ABSTRACT

Automatic Discovery and Classification of Discourse Contingency Relations

Kam A. Woods

Mentor: C. Roxana Girju, Ph.D.

This thesis explores the use of Support Vector learning applied as part of a semi-automatic system for the discovery and classification of the discourse contingency relations *cause*, *concession*, *condition*, *purpose*, *reason*, and *result* at the discourse level in open text. The semantic spaces associated with these finely grained and highly ambiguous discourse relations are represented via a compact set of carefully selected syntactic, grammatical, and semantic features from a series of automatically parsed data sets. We perform a series of one-vs-one and one-vs-all style classification experiments on this data using the libSVM-2.5 machine learning tool with a Radial Basis Function kernel. This research has broad applications, including the deepening of inferential ability in Question Answering systems, and general improvements in text understanding.

Approved by the School of Engineering and Computer Science:

Donald L. Gaitros, Ph.D., Chairperson

Approved by the Thesis Committee:

C. Roxana Girju, Ph.D., Chairperson

David B. Sturgill, Ph.D.

Linda M. McManness, Ph.D.

Approved by the Graduate School:

J. Larry Lyon, Ph.D., Dean

Automatic Discovery and Classification of Discourse Contingency Relations

A Thesis Submitted to the Faculty of

Baylor University

in Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

By

Kam A. Woods

Waco, Texas

December 2004

TABLE OF CONTENTS

APPENDIX

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Radu Soricut and Daniel Marcu provide the SPADE tool for discourse parsing without which this work would not have been possible. I would like to thank Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin for providing the libSVM package, in addition to the community on the Kernel Machines list-serv for invaluable information and support. Finally, I would like to thank my thesis committee for their time and insightful commentary.

CHAPTER ONE

Introduction

1.1 Motivation

One of the long-term goals in current natural language processing research is the design and construction of domain-independent, adaptive systems capable of discourse interpretation in open text. Simply stated, the desired system would be capable, in an unsupervised setting, of extraction and interpretation of semantic context within a cohesive text span of arbitrary length. With such a system, we