the presence of time references and word similarity following WordNet::Similarity (Extended Gloss Overlap). It is not surprising that these features are important, but it is that the cue phrases are lacking in the list. Despite the fact that the literature, the data and our intuitions tell us that lexical information is one of the most important types of features, Naive Bayes performs worse (55.3%) than the baseline (56.0%) when only provided with the lexical features.

In the list of the 50 most relevant features according to Maximum Entropy, various lexical features are present: word overlap (tokens, lemmas and stems), word similarity (Lin and WordNet), the presence of time references and a number of word cues. The only cue phrase included is *but*, which is a very clear indicator of rhetorical relations, although there is a risk that it concerns a relation within a sentence instead of between (M-)SDUs (Timmerman 2007). Apparently, the cue phrase *but* is useful for the latter case. In the list are also the NP cue *result* and the VP cues *to assume*, *to bring*, *to have to*, *to include* and *to make*. Again, not all of the cues are intuitively obvious predictors of rhetorical relations and therefore the question arises whether the found features are really relevant or only relevant in the used data set. Offered only the lexical features, Maximum Entropy performs only slightly better (57.0%) better than the baseline (56.0%).
In the overall top 50 of most relevant features, word overlap is useful only in the right pair in the triple. For tokens and stems, the overlap in the left pair is present in the top 100. As Table 7.5 shows, a relatively high word overlap implies there is a rhetorical relation between the two (M-)SDUs concerned. This is exactly what one expects.

Table 7.5: Lexical features in overall top 50: word overlap

overlap type	position	direction CSS	direction ME
token overlap[a]	Right	Right	Right
lemma overlap	Right	Right	Right
stem overlap[a]	Right	Right	-

[a]the feature for *left* is present in the top 100

Table 7.6 shows that the same expected pattern is found for word similarity. The higher the similarity, the higher the chance that a rhetorical relation exists. It is commonly known that the wide coverage of WordNet may lead to problems when applied to specific domains such as financial newspaper texts. Because Lin's Thesaurus was trained on Wall Street Journal texts, it is not surprising that the similarity based on Lin's thesaurus is more useful for our task than that based on WordNet.

Table 7.6: Lexical features in overall top 50: word similarity

feature	position	direction CSS	direction ME
Lin's Dependency Thesaurus	Left	Left	Left
Lin's Dependency Thesaurus	Right	Right	Right

Time references are only useful enough to be in the top 50 when they occur in the right-most (M-)SDU in the triple. According to CSS and the model of Maximum Entropy, the presence of such reference in the last (M-)SDU indicates that the relation is probably on the left. This could mean that time references introduce new topics that are not rhetorically related to the previous (M-)SDUs. Because of the way the features have been determined, it is not possible to conclude whether a time reference has the same function when an adjacent (M-)SDU contains a time reference as well. It seems more logical that in those cases, time references predict the *presence* of a relation rather than the *absence*. As previously mentioned, future research could be directed at the interaction of features in order to discover in which cases time references are useful for

establishing a relations is absent or present.

Table 7.7: Lexical features in overall top 50: time references

feature	position	direction CSS	direction ME
time references	right	Left	Left

The overall top 50 does not include any of the cue phrases, NP cues or VP cues. On the one hand, this is surprising when one relates this to the literature, the researched data and human intuitions, but on the other hand, the data set is too small to enable us to find relevance for these words and phrases. Another explanation is Timmerman's (2007) remark that cue phrases can be indicators of relations with other (M-) SDUs as well as indicators for internal relations.

7.4 Reference features

In Chapter 3, in which we established the list of potentially relevant features, we found a rather long list of reference features. Of course, when a writer refers back to something or someone mentioned in the previous (M-)SDU, the two (M-)SDU share a topic and therefore are probably rhetorically related. This section shows that our expectations that reference features are very useful in the present task are confirmed.
Relief's top 50 of most relevant features contains six different reference features. The presence of a personal pronoun or a definite article in the first clause of the right (M-)SDU in an (M-)SDU pair is a good indicator of a rhetorical relation according to Relief. This confirms our initial expectations. Another useful feature is the relative number of personal pronouns in an (M-)SDU, as already became apparent in the section on surface features. As expected, anaphoric relations are also good predictors of rhetorical relations. The same is the case for missing modifiers (NP simplification), in which a modified noun phrase is repeated in a next (M-)SDU but without the modifiers. The top 50 of Relief also includes one helpful reference word, namely *more*.
The list of the best features according to CSS contains the same reference features as those according to Relief except *more*, plus some others. Next to missing modifiers, CSS also considers the other type of NP simplification, a missing NP head, relevant. Also the relative number of demonstrative pronouns is included in the top 50. It is rather surprising that this feature is not in Relief's top 50 since intuitively it is a very clear predictor of a rhetorical relation. The reference words *added*, *further*, *less* and *other* are also in the top 50 of CSS. All features mentioned here are present in more than one partition. In the list including only features present in more than one partition, one more reference word is included: *previous*. The benefit of the reference words is self-evident.
In the top 50 of the model of Naive Bayes, only personal pronouns and definite articles are present, both represented by their presence or absence in the first clause and their relative number in the whole (M-)SDU. In the list with only the features present in more than one partition, two more reference features are present: missing modifiers and the relative number of demonstrative pronouns. Only offering reference features to Naive Bayes leads to an accuracy that is (nearly) as high (59.5%) as that reached with all features (60.0%). This result underlines the importance of reference information for the detection of rhetorical relations between (M-)SDUs.

Most features in the top 50 of Maximum Entropy are also found by the other algorithms: the presence of a personal pronoun in the first clause, the presence of a definite article in the first clause, the relative number of definite articles in the (M-)SDU, the relative number of demonstrative pronouns, anaphora, missing modifiers and the words *added*, *further*, *less* and *other*. Features that have not yet been mentioned but are present in the list of best features according to Maximum Entropy are the relative number of determiners, and the reference word *later*. When offered the reference features solely, Maximum Entropy obtains a classification accuracy of 59.7%, which is almost as good as that reached with all features (60.9%).
Table 7.8 shows that anaphora are useful in both pairs in the machine learning triples. When there is an anaphoric relation between two (M-)SDUs, it is an indication that there is a rhetorical relation as well.

Table 7.8: Reference features in overall top 50: anaphora

reference feature	position	direction CSS	direction ME
anaphora	Left	-	Left
anaphora	Right	Right	-

The presence of personal pronouns in the second (M-)SDU in a pair is also important in both pairs in the triple (Table 7.9). For the features that are also present in the top 50s of CSS and/or Maximum Entropy, the predicted direction is as we would expect: the presence of personal pronouns predicts a rhetorical relation.

Table 7.9: Reference features in overall top 50: personal pronouns

reference feature	position	direction CSS	direction ME
nr of pers. pronouns	Left	-	-
nr of pers. pronouns	Right	Right	-
pers. pronoun in first clause	Left	-	Left
pers. pronoun in first clause	Right	Right	Right

In Table 7.10, the features for the definite article are presented. Only the features concerning the right pair in the triple are present in the top 50. As we already saw in the discussion of the surface feature *the* in Section 7.1, the presence of a definite article in the first clause of the second (M-)SDU in the right pair (thus in the right-most (M-)SDU) indicates that the relation is on the left. This is not the direction we have hypothesized, since we expected a definite article to refer back to a previously mentioned entity. Journalists of the Wall Street Journal probably assume that readers are familiar with the economy and other related notions and topics, in other words they expect given background. Definite articles can thus not only refer to what has been mentioned in the article, but also to what is assumed to be known. Although the algorithms have benefited from this feature, there is also still a risk that its relevance is based on coincidence.
The relative number of demonstrative pronouns in the right pair of the triple is in the top 50, while that in the left pair is in the top 100 (Table 7.11). It is to be expected that the presence of demonstrative pronouns in the second (M-)SDU of a pair predicts a rhetorical relation, and this is indeed confirmed by both CSS and Maximum Entropy.

Table 7.10: Reference features in overall top 50: definite articles

reference feature	position	direction CSS	direction ME
nr of def. articles	Right	-	-
def. article in first clause[a]	Right	Left	Left

[a]the feature for *Left* is present in the top 100

Table 7.11: Reference features in overall top 50: demonstrative pronouns

reference feature	position	direction CSS	direction ME
nr of dem. pronouns[a]	Right	Right	Right

[a]the feature for *Left* is present in the top 100

In the list of reference words in Table 7.12[3], only *further* shows a pattern that is unexpected. One would expect that the presence of *further* in the middle (M-)SDU indicates that the middle (M-)SDU refers back to the first (M-)SDU and thus that the relation is on the left. This appears not to be the case. In our data, *further* seems to ask for an elaboration. We therefore must conclude that the example presented in Section 3.2.3, from which we concluded that *further* is a reference words referring to a something previously mentioned, is not representative. Our intention to find a feature set that is as complete as possible has obviously led to overgeneralization. This could also have consequences for other features and their relevance scores.

Table 7.12: Reference features in overall top 50: reference words

reference word	position	direction CSS	direction ME
added[a]	Right	Right	Right
further	Left	Right	Right
less	Left	Left	Left
less	Right	-	Right
more	Right	-	-
other	Left	-	Left
other	Right	Right	Right

[a]the feature for *Left* is present in the top 100

NP simplification in the form of missing modifiers is useful in both pairs in the machine learning triples. The pattern in Table 7.13 is exactly as expected. This shows that we have introduced a feature that is indeed useful for the given task.

[3]The presence of reference words has only been checked in the second (M-)SDU of each pair in the triple, because we assume they refer to something that has been previously mentioned.

Table 7.13: Reference features in overall top 50: NP simplification

reference feature	position	direction CSS	direction ME
missing modifier	Left	-	Left
missing modifier	Right	Right	Right

7.5 Discourse features

We have introduced discourse features in Chapter 3 in order to make the notion of right-skewedness (Marcu 2000) concrete. Also, the data showed us that punctuation is useful in finding rhetorical relations. Each of the four algorithms includes at least one discourse feature in their top 50s.

According to Relief, only the internal discourse structure of an (M-)SDU is relevant enough to be in the top 50. It seems that the rhetorical structure of the context helps in predicting the presence of a rhetorical relation, and thus that the discourse structure of paragraphs is predictable, at least in our data set. An explanation is that journalists are trained in writing articles that describe certain events (related to the economy in our case) in a limited number of words. It is not surprising that their writing has a standard, probably right-skewed, structure.

Both lists with the best features following CSS (one with all features and one with only those present in more than one partition) include the same discourse features: the position in the paragraph (sentence number) and continuous punctuation in the form of quotation marks. We had no expectations regarding the former, but its importance can again be explained by the fact that the paragraphs in newspaper articles are written in a certain structure. It seems natural that two (M-)SDUs containing spoken language (indicated by the quotation marks) are rhetorically related, though one could think of examples in which it is not the case.

In the top 50 following the model of Naive Bayes, the features internal discourse structure and continuous punctuation (quotation marks) can be be found. They are considered useful by the above mentioned algorithms as well. When only taking the features into account that are present in more than one partition, the position in the text (paragraph number) and in the paragraph (sentence number) are also in the top 50. Once again we suspect this to be caused by the rather standard newspaper style. Information on the structure, however, is not enough to be able to detect rhetorical relations: Naive Bayes performs worse than the baseline (56.0%) when provided with only discourse features (55.7%).

Maximum Entropy considers discourse structure and continuous punctuation in the form or quotation marks relevant features. As is the case with Naive Bayes, discourse features solely lead to a classification accuracy (55.9%) that does not exceed the baseline (56.0%).

The overall top 50 includes both features describing the position of the (M-)SDU in the text (the paragraph number in the text and the sentence number in the paragraph), and they are in the top 100 for each of the three (M-)SDUs in a triple (Table 7.14). As mentioned above, we believe the relevance of these features can be explained by the rather general structure of newspaper articles.

Table 7.15 shows that continuous punctuation is only in the overall top 50 (and 100) in the form of quotation marks in the right pair of the machine learning triple. As expected, the presence of quotation marks in both (M-)SDUs in this pair is a cue for the presence

Table 7.14: Discourse features in overall top 50: position in the text

position in	position	direction CSS	direction ME
paragraph (sentence nr)[a]	middle	Right	-
paragraph (sentence nr)	right	Right	-
text (paragraph nr)	left	-	-
text (paragraph nr)	middle	-	-
text (paragraph nr)	right	-	-

[a]the feature for *left* is present in the top 100

of a rhetorical relation in both CSS and the model of Maximum Entropy.

Table 7.15: Discourse features in overall top 50: continuous punctuation

punctuation	position	direction CSS	direction ME
quotation marks	Right	Right	Right

The assumed newspaper discourse structure is also described by the internal discourse structure of (M-)SDUs. The overall top 50 only contains the feature for the right-most (M-)SDU in the triple, but the feature for the left (M-)SDU is present in the top 100. Testing the feature on different text genres is necessary to establish whether our intuitions about newspaper structure are valid.

Table 7.16: Discourse features in overall top 50: internal discourse structure

feature	position	direction CSS	direction ME
internal discourse structure[a]	Right	-	-

[a]the feature for *Left* is present in the top 100

Chapter 8

Conclusion and recommendations for future research

The goal of this thesis was to find features that can be used to predict the presence of rhetorical relations between (M-)SDUs within paragraphs. To achieve this, potentially relevant features have been derived from literature on existing systems for discourse analysis and from a short study of a subset of the data. Next, the features have been made concrete in such a way that they could be extracted automatically. This was not possible for some features, e.g. ellipsis, newspaper style and world knowledge. We have developed a metric for syntactic similarity, and introduced the feature NP simplification.

After all feature values were automatically extracted, they were offered to various machine learning algorithms. We have simplified discourse analysis to a task in which the algorithm has to decide whether an (M-)SDU is rhetorically related to the preceding or the following (M-)SDU. Creating algorithms suitable for the described task ourselves was beyond the scope of this thesis. As a result, we used existing machine learning tools that were designed to handle numerical data. In order to be able to combine all possible features in a single machine learning approach, we converted our features to numerical features (either discrete or continuous), also in cases where the features were in fact nominal in nature.

The performance of the classification algorithms was disappointing: Of the five classification algorithms applied, only the models of Naive Bayes (Demsar et al. 2004) and Maximum Entropy (Zhang 2004) reached significant improvement over the baseline of selecting the most common direction (*right*). Causes may be the small data set, the large number of features, the parameter settings of the algorithms, the artificiality of the task and/or the extent to which the algorithms are able to deal with the type of information provided. Assuming that the two algorithms are able to sift the information to some degree and that the sifting is expressed in the model parameters, we have developed methods to rank the features according to their relevance. This was also performed for the feature selection algorithms Relief (Kononenko 1994) and CSS (van Halteren, personal communication). From the four rankings based on the separate algorithms, a final ranked list was created. An in-depth study of the suitability of the algorithms and our methods is not included in this thesis, so we must advise other researchers caution in taking the results described below for granted.

We have included five different feature types: surface, syntactic, lexical, reference and discourse. The most relevant surface features concern text characteristics that have also

been covered by the other (more sophisticated) feature types. Syntax appeared to be very useful for the detection of rhetorical relations: the higher the syntactic similarity, the more chance the (M-)SDUs in question were rhetorically related. Lexical cues, which have been used by all researchers mentioned in Chapter 2, were also beneficial in our task. A high word overlap or word similarity often meant there was a rhetorical relation. As expected, reference features were also very useful: the presence of anaphora, personal pronouns, demonstrative pronouns, reference words and missing modifiers were cues that a rhetorical relation was present. Discourse structure also helped in finding rhetorical relations, perhaps due to the rather common newspaper style (with right-skewedness within paragraphs). The presence of direct speech (indicated by quotation marks) also predicted the presence of a rhetorical relation.

Some features were relevant as we had expected, but the way in which they contributed deviated from our hypotheses. Time references often introduced new (not directly rhetorically related) topics, while we expected them to indicate the continuation of the topic discussed at that point. There were also differences between the expectations and observations for the word *further*. We expected it to refer back to something previously mentioned, while it often asked for elaboration. Also, the definite article *the* appeared not always to refer back to what had been said. Perhaps the writers of the Wall Street Journal assume the reader has *given knowledge* to which they can refer. These examples show that although our intention to find as many potentially relevant features as possible by looking at the data was good, it could have easily led to the generalization of single occurrences to rules that seem not to be representative of what actually happens. Fortunately, our machine learning based method prevented this by contradicting our false assumptions.

Some of the more surprising features that we found relevant may be explained by the methodology chosen or the data set applied. For example, cue phrases, NP cues and VP cues were not in the top 50 of most relevant features, while intuitively they are the most obvious indicators of rhetorical relations. Since the algorithms treat continuous and discrete features differently, and the features mentioned are typically represented in discrete (binary) features, their lack in the overall top 50 might be a result of their representation rather than their relevance. Also, a topic specific word such as *farmer* was considered relevant.

The experiments seem to indicate that when the features are based on the NUCLEUS sentence, the classification is equally good as or even better than when they are based on the full (M-)SDU. Still, it could be that it depends on the situation (the machine learning case), and the accuracies reached might improve when offered with the features based on both the NUCLEUS sentence and the full (M-)SDU. In the future, feature values should thus be determined for both to be able to draw final conclusions.

Due to computational complexity of the features, the focus of the thesis research has been on the process of making the features concrete and extracting them automatically from various resources. Objectives 3 and 4, which concern investigating which of the found features are most important and trying to explain why, have received minor attention. The systems and software employed were already existing and a thorough study of their suitability for the given task and data was not possible. Especially the consequences of the conversion of nominal data to continuous or discrete data should be studied in detail. In future research, such a study seems advisable. Also, the rather artificial decision task should be compared to the actual problem of discourse analysis. Future research should also try to answer the question whether it is sensible to rank features with different types

(continuous, discrete) with the systems employed here, and whether it is legitimate to derive a single ranking from them.
It would be useful to repeat the present research when more RST data is available. Future research could also be directed at the interaction between features, because features often seem more informative when they occur together with another feature. When two adjacent (M-)SDUs both contain time references, for example, a rhetorical relation may be more probable than when a single time reference occurs. The chance that a definite article refers back to something mentioned in a previous (M-)SDU is greater when the same noun has been used in this previous (M-)SDU but with an indefinite article. Another suggestion for future research is to employ other text genres than financial news papers in order to discover which of the features are considered relevant only because of the limitations of the writing style in this genre.
When all aforementioned recommendations have been taken into account, other research questions may arise, such as: Are the found features useful for the labeling of the rhetorical relations and establishing the nuclearity distribution as well? Such questions are only relevant when there is more certainty about the suitability of the methods and data.

Bibliography

Baayen, R. H., Piepenbrock, R. and Gulikers, L.(1995), The CELEX Lexical Database (CD-ROM), *Technical report*, Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.

Banerjee, S. and Pedersen, T.(2003), Extended gloss overlaps as a measure of semantic relatedness, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, pp. 805–810.

Benson, S. and More, J.(2001), A limited memory variable metric method for bound constrained minimization, *Technical report*, Argonne National Laboratory. Preprint ANL/ACSP909-0901.

Bernstein, Y. and Zobel, J.(2005), Redundant documents and search effectiveness, *Proceedings of the 14th ACM international conference on Information and knowledge management*, ACM Press, New York, Bremen, Germany, pp. 736–743.

Black, E., Abney, S., Flickenger, S., Gdaniec, C., Grishman, C., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B. and Strzalkowski, T.(1991), Procedure for quantitatively comparing the syntactic coverage of English grammars, *Proceedings of the workshop on Speech and Natural Language*, Leiden, pp. 306–311.

Bosma, W.(2005), Query-based summarization using rhetorical structure theory, *in* T. van der Wouden, M. Po, H. Reckman and C. Cremers (eds), *Proceedings of the 15th meeting of Computational Linguistics in the Netherlands (CLIN 2004)*, Leiden, pp. 29–44.

Carlson, L., Marcu, D. and Okurowski, M.(2003), Building a discourse-tagged corpus in the framework of rhetorical structure theory, *in* J. van Kuppevelt and R. Smith (eds), *Current Directions in Discourse and Dialogue*, Kluwer Academic Publishers, pp. 85–112.

Chang, C.-C. and Lin, C.-J.(2001), *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Charniak, E.(2000), A Maximum-Entropy-Inspired Parser, *Proceedings of the North American Chapter of the Association for Computational Linguistics 2000 (NAACL-2000)*, pp. 132–139.

Chomsky, N.(1957), *Syntactic structures*, Mouton, The Hague.

Corston-Oliver, S.(1998), *Computing Representation of Discourse Structure*, PhD thesis, Dept. of Linguistics, University of California, Santa Barbara.

Demsar, J., Zupan, B. and Leban, G.(2004), Orange: From Experimental Machine Learning to Interactive Data Mining, *Technical report*, Faculty of Computer and Information Science, University of Ljubljana. Software available at `http://www.ailab.si/orange`.

Fellbaum, C. E.(1998), *WordNet: An Electronic Lexical Database*, MIT Press (Cambridge, Mass.).

Hearst, M.(1997), TextTiling: Segmenting text into multi-paragraph subtopic passages, *Computational Linguistics* **23**(1), 33–64.

Kira, K. and Rendell, L.(1992), A practical approach to feature selection, *ML92: Proceedings of the ninth international workshop on Machine learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 249–256.

Kononenko, I.(1994), Estimating Attributes: Analysis and Extensions of RELIEF, *European Conference on Machine Learning*, pp. 171–182.

Lesk, M.(1998), Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone., *Proceedings of SIGDOC '86*.

LeThanh, H.(2004), *Investigation into an Approach to Automatic Text Summarisation*, PhD thesis, Middlesex University, U.K.

Lin, D.(1998), Automatic retrieval and clustering of similar words, *Proceedings of the 17th international conference on Computational linguistics*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 768–774.

Malouf, R.(2002), A comparison of algorithms for maximum entropy parameter estimation, *COLING-02: proceeding of the 6th conference on Natural language learning*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 1–7.

Mann, W. and Thompson, S.(1988), Rhetorical structure theory: Toward a functional theory of text organization, *Text* **8**(3), 243–281.

Marcu, D.(1999), A decision-based approach to rhetorical parsing, *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, College Park, Maryland, pp. 365–372.

Marcu, D.(2000), The rhetorical parsing of unrestricted texts: a surface-based approach, *Computational Linguistics* **26**(3), 395–448.

Marcus, P., Marcinkiewicz, M. and Santorini, B.(1993), Building a large annotated corpus of English: the Penn Treebank, *Computational Linguistics* **19**(2), 313–330.

Mitkov, R., Evans, R. and Orasan, C.(2002), A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method, *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002)*, Mexico City.

Ng, H., Teo, L. and Kwan, L.(2000), A machine learning approach to answering questions for reading comprehension tests, *in* T. van der Wouden, M. Po, H. Reckman and C. Cremers (eds), *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, Hong Kong, pp. 124–132.

O'Donnell, M.(2000), RSTTool 2.4: a markup tool for rhetorical structure theory, *Proceedings of the 1st International Natural Language Generation Conference*.

Pedersen, T., Patwardhan, S. and Michelizzi, J.(2004), WordNet::Similarity – Measuring the Relatedness of Concepts, *Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04)*, Boston, MA., pp. 38–41.

Poesio, M. and Alexandrov-Kabadjov, M.(2004), A general-purpose, off the shelf anaphoric resolver, *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC)*, Lisbon.

Polanyi, L., Culy, C., Thione, G. and Ahn, D.(2004), A Rule Based Approach to Discourse Parsing, *Proceedings of SigDial2004*.

Reitter, D. and Stede, M.(2003), Step by step: underspecified markup in incremental rhetorical analysis, *Proceedings of the 4th Int'l Workshop on Linguistically Interpreted Corpora (LINC-03)*, Budapest.

Sampson, G., Haigh, R. and Atwell, E.(1989), Natural language analysis by stochastic optimization: a progress report on project APRIL, *Journal of Experimental and Theoretical Artificial Intelligence* pp. 271–287.

Schilder, F.(2002), Robust discourse parsing via discourse markers, topicality and position, *Natural Language Engineering*.

Soricut, R. and Marcu, D.(2003), Sentence Level Discourse Parsing using Syntactic and Lexical Information, *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edmonton, Canada.

Stede, M. and Heintze, S.(2004), Machine-assisted rhetorical structure annotation, *Proceedings of the 20th Int'l Conference on Computational Linguistics (COLING)*, Geneva.

Taboada, M. and Mann, W.(2006), Applications of Rhetorical Structure Theory, *Discourse Studies* **8**(8), 67–588.

Timmerman, S.(2007), *Automatic Recognition of Structural Relations in Dutch Text*, Master's thesis, University of Twente, The Netherlands. Available at `http://vaporiser.student.utwente.nl/~{}timmie/afstuderen/Thesis.pdf`.

van Rijsbergen, C.(1979), *Information Retrieval*, Butterworths (London).

Verberne, S., Boves, L., Oostdijk, O. and Coppen, P.(2007), Discourse-based answering of why-questions, *TAL journal, special issue on Computational Approaches to Discourse and Document Processing*.

Vieira, R. and Poesio, M.(2000), An empirically-based system for processing definite descriptions, *Computational Linguistics*.

Voorhees, E.(2003), Overview of the TREC 2003 Question Answering Track, *Overview of TREC 2003*, pp. 1–13.

Wattanamethanont, M., Sukvaree, T. and Kawtrakul, A.(2005), Thai discourse relations recognition by using naive bayes classifier, *The Sixth Symposium on Natural Language Processing 2005 (SNLP 2005)*.

Webber, B.(2004), D-LTAG: Extending Lexicalized TAG to Discourse, *Cognitive Science* **28**(5), 751–779.

Zhang, L.(2004), *A Maximum Entropy Modeling Toolkit for Python and C++*. Software available at `http://www.nlplab.cn/zhangle/maxent.html`.

Appendix A

Resources, tools and scripts

A.1 Treebanks and thesauri

- RST Treebank (Carlson et al. 2003)
  - wsj_0000.out.dis
- Penn Treebank (Marcus et al. 1993) release 2 (1994): Wall Street Journal
  - WSJ_0000.POS
  - WSJ_0000.PRD
  - WSJ_0000.MRG
- WordNet-2.1 (Fellbaum 1998)
- CELEX (Baayen et al. 1995)
  - english/emw/emw.cd
  - english/eml/eml.cd
- Dependency-Based Thesaurus (Lin 1998)
  - simN.lsp
  - simV.lsp
  - simA.lsp

A.2 Tools

- WordNet-Similarity-1.04 (Pedersen et al. 2004) (needs WordNet-QueryData-1,45, Text_Similarity-0.02 and WordNet-2.1)
- GuiTAR (Poesio and Alexandrov-Kabadjov 2004)
- Orange (Demsar et al. 2004)

A.3 Other sources

- matrix_Levenshtein_costs.txt (based on intuitions)
- cue_phrases_LeThanh04.txt (based on LeThanh 2004)
- NP_cues_LeThanh04.txt (based on LeThanh 2004)
- VP_cues_LeThanh04.txt (based on LeThanh 2004)
- ref_adv_adj.txt (found words supplemented by synonyms taken from the thesaurus of Microsoft Word 2003)
- folds_nr_by_textID.txt and sub_folds_nr_by_textID.txt (division in partitions, see Appendix E)

A.4 Scripts

All scripts can be downloaded from `http://lands.let.ru.nl/~daphne`

A.4.1 find_MSDU+relations_triples.pl

This script reads dis-files from the RST Discourse Treebank (Carlson et al. 2003) and finds relations between two (Multi-)Sentential Discourse Units within the same paragraph. Moreover, it finds triples suitable for machine learning. Before using machine learning algorithms, we try to find the features by looking at a number of discourse trees. To make this easier, this script creates a MSDUrel-file (see output). The procedure taken in this script is to make hashes for all (M-)SDUs, all relations and all triples. These hashes are used to print information to separate files that will be used in the feature extraction process. The hashes are also created for (relations/triples with) (M-)SDUs for which we were able to find a single Nucleus sentence. Some remarks: Nodes that include incomplete sentences are not included in the hash of (M-) SDUs, and relations that are not binary are excluded from the relations hash. See documentation in the script itself for more detailed information.

Input:

- dis-file from the RST Discourse Treebank (wsj_0000.out.dis)

Output:

- MSDUrel-file (wsj_0000.MSDUrel), showing:
  - a full representation of the source text, where sentence breaks are indicated by a single return and paragraph endings by a blank line. Each EDU is followed by its number between brackets.
  - for each paragraph the relations between (M-)SDUs in the same paragraph, printed in output file like this:
    * sentence number(s) <tab> (EDU-number(s)) <tab> sentence-text
    * sentence number(s) of related span <tab> (EDU-number(s)) <tab> span-text

    - rhetorical relation
    - nuclearity role
  - the number of relations between two SDUs (one sentence), one SDU and one M-SDU (more than one sentence), one M-SDU and one SDU, and two M-SDUs.

- MSDU-file (wsj_0000.MSDU), showing:
  - the unique name of the (M-)SDU
  - the number of sentences in the (M-)SDU => length
  - the number of the first sentence of the (M-)SDU in the paragraph => position
  - the number of the paragraph in which the (M-)SDU is present) => position
  - the text of the whole (M-)SDU

- NUCL_MSDU-file (wsj_0000.NUCL_MSDU): Same as MSDU-file above, but including only (M-)SDUs for which we were able to find a single Nucleus sentence (either an SDU or not)

- TRIPLE-file (wsj_0000.TRIPLE), showing
  - unique name of left (M-)SDU <tab>
  - unique name of middle (M-)SDU <tab>
  - unique name of right (M-)SDU <tab>
  - which side is the correct relation (left or right)

- NUCL_TRIPLE-file (wsj_0000.NUCL_TRIPLE): Same as TRIPLE-file above, but including only triples consisting of (M-)SDUs for which we were able to find a single Nucleus sentence (either an SDU or not)

- COUNT-file (wsj_0000.COUNT), various counts on the data

- NUCL_COUNT-file (wsj_0000.NUCL_COUNT): Same as COUNT-file above, but based only on triples and relations consisting of (M-) SDUs for which we were able to find a single Nucleus sentence (either an SDU or not)

A.4.2 find_PTBtags_of_msdu.pl

This script finds the Penn Treebank part-of-speech tags corresponding to a given (M-)SDU in a POS-file and save it to a file. It reads a given POS-file from the Penn Treebank, normalises it to match the data in the RST Discourse Treebank, and saves it in an array. It then reads the text of an (M-)SDU and locates it in the POS-array. The lines are printed to the output file. The EOL's and indentation are not saved since the brackets suffice in describing the structure.

Input:

- WSJ_0000.POS (from the Penn Treebank)

- wsj_0000.MSDU or wsj_0000.NUCL_MSDU (from script A.4.1)

Output:

- wsj_0000.TAGS or wsj_0000.NUCL_TAGS, showing:
  - unique (NUCL_)MSDU name <tab>
  - tags

A.4.3 find_POS_word_trigrams_of_MSDU.pl

This script finds trigrams in the sentences in each (M-)SDU in the TAGS(or NUCL_TAGS)-files created by find_MSDU+relations+triples.pl (script A.4.1). The trigrams consist of three slots which can be filled with either the POS-tag or the word. This means that there are 8 different types of trigrams for each group of three words. We include all trigrams within sentence. Sentence boundaries are defined as full stops, question marks, exclamation marks or paragraph endings. Full stops in abbreviations can easily be excluded from this group since they are not tagged as punctuation marks.

Resources:

- TAGS-files (concatenated in all_tags.txt) or NUCL_TAGS-files (concatenated in all_nucl_tags.txt) from script A.4.2

Output:

- all_tag_trigrams.txt or all_nucl_tag_trigrams.txt, showing:
  - (nucl_)msdu-name <tab>
  - trigram

A.4.4 create_lemmalex_with_pos.pl

This script reads the emw.cd file in CELEX and finds the lemma and word class of each word in eml.cd. The found word classes are changed to their Penn Treebank equivalents. The information is saved in a hash and printed to the output file.

Resources:

- emw.cd (from CELEX)
- eml.cd (from CELEX)

Output:

- lemmas_for_word_with_POStag.txt, showing:
  - word <tab>
  - Penn Treebank POS tag <tab>
  - lemma

A.4.5 create_stemlex_for_lemmas.pl

This script reads the eml.cd file in CELEX and finds the ultimate stem of each lemma. If a lemma consists of more than one part, the parts are separated by '+' in CELEX. Since we want only one stem per lemma, we find the stem of the first part only. We search the ultimate stem: we search the stem of the stem until no new stem can be found or the new stem has been found already (circular). The information is saved in a hash and printed to the output file.

Resources:

- eml.cd (from CELEX)

Output:

- stems_for_lemmas.txt, showing:
  - lemma <tab>
  - ultimate stem

A.4.6 perl find_PTBtree_of_msdu.pl

This script finds the Penn Treebank syntactic tree corresponding to a given (M-)SDU in a PRD-file and save it to a file. It reads a given PRD-file from the Penn Treebank, normalises it to match the data in the RST Discourse Treebank, and saves it in an array. It then reads the text of an (M-)SDU and locates it in the PRD-array. The lines are printed to the output file. The EOL's and indentation are not saved since the brackets suffice in describing the structure.

Input:

- WSJ_0000.PRD (from the Penn Treebank)
- wsj_0000.MSDU or wsj_0000.NUCL_MSDU (from script A.4.1)

Output:

- wsj_0000.TREE or wsj_0000.NUCL_TREE, showing:
  - unique (NUCL_)MSDU name <tab>
  - tree

A.4.7 find_PTBtagtree_of_msdu.pl

This script finds the Penn Treebank syntactic tree corresponding to a given (M-)SDU in a MRG-file and save it to a file. It reads a given MRG-file from the Penn Treebank, normalises it to match the data in the RST Discourse Treebank, and saves it in an array. It then reads the text of an (M-)SDU and locates it in the MRG-array . The lines are printed to the output file. The EOL's and indentation are not saved since the brackets suffice in describing the structure.

Input:

- WSJ_0000.MRG (from the Penn Treebank)
- wsj_0000.MSDU or wsj_0000.NUCL_MSDU (from script A.4.1)

Output:

- wsj_0000.TAGTREE or wsj_0000.NUCL_TAGTREE
- unique (NUCL_)MSDU name <tab>
- tagtree

A.4.8 make_list_wordpairs_WordNetSimilarity.pl

This script finds all nouns and verbs in each (M-)SDU present in a (nucl_)triple and saves all possible pairs of nouns/verbs in two adjacent (M-)SDUs in a (nucl_)triple. After nouns, the code '#n' is added, after verbs '#v'. The list is printed to the output file, which can then be offered to the tool WordNet::Similarity.

Resources:

- TAGS-files (from script A.4.2, concatenated in all_tags.txt)
- NUCL_TAGS-files (from script A.4.2, concatenated in all_nucl_tags.txt)
- TRIPLE-files (from script A.4.1, concatenated in all_triples.txt)
- NUCL_TRIPLE-files (from script A.4.1, concatenated in all_nucl_triples.txt)

Output:

- list_wordpairs_WordNetSimilarity.txt, including all word pairs (with #n/v-codes)

A.4.9 add_MSDUlabels_to_MRG.pl

This script finds the Penn Treebank syntactic tree corresponding to a given (M-)SDU in a MRG-file and inserts codes to indicate the boundaries. It reads a given MRG-file from the Penn Treebank, normalizes it to match the data in the RST Discourse Treebank, and saves it in an array. It then reads the text of an (M-)SDU and locates it in the MRG-array (exactly as in script 8). The POS-tags in the tree are simplified in such a way that the output can be offered to the anaphora resolver GuiTAR. It then adds codes to the array that denote the boundaries of (M-)SDUs. In this way the (M-)SDUs can be found in the GuiTAR ouput (the codes are ignored by GuiTAR and therefore have no influence on the quality of the anaphora resolution).

Input:

- WSJ_0000.MRG (from the Penn Treebank)
- wsj_0000.MSDU or wsj_0000.NUCL_MSDU (from script A.4.1)

Output:

- wsj_0000.MRG_LABEL or wsj_0000.NUCL_MRG_LABEL

A.4.10 find_anaphora_in_msdu.pl

This script finds the numbered nominal expressions ('ne's) and anaphora (two linked 'ne's) in the output of GuiTAR.

Input:

- wsj_0000.GuiTAR.xml or wsj_0000.NUCL_GuiTAR.xml (GuiTAR output)
- wsj_0000.MSDU or wsj_0000.NUCL_MSDU (from script A.4.1)

Output:

- wsj_0000.ne or wsj_0000.NUCL_ne, showing:
  - unique (NUCL_)MSDU name <tab>
  - codes for nominal expressions (e.g. ne1), separated by <space>
- wsj_0000.anaphora or wsj_0000.NUCL_anaphora, showing:
  - unique (NUCL_)MSDU name <tab>
  - codes for anaphora (e.g. ne4-ne1), separated by <space>

A.4.11 find_labels_for_cost_matrix_levenshtein.pl

This script takes all unique (M-)SDUnames and trees as input and prints the labels at the main levels of all clauses to an output file. This file is used to determine the Levenshtein substitution costs (we need to now which labels are present).

Resource:

- TREE-files (from script A.4.6, concatenated in all_trees.txt)

Output:

- labels_for_cost_matrix_levenshtein.txt, showing for each (M-)SDU:
  - the (M-)SDU name
  - for each clause: the POS-tags at the main level

A.4.12 find_feature_values_triples_in_folds.pl

This script uses the output of all scripts above, and determines the feature values of each triple in the training and test set in each partition. Moreover, it creates data sets in which only certain feature types are included (see Section 5.3.3). It reads all input files and saves them into hashes. Then it loops through both sets (*training* and *test*) and all partitions (1-10). For each combination, the headers are printed to the output files, as Orange requests. Next, it loops through all triples and determines the feature values and prints them to the output files. Also, the triple names are printed to a separate file in case they are needed later (e.g. to check the output of the script or to find example of certain cases). The only argument the script takes is which data set is concerned: *all* for *all2136_MSDU*, *sub* for *sub1866_MSDU* and *nucl* for *sub1866_Nucl*. For details, the reader is referred to the documentation in the script itself.

Resources:

- all_(nucl_)triples.txt (concatenated output of script A.4.1)
- all_(nucl_)MSDU.txt (concatenated output of script A.4.1)
- all_(nucl_)tags.txt (concatenated output of script A.4.2)
- all_(nucl_)tag_trigrams.txt (concatenated output of script A.4.3)
- all_(nucl_)tagtrees.txt (concatenated output of script A.4.7)
- all_(nucl_)trees.txt (concatenated output of script A.4.6)
- all_(nucl_)rel.txt (concatenated output of script A.4.1)
- (nucl_)folds_nr_by_textID.txt (see Appendix E)
- Penn_Treebank_POS_tags.txt (see Appendix D.1)
- all_lemmas.txt (list of all lemmas present in all partitions)
- lemmas_for_word_with_POStag.txt (output of script A.4.4)
- stems_for_lemmas.txt (output of script A.4.5)
- matrix_Levenshtein_costs.txt (see Appendix D.2)
- simA.lsp
- simN.lsp
- simV.lsp
- all_similarity_wordpairs_WordNetSimilarity.txt (output of WordNet::Similarity::lesk for list created by script A.4.8)
- cue_phrases_LeThanh04.txt (see Appendix D.3)
- NP_cues_LeThanh04.txt (see Appendix D.4)
- VP_cues_LeThanh04.txt (see Appendix D.4)
- ref_adv_adj.txt (see Appendix D.5)
- all_(nucl_)ne.txt (concatenated output of script A.4.10)
- all_(nucl_)anaphora.txt (concatenated output of script A.4.10)

Output for each partition X (1-10) and set *Set* (*train*, *test*):

- (sub_(nucl_))all_selected_features_SetX.tab
- (sub_(nucl_))surf_only_features_SetX.tab
- (sub_(nucl_))no_surface_features_SetX.tab
- (sub_(nucl_))synt_only_features_SetX.tab
- (sub_(nucl_))no_syntactic_features_SetX.tab

- (sub_(nucl_))lex_only_features_SetX.tab
- (sub_(nucl_))no_lexical_features_SetX.tab
- (sub_(nucl_))ref_only_features_SetX.tab
- (sub_(nucl_))no_reference_SetX.tab
- (sub_(nucl_))disc_only_features_SetX.tab
- (sub_(nucl_))no_discourse_features_SetX.tab
- triples_(all/sub/nucl)_SetX.txt

Appendix B

The data sample

B.1 Texts in the data sample

text	#paragraphs	#sentences	#(M-)SDUs	#relations
wsj_0615	3	7	10	2
wsj_0619	10	23	30	7
wsj_0641	5	11	15	4
wsj_0645	5	8	11	3
wsj_0648	14	24	33	9
wsj_0656	5	9	12	3
wsj_0668	4	7	8	1
wsj_0671	39	73	94	23
wsj_0688	8	18	26	8
wsj_0693	22	36	44	9
wsj_1302	32	99	136	35
wsj_1303	18	41	55	15
wsj_1311	24	57	81	23
wsj_1312	11	44	51	7
wsj_1334	10	19	28	9
wsj_1350	3	6	7	1
wsj_1360	5	7	8	1
wsj_1367	19	84	127	45
wsj_1372	19	32	41	9
wsj_1377	27	70	88	23
wsj_1381	6	11	16	5
wsj_1388	14	58	84	26
wsj_1391	14	29	34	9
wsj_1988	11	18	23	5
wsj_2352	6	8	11	3
wsj_2356	15	36	57	21
wsj_2365	23	51	70	20
wsj_2381	19	43	60	17
wsj_2382	6	9	10	1
wsj_2391	5	12	17	5
Total	402	950	1287	350
Average	13.40	31.67	42.90	11.67

B.2 Relations in the data sample

text	SDU, SDU		SDU, M-SDU		M-SDU, SDU		M-SDU, M-SDU	
	total	sample	total	sample	total	sample	total	sample
wsj_0615	2	2	0	0	1	0	0	0
wsj_0619	6	6	0	0	0	0	1	1
wsj_0641	2	2	1	1	1	1	0	0
wsj_0645	3	3	0	0	0	0	0	0
wsj_0648	6	6	2	2	1	0	0	0
wsj_0656	3	3	0	0	0	0	0	0
wsj_0668	1	1	0	0	0	0	0	0
wsj_0671	18	8	3	2	2	1	0	0
wsj_0688	6	6	0	0	2	2	0	0
wsj_0693	9	9	0	0	0	0	0	0
wsj_1302	22	6	12	7	1	0	0	0
wsj_1303	9	7	4	2	1	1	1	1
wsj_1311	15	9	3	2	4	0	1	1
wsj_1312	6	5	1	1	0	1	0	0
wsj_1334	6	6	1	1	2	2	0	0
wsj_1350	1	1	0	0	0	0	0	0
wsj_1360	1	1	0	0	0	0	0	0
wsj_1367	19	3	14	6	9	3	3	3
wsj_1372	8	8	0	0	1	1	0	0
wsj_1377	13	4	5	4	4	2	1	1
wsj_1381	4	4	0	0	1	1	0	0
wsj_1388	12	5	8	5	5	2	1	1
wsj_1391	6	6	1	1	2	2	0	0
wsj_1988	5	5	0	0	0	0	0	0
wsj_2352	2	2	0	0	1	1	0	0
wsj_2356	14	7	0	0	7	3	0	0
wsj_2365	12	6	4	3	4	2	0	0
wsj_2381	13	9	4	3	0	0	0	0
wsj_2382	1	1	0	0	0	0	0	0
wsj_2391	3	3	1	1	1	1	0	0
Total	228	144	64	41	50	26	8	8

Appendix C

Potentially relevant features

C.1 Source(s)

The table shows in which source the potentially relevant features have been found: Corston-Oliver (1998), Marcu (1999), Marcu (2000), LeThanh (2004), Timmerman (2007) and the data sample.

type	*feature*	*C-O 98*	*M 99*	*M 00*	*LT 04*	*T 07*	*data*
Surface	words		x				
	POS tags		x				
	(M-)SDU length						
Syntactic	syntactic similarity						x
	tense, aspect and polarity	x					
Lexical	cue phrases	x	x	x	x	x	x
	NP and VP cues				x		
	word overlap		x	x		x	x
	word similarity		x		x	x	x
	time references				x		x
Reference	anaphora resolution	x			x		
	personal pronouns					x	x
	definite articles						x
	demonstrative pronouns						x
	reference words (e.g. *further*)						x
	(wh-)determiners (e.g. *which*)						x
	NP simplification						x
Discourse	position in the text						
	continuous punctuation						x
	internal discourse structure						

C.2 Characteristics

The last column indicates whether the feature should be determined for each of the three (M-)SDUs in a triple (*left*, *middle* and *right*) or for both (M-)SDU pairs (*Left* and *Right*) in a triple.

feature type	*feature*	*decription*	*number*	*type*	*values*	*for each*
Surface	words	presence of lemmas	around 2,700	nominal	*absent*, *present*	(M-)SDU l,m,r
	POS tags	presence of trigrams	almost 16,000	nominal	*absent*, *present*	(M-)SDU l,m,r
		relative frequency of unigrams	36	continuous	0-1	(M-)SDU l,m,r
	(M-)SDU length	number of words	1	discrete	1-194	(M-)SDU l,m,r
		number of sentences	1	discrete	1-11	(M-)SDU l,m,r
Syntactic	syntactic similarity	syntactic similarity	1	continuous	0-1	pair L,R
	tense, aspect and polarity	relative frequency of verb forms	6	continuous	0-1	(M-)SDU l,m,r
Lexical	cue phrases	presence of cue phrases	207	nominal	*absent*, *present*	pair L,R
	NP and VP cues	presence of NP and VP cues	97	nominal	*absent*, *present*	(M-)SDU l,m,r
	word overlap	rel. overlap of tokens,lemmas,stems	3	continuous	0-1	pair L,R
	word similarity	word similarity (WordNet, Lin)	2	continuous	0-1	pair L,R
	time references	presence of time references	1	nominal	*absent*, *present*	(M-)SDU l,m,r
Reference	anaphora resolution	presence of anaphoric relation	1	nominal	*absent*, *present*	pair L,R
	personal pronouns	relative frequency of pers. pr.	1	continuous	0-1	pair L,R
		presence of pers. pr. in 1st clause	1	nominal	*absent*, *present*	pair L,R
	definite articles	relative frequency of def. art.	1	continuous	0-1	pair L,R
		presence of def. art. in 1st clause	1	nominal	*absent*, *present*	pair L,R
	demonstrative pronouns	relative frequency of dem. pr.	1	continuous	0-1	pair L,R
		presence of dem. pr. in 1st clause	1	nominal	*absent*, *present*	pair L,R
	reference words (e.g. further)	presence of reference words	31	nominal	*absent*, *present*	pair L,R
	(wh-)determiners (e.g. which)	relative frequency of (wh-)det.	1	continuous	0-1	pair L,R
		presence of (wh-)det. in 1st clause	1	nominal	*absent*, *present*	pair L,R
	NP simplification	presence of missing mod. or head	2	nominal	*absent*, *present*	pair L,R
Discourse	position in the text	sentence number	1	discrete	1-13	(M-)SDU l,m,r
		paragraph number	1	discrete	1-45	(M-)SDU l,m,r
	continuous punctuation	presence of cont. punct.	3	nominal	*absent*, *present*	pair L,R
	internal discourse structure	disc. structure at highest node	1	nominal	see[a]	pair L,R

[a] present internal discourse structures: *not applicable* (>80%), *N1-S1*, *N1-N1*, *N1-S2*, *S1-N1*, *N2-S1*, *N1-N2*, *N1-S3*, *N2-S2*, *N1-S4*, *S2-N1*, *N3-S1*, *N3-S2*, *N4-S1*, *N2-N1*, *N1-N3*, *N1-S6*, *N2-S4*, *N2-S3*, *S1-N2*, *S2-N3*, *N1-S10*, *N3-N2*, *N3-S7*, *N4-S3*, *N5-S1*, *S3-N1*, *S4-N1*, *S5-N1*

Appendix D

Extracting features values

D.1 Tag set of Penn Treebank

Code	Meaning
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present

Code	Meaning
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

D.2 Substitution costs for syntactic similarity

First category	Second category	Substitution cost
ADJP	ADVP	0.5
ADJP	WHADJP	0
ADJP	WHADVP	0.5
ADVP	CONJP	0.5
ADVP	WHADJP	0.5
ADVP	WHADVP	0
CONJP	WHADVP	0.5
FRAG	INTJ	0.75
INTJ	X	0.5
NP	QP	0.5
NP	WHNP	0
PP	WHPP	0
S	SBAR	0
S	SBARQ	0.25
S	SINV	0.25
S	SQ	0.25
SBAR	SBARQ	0.25
SBAR	SINV	0.25
SBAR	SQ	0.25
SBARQ	SINV	0
SBARQ	SQ	0
SINV	SQ	0
WHADJP	WHADVP	0.5

D.3 Cue phrases from LeThanh (2004)

This list shows all cue phrases from LeThanh (2004), including information on their position in the (M-)SDU and the relation. When a mark (x) is present in the last column, the cue phrase has been found in our data.

Cue phrase	Pos. in (M-)SDU	Pos. in rel.	found
above all	beginning	right	
actually	any	right	
add to this	beginning	right	
additionally	any	right	
after a time	any	any	
after all	beginning	right	
after that	any	right	

Cue phrase	Pos. in (M-)SDU	Pos. in rel.	found
after this	any	any	
afterwards	any	right	
again	any	right	
all this time	beginning	right	
already	any	right	
also	any	right	
alternatively	any	right	
and another	beginning	right	
and then	beginning	right	
and	beginning	right	x
another time	any	right	
anyhow	end	right	
anyway	any	right	
arguably	any	any	
as a consequence	beginning	right	
as a corollary	beginning	right	
as a logical conclusion	beginning	right	
as a result	beginning	right	x
as against	beginning	right	
as it turned out	beginning	right	
as well	any	right	
at first	beginning	left	
at last	any	right	
at least	beginning	right	x
at that moment	beginning	right	
at that time	beginning	right	
at the beginning	any	left	
at the end	any	right	
at the moment	beginning	right	
at the outset	beginning	left	
at the same time	any	right	
at this date	any	right	
at this moment	any	any	
at this point	any	any	
at this stage	any	right	
because of this	beginning	right	
before	any	any	
besides that	beginning	right	
besides	beginning	right	x
briefly speaking	beginning	right	
but	beginning	right	x
by comparison	any	right	
by contrast	beginning	right	x
by that time	any	right	
by then	any	right	
certainly	any	right	
clearly	any	right	

Cue phrase	Pos. in (M-)SDU	Pos. in rel.	found
compactly	any	right	
compendiously	any	right	
conceivably	any	right	
consequently	beginning	right	
contrariwise	any	right	
conversely	any	right	
decidedly	any	any	
definitely	any	any	
doubtless	any	any	
earlier	any	right	
either case	any	right	
either event	any	right	
either way	any	right	
equally	any	right	
eventually	any	right	
every time	any	any	
everywhere	any	any	
for example	any	right	
for instance	any	right	
for the matter	beginning	any	
for this reason	beginning	any	
for this	beginning	any	
formerly	any	any	
from now on	any	any	
from then on	any	any	
furthermore	beginning	right	x
here	any	any	
heretofore	any	any	
hitherto	any	any	
however	any	right	
if not	beginning	right	
in a contrary	beginning	right	
in a different point of view	beginning	right	
in addition	beginning	right	x
in any case	any	any	
in brief	beginning	right	
in comparison	beginning	right	
in conclusion	beginning	right	
in consequence	beginning	right	
in contrast	beginning	right	x
in fact	beginning	right	x
in general	any	right	
in other respects	beginning	right	
in other ways	beginning	right	
in other words	beginning	right	x
in particular	any	right	
in point of fact	beginning	right	

Cue phrase	Pos. in (M-)SDU	Pos. in rel.	found
in short	beginning	right	
in summarisation	beginning	right	
in the beginning	any	left	
in the end	any	right	
in the event	any	any	
in the first place	any	right	
in the meantime	any	right	
in the opposite	beginning	right	
in this case	any	right	
in this connection	any	right	
in this respect	any	right	
in this way	any	right	
in turn	middle	right	x
initially	any	left	
instantly	any	any	
instead	any	right	
it can be concluded that	beginning	right	
it is because	beginning	right	
it is clear that	beginning	right	
it is easy to understand that	beginning	right	
it is explained that	beginning	right	
it is possible that	beginning	right	
it is true that	beginning	right	
it may be the case that	beginning	right	
it may seem that	beginning	right	
it stands to reason that	beginning	right	
just then	any	right	
last	any	right	
lastly	any	right	
latter	any	right	
let us assume	beginning	right	
let us consider	beginning	right	
meanwhile	beginning	right	x
more accurately	any	any	
more importantly	any	right	
more precisely	any	any	
more specifically	any	any	
more to the point	beginning	right	
moreover	beginning	right	x
most likely	any	right	
neither	any	any	
never again	any	right	
next	beginning	right	x
now that	any	any	
now	any	any	
on a different note	any	right	
on another	beginning	right	

Cue phrase	Pos. in (M-)SDU	Pos. in rel.	found
on one side	beginning	left	
on the bases	any	any	
on the basis	any	any	
on the contrary	beginning	right	
on the other hand	beginning	right	x
on the other side	beginning	right	
on this basis	any	right	
once again	any	right	
once more	any	right	
only because	any	right	
or	beginning	right	x
otherwise	any	right	
parenthetically	any	any	
perhaps	any	right	
possibly	any	right	
presently	any	any	
presumably	any	any	
previously	any	right	
similarity	any	right	
simply because	any	right	
simultaneously	any	any	
since	beginning	any	
so	beginning	right	x
some time	any	any	
still	beginning	right	x
subsequently	any	right	
succinctly	any	right	
summarising	any	right	
summing up	any	right	
suppose that	beginning	left	
that is how	beginning	right	
that is to say	beginning	right	
that is why	beginning	right	
the fact is	beginning	right	
the moment	any	any	
the more	any	right	
then again	any	right	
then	any	right	
thereafter	any	right	
thereby	any	right	
therefore	any	right	
thereupon	any	right	
this time	any	right	
thus far	any	right	
thus	any	right	
to conclusion	beginning	right	
to explain	beginning	right	

Cue phrase	Pos. in (M-)SDU	Pos. in rel.	found
to summary	beginning	right	
to this end	any	right	
to wit	beginning	right	
ultimately	any	right	
under the circumstances	any	any	
under these circumstances	any	any	
until then	beginning	right	
up to now	beginning	any	
up to this	beginning	right	
we can understand that	beginning	right	
whereby	beginning	right	
yet	beginning	right	x

D.4 NP and VP cues from LeThanh (2004)

This list shows all NP and VP cues from LeThanh (2004). When a mark (x) is present in the second column, the cue has been found in our data.

NP cue	found	VP cue	found
abstract		*affect*	x
aim		*aim*	x
assumption		*answer*	x
brief	x	*assume*	x
cause	x	*be because*	x
condition	x	*be essential*	x
conjecture		*be important*	x
effect	x	*be necessary*	
essential		*be requisite*	
fact	x	*be why*	
following		*brief*	x
goal	x	*bring*	x
guess		*can be*	x
hypothesis/ses		*cause*	x
hypothezis/zes		*come after*	x
important		*conclude*	x
intent	x	*conjecture*	
intention	x	*consist*	x
main idea		*create*	x
meaning		*decrease*	x
necessary		*drop*	x
objective	x	*effectuate*	
opposite		*fail*	x
outcome	x	*fall*	x
outline		*follow*	x
possibility	x	*guess*	x
purpose	x	*have to*	x
reason	x	*hypothesise*	

reckoning		*hypothesize*	
requirement	x	*imagine*	x
requisite		*include*	x
result	x	*increase*	x
situation	x	*induce*	
solution	x	*make*	x
speculation	x	*mean*	x
summarisation		*must*	
supposition		*opine*	
target	x	*purpose*	
turning point	x	*raise*	x
		react	x
		reckon	x
		reply	
		require	x
		resolve	x
		respond	x
		result from	x
		solve	x
		speculate	x
		stand for	x
		succeed	x
		succeed	x
		summary	
		suppose	x
		suspect	x
		think	x
		to (+verb)	x
		translate	x
		understand	x

D.5 Reference words

Reference word	found in our data
added	x
additional	x
additionally	
alternative	x
alternatively	x
another	x
concluding	x
different	x
differently	x
extra	x
final	x
finally	x
following	x
former	x
formerly	x
further	x
last	x
lastly	
later	x
latter	x
less	x
more	x
next	x
other	x
previous	x
previously	x
subsequent	x
subsequently	x
succeeding	x
successive	x
supplementary	

Appendix E

Machine Learning

see next page...

E.1 Text IDs and number of triples in partitions of *all2136*

1		**2**		**3**		**4**		**5**		**6**		**7**		**8**		**9**		**10**	
ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr
0624	1	0606	6	0621	1	0655	1	0632	7	0629	20	0601	2	0609	9	0602	4	0604	18
0627	2	0633	25	0634	6	0674	1	0675	2	0682	7	0607	1	0610	10	0619	5	0616	5
0635	1	0679	1	0696	1	0689	6	0676	8	0688	3	0617	37	0615	2	0638	2	0637	3
0636	1	0683	1	1128	58	0694	8	0686	2	0695	2	0628	3	0623	2	0641	2	0642	2
0640	3	0687	10	1130	1	1110	4	0693	2	1101	4	0651	4	0631	4	0671	10	0648	3
0646	1	1127	2	1131	26	1135	2	1125	4	1102	2	0666	4	0664	30	0692	34	0669	1
0654	1	1138	2	1139	10	1147	11	1141	6	1105	4	0681	20	0690	5	1121	18	0672	1
0657	1	1301	2	1159	1	1149	10	1145	2	1120	28	1103	5	1111	4	1143	9	0677	10
0668	1	1307	26	1161	24	1174	23	1151	5	1140	5	1107	3	1122	18	1189	3	1126	1
1119	1	1320	64	1178	1	1184	1	1154	40	1150	6	1118	3	1142	7	1192	12	1144	3
1124	1	1327	15	1180	8	1193	5	1157	4	1166	4	1136	7	1156	3	1309	7	1152	3
1146	187	1340	1	1306	2	1303	14	1158	16	1182	1	1137	29	1160	4	1311	19	1171	9
1179	1	1343	1	1330	2	1323	16	1162	10	1183	4	1163	18	1164	4	1332	2	1175	2
1181	1	1931	1	1331	16	1367	58	1313	2	1312	10	1172	6	1165	12	1334	3	1302	33
1304	1	1934	1	1364	2	1985	1	1317	28	1347	3	1300	4	1169	3	1339	4	1315	26
1325	1	1962	15	1381	1	1992	1	1319	11	1353	3	1305	4	1190	20	1368	7	1316	9
1350	1	1963	2	1386	2	2316	7	1322	14	1377	17	1372	3	1314	6	1371	4	1328	33
1358	1	1980	1	1387	5	2339	1	1352	2	1388	38	1376	7	1375	7	1373	4	1337	7
2315	1	1999	8	1973	2	2352	1	1366	21	1389	7	1380	3	1924	4	1394	32	1382	1
2326	2	2325	12	2336	1	2362	8	1379	2	1397	11	1391	3	1970	4	1974	4	1390	12
2327	1	2331	8	2340	2	2366	28	1396	4	1930	5	1984	10	2303	4	1988	5	2332	1
2328	1	2360	7	2343	8	2367	2	1399	4	2321	13	2320	11	2341	9	2308	6	2344	5
2353	1	2394	2	2348	12	2380	3	1944	7	2345	4	2342	5	2350	5	2338	9	2346	19
2393	1	2396	1	2356	7	2391	2	2309	8	2373	3	2365	13	2359	3	2347	5	2375	6
				2399	15			2357	2	2398	9	2381	9	2386	35	2358	3		
24	214	24	214	25	214	24	214	25	213	25	213	25	214	25	214	25	213	24	213

E.2 Text IDs and number of triples in partitions of *sub1866*

1		**2**		**3**		**4**		**5**		**6**		**7**		**8**		**9**		**10**	
ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr	ID	#tr
0624	1	0606	6	0621	1	0655	1	0632	7	0629	17	0601	2	0609	9	0602	4	0604	12
0635	1	0627	1	0634	5	0674	1	0675	2	0682	7	0607	1	0610	10	0619	5	0616	5
0636	1	0633	20	0696	1	0689	5	0676	5	0688	3	0617	35	0615	2	0623	2	0637	3
0640	3	0679	1	1128	56	1110	4	0686	2	0695	2	0628	3	0631	3	0641	2	0638	1
0646	1	0683	1	1130	1	1135	2	0693	2	1101	4	0651	3	0664	26	0671	10	0642	2
0654	1	0687	9	1131	21	1147	8	1125	4	1102	2	0666	4	0690	5	0692	27	0648	3
0657	1	1138	2	1139	10	1149	9	1141	4	1105	4	0681	19	1111	4	1121	16	0669	1
0668	1	1301	2	1159	1	1174	19	1145	2	1120	18	0694	3	1122	18	1143	9	0672	1
1119	1	1307	17	1161	18	1184	1	1151	5	1140	5	1103	4	1142	6	1189	3	0677	7
1124	1	1320	56	1178	1	1193	3	1154	34	1150	6	1107	3	1156	3	1309	7	1144	2
1127	2	1327	13	1180	8	1323	13	1157	4	1166	3	1118	3	1160	4	1311	18	1171	8
1146	154	1340	1	1192	10	1367	54	1158	13	1182	1	1136	7	1164	3	1332	2	1175	2
1179	1	1343	1	1306	2	1376	6	1162	9	1183	4	1137	22	1165	12	1334	3	1302	33
1181	1	1931	1	1330	2	1985	1	1313	2	1303	14	1163	14	1169	3	1339	4	1315	22
1304	1	1934	1	1331	14	1992	1	1317	22	1312	9	1172	6	1190	15	1368	7	1316	7
1325	1	1962	15	1364	2	2316	7	1319	11	1347	3	1300	4	1314	6	1371	4	1328	29
1350	1	1963	2	1381	1	2339	1	1322	14	1353	3	1305	4	1375	7	1373	4	1337	7
1358	1	1980	1	1386	2	2352	1	1352	2	1377	14	1372	3	1924	4	1394	26	1382	1
1396	4	1999	8	1387	5	2362	7	1366	20	1388	31	1380	3	1970	4	1974	4	1390	12
2315	1	2325	12	1973	2	2366	28	1379	2	1389	6	1391	2	2303	4	1988	5	2332	1
2327	1	2331	7	2336	1	2367	2	1399	4	1397	8	1984	8	2341	9	2308	6	2344	5
2328	1	2360	7	2340	2	2380	3	1944	7	1930	5	2320	9	2350	5	2338	8	2346	15
2353	1	2394	2	2343	5	2391	2	2309	8	2321	11	2342	3	2359	3	2347	4	2357	1
2358	3	2396	1	2348	10	2398	8	2326	2	2345	3	2365	13	2386	22	2356	7	2375	6
2393	1			2399	5					2373	3	2381	9						
25	186	24	187	25	186	24	187	24	187	25	186	25	187	24	187	24	187	24	186

Appendix F

Relevant features

This Appendix shows the 50 most relevant features according to different machine learning algorithms. The small letters *l* (*left*), *m* (*middle*) and *r* (*right*) in the feature prefix refer to the (M-)SDU in the triple on which the feature is based; the capital letters *L* (*Left*) and *R* (*Right*) in the prefix indicate on which (M-)SDU pair in the triple the features are based. The features are here represented by self-explaining codes. In case of doubt, the reader is referred to Chapter 4, in which all features are described.

F.1 Relief: 50 best scoring features

The class (*left*, *right*) towards which the feature values point cannot be extracted from Relief.

Feature	*Feature type*	*Relief score*
R_pers_pr_cl1	reference	0.0401
m_time_ref	lexical	0.0291
r_*to*	surface	0.0233
l_*and*	surface	0.0219
m_past	syntactic	0.0210
r_past	syntactic	0.0205
R_def_art_cl1	reference	0.0200
r_present	syntactic	0.0200
L_def_art_cl1	reference	0.0199
m_present	syntactic	0.0196
r_*the*	surface	0.0190
l_time_ref	lexical	0.0188
r_time_ref	lexical	0.0185
m_*to*	surface	0.0182
r_*it*	surface	0.0182
L_pers_pr_cl1	reference	0.0174
m_*be*	surface	0.0152
l_*to*	surface	0.0149
l_past	syntactic	0.0148
l_present	syntactic	0.0144
m_*a*	surface	0.0142
l_*of*	surface	0.0140

Feature	*Feature type*	*Relief score*
m_*that*	surface	0.0140
m_*in*	surface	0.0134
r_*in*	surface	0.0134
r_*a*	surface	0.0127
m_*have*	surface	0.0126
r_PRP	surface	0.0124
m_*of*	surface	0.0122
r_*and*	surface	0.0121
l_*for*	surface	0.0119
L_anaphora	reference	0.0117
m_*but*	surface	0.0109
m_*for*	surface	0.0108
l_*in*	surface	0.0106
L_missing_mod	reference	0.0105
m_*it*	surface	0.0105
R_disc_str	discourse	0.0104
l_modal	syntactic	0.0102
r_modal	syntactic	0.0099
r_*i*	surface	0.0097
m_*make* (VP)	lexical	0.0096
m_*as*	surface	0.0095
r_*mr.*	surface	0.0095
R_anaphora	reference	0.0093
l_*it*	surface	0.0093
R_nr_pers_pr	reference	0.00920
m_*'s*	surface	0.0092
R_*more*	reference	0.00890
l_*be*	surface	0.0089

F.2 CSS: 50 best scoring features

The direction shows the class most selected when the value of a continuous feature is high (e.g. syntactic overlap) or when a binary feature (e.g. a cue phrase) is present.

Feature	*Feature type*	*CSS score*	*direction*
R_pers_pr_cl1	reference	0.1469	right
R_nr_pers_pr	reference	0.1331	right
R_cont_quotmarks	discourse	0.1256	right
R_wordsim_Lin	lexical	0.1181	right
r_PRP	surface	0.1168	right
L_wordsim_Lin	lexical	0.1125	left
R_token_overlap	lexical	0.1084	right
L_synt_sim	syntactic	0.0973	left
R_stem_overlap	lexical	0.0949	right
R_lemma_overlap	lexical	0.0919	right

Feature	*Feature type*	*CSS score*	*direction*
R_def_art_cl1	reference	0.0858	left
R_missing_mod	reference	0.0841	right
m_*affect*	lexical	0.0819	right
L_missing_head	reference	0.0797	left
l_past	syntactic	0.0765	left
l_*situation*	lexical	0.0764	right
r_time_ref	lexical	0.0743	left
m_gerund	syntactic	0.0727	right
m_*of*	surface	0.0724	right
m_*assume*	lexical	0.0692	right
R_*added*	reference	0.0679	right
r_NNP	surface	0.0674	left
r_*the*	surface	0.0659	left
l_*effect* (NP)	lexical	0.0646	right
l_present	syntactic	0.0643	right
l_*requirement*	lexical	0.0642	right
m_pos_in_par	discourse	0.0641	right
L_*further*	reference	0.0622	right
r_pos_in_par	discourse	0.0621	right
R_anaphora	reference	0.0612	right
m_*create*	lexical	0.0605	right
r_*mean* (VP)	lexical	0.0604	right
l_*speculation*	lexical	0.0604	right
m_*goal*	lexical	0.0603	right
L_*added*	reference	0.0601	left
l_*everyone*	surface	0.0594	right
m_NNP	surface	0.0592	right
m_*tractor*	surface	0.0581	right
L_*less*	reference	0.0580	left
R_*as_a_result*	lexical	0.0576	right
l_*condition* (NP)	lexical	0.0576	right
m_*to*	surface	0.0568	right
r_*it*	surface	0.0564	right
R_nr_dem_pr	reference	0.0560	right
L_*in_addition*	lexical	0.0560	left
l_*result_from*	lexical	0.0549	right
m_*a*	surface	0.0533	right
m_*little*	surface	0.0528	left
m_*equipment*	surface	0.0527	right
R_*other*	reference	0.0520	right

F.3 CSS: 50 best scoring features present in more than one partition

The direction shows the class most selected when the value of a continuous feature is high (e.g. syntactic overlap) or when a binary feature (e.g. a cue phrase) is present.

Feature	*Feature type*	*CSS score*	*direction*
R_pers_pr_cl1	reference	0.1469	right
R_nr_pers_pr	reference	0.1331	right
R_cont_quotmarks	discourse	0.1256	right
R_wordsim_Lin	lexical	0.1181	right
r_PRP	surface	0.1168	right
L_wordsim_Lin	lexical	0.1125	left
R_token_overlap	lexical	0.1084	right
L_synt_sim	syntactic	0.0973	left
R_stem_overlap	lexical	0.0949	right
R_lemma_overlap	lexical	0.0919	right
R_def_art_cl1	reference	0.0858	left
R_missing_mod	reference	0.0841	right
m_*affect*	lexical	0.0819	right
L_missing_head	reference	0.0797	left
l_past	syntactic	0.0765	left
l_*situation*	lexical	0.0764	right
r_time_ref	lexical	0.0743	left
m_gerund	syntactic	0.0727	right
m_*of*	surface	0.0724	right
m_*assume*	lexical	0.0692	right
R_*added*	reference	0.0679	right
r_NNP	surface	0.0674	left
r_*the*	surface	0.0659	left
l_*effect* (NP)	lexical	0.0646	right
l_present	syntactic	0.0643	right
l_*requirement*	lexical	0.0642	right
m_pos_in_par	discourse	0.0641	right
L_*further*	reference	0.0622	right
r_pos_in_par	discourse	0.0621	right
R_anaphora	reference	0.0612	right
m_*create*	lexical	0.0605	right
r_*mean* (VP)	lexical	0.0604	right
l_*speculation*	lexical	0.0604	right
m_*goal*	lexical	0.0603	right
L_*added*	reference	0.0601	left
m_NNP	surface	0.0592	right
L_*less*	reference	0.0580	left
R_*as_a_result*	lexical	0.0576	right
l_*condition* (NP)	lexical	0.0576	right
m_*to*	surface	0.0568	right
r_*it*	surface	0.0564	right

Feature	*Feature type*	*CSS score*	*direction*
R_nr_dem_pr	reference	0.0560	right
L_*in_addition*	lexical	0.0560	left
m_*a*	surface	0.0533	right
m_*little*	surface	0.0528	left
R_*other*	reference	0.0520	right
r_*bring*	lexical	0.0516	left
m_length_words	surface	0.0514	right
l_*can_be*	lexical	0.0505	right
L_*previous*	reference	0.0500	right

F.4 Naive Bayes: 50 best scoring features

The class (*left*, *right*) towards which the feature values point cannot be extracted from the model of Naive Bayes.

Feature	*Feature type*	*Naive Bayes score*
R_nr_pers_pr	reference	0.0476
r_PRP	surface	0.0447
R_pers_pr_cl1	reference	0.0428
R_wordsim_Lin	lexical	0.0421
L_wordsim_Lin	lexical	0.0339
R_cont_quotmarks	discourse	0.0317
R_token_overlap	lexical	0.0307
R_disc_str	discourse	0.0257
r_NNP	surface	0.0255
L_disc_str	discourse	0.0243
m_NNP	surface	0.0241
l_past	syntactic	0.0238
R_lemma_overlap	lexical	0.0213
l_VBZ	surface	0.0199
R_stem_overlap	lexical	0.0191
m_gerund	syntactic	0.0189
m_length_words	surface	0.0185
l_present	syntactic	0.0172
L_synt_sim	syntactic	0.0171
R_def_art_cl1	reference	0.0149
l_infinitive	syntactic	0.0144
R_nr_def_art	reference	0.0143
R_missing_mod	reference	0.0142
m_*of*	surface	0.0132
R_nr_dem_pr	reference	0.0131
m_VBG	surface	0.0130
m_CC	surface	0.0129
r_*the*	surface	0.0125
m_IN	surface	0.0124

Feature	*Feature type*	*Naive Bayes score*
m_DT	surface	0.0117
m_infinitive	syntactic	0.0116
r_time_ref	lexical	0.0111
L_nr_pers_pr	reference	0.0109
m_RB	surface	0.0107
l_gerund	syntactic	0.0107
L_nr_dem_pr	reference	0.0105
l_pos_in_text	discourse	0.0098
m_pos_in_text	discourse	0.0098
r_pos_in_text	discourse	0.0098
L_stem_overlap	lexical	0.0097
r_negative	syntactic	0.0096
m_PRP	surface	0.0096
r_pos_in_par	discourse	0.0096
L_token_overlap	lexical	0.0096
m_TO	surface	0.0096
R_wordsim_lesk	lexical	0.0094
m_JJ	surface	0.0090
r_VBD	surface	0.0089
r_present	syntactic	0.0087
l_negative	syntactic	0.0087

F.5 Maximum Entropy: 50 best scoring features

The direction shows the class most selected when the value of the discrete version of a continuous feature is high (e.g. syntactic overlap) or when a binary feature (e.g. a cue phrase) is present. For some discrete versions of continuous features there is no clear line. For *disc_str*, the class depends on each separate (discrete) feature value.

Feature	*Feature type*	*ME score*	*direction*
R_cont_quotmarks	discourse	0.3007	right
R_missing_mod	reference	0.2395	right
R_pers_pr_cl1	reference	0.2179	right
R_wordsim_Lin	lexical	0.2107	right
L_cont_quotmarks	discourse	0.1735	left
R_*added*	reference	0.1706	right
R_*other*	reference	0.1599	right
R_nr_dem_pr	reference	0.1568	right
L_nr_determiner	reference	0.1509	left
l_*include*	lexical	0.1503	left
L_wordsim_Lin	lexical	0.1483	left
m_NNP	surface	0.1409	right
R_*less*	reference	0.1338	right
m_*have_to*	lexical	0.1283	right
l_*make* (VP)	lexical	0.1281	right

Feature	*Feature type*	*ME score*	*direction*
L_pers_pr_cl1	reference	0.1280	left
R_token_overlap	lexical	0.1269	right
R_*but*	lexical	0.1267	right
R_def_art_cl1	reference	0.1236	left
L_*other*	reference	0.1188	left
r_NNP	surface	0.1185	left
R_wordsim_lesk	lexical	0.1173	unclear
m_*as*	surface	0.1165	right
r_*bring*	lexical	0.1148	left
L_*further*	reference	0.1135	right
m_*little*	surface	0.1134	left
L_disc_str	discourse	0.1133	depends on value
L_missing_mod	reference	0.1113	left
m_infinitive	syntactic	0.1081	unclear
l_*now*	surface	0.1039	right
m_*new*	surface	0.1035	left
L_*later*	reference	0.1034	left
m_*assume*	lexical	0.1001	right
m_*result* (NP)	lexical	0.1000	left
r_present	syntactic	0.0985	right
l_past	syntactic	0.0984	unclear
R_*later*	reference	0.0978	right
m_modal	syntactic	0.0962	unclear
r_*farmer*	surface	0.0958	left
l_VBZ	surface	0.0956	unclear
m_time_ref	lexical	0.0934	left
L_*less*	reference	0.0929	left
L_nr_def_art	reference	0.0919	unclear
L_anaphora	reference	0.0918	left
L_wordsim_lesk	lexical	0.0913	unclear
L_stem_overlap	lexical	0.0908	left
L_synt_sim	syntactic	0.0907	unclear
r_time_ref	lexical	0.0903	left
R_lemma_overlap	lexical	0.0900	right
R_synt_sim	syntactic	0.0893	right

F.6 50 best scoring features following ranks in all 4 algorithms

Rank	*Feature*	*Feature type*
1	R_pers_pr_cl1	reference
2	R_def_art_cl1	reference
3	R_cont_quotmarks	discourse
4	l_past	syntactic

Rank	*Feature*	*Feature type*
5	R_token_overlap	lexical
6	r_PRP	surface
7	r_time_ref	lexical
8	R_missing_mod	reference
9	l_present	syntactic
10	R_lemma_overlap	lexical
11	R_nr_pers_pr	reference
12	r_NNP	surface
13	R_wordsim_Lin	lexical
14	L_pers_pr_cl1	reference
15	r_*the*	surface
16	R_disc_str	discourse
17	m_*little*	surface
18	R_stem_overlap	lexical
19	m_NNP	surface
20	m_infinitive	syntactic
21	m_*to*	surface
22	L_synt_sim	syntactic
23	R_nr_def_art	reference
24	m_*as*	surface
25	R_anaphora	reference
26	R_*other*	reference
27	L_anaphora	reference
28	r_present	syntactic
29	l_VBZ	surface
30	L_*other*	reference
31	r_pos_in_par	discourse
32	m_pos_in_par	discourse
33	r_*it*	surface
34	R_*added*	reference
35	L_*less*	reference
36	R_nr_dem_pr	reference
37	m_gerund	syntactic
38	L_*further*	reference
39	L_wordsim_Lin	lexical
40	L_nr_pers_pr	reference
41	r_*farmer*	surface
42	l_infinitive	syntactic
43	L_missing_mod	reference
44	l_pos_in_text	discourse
45	m_pos_in_text	discourse
46	r_pos_in_text	discourse
47	R_*more*	reference
48	m_*a*	surface
49	r_modal	syntactic
50	R_*less*	reference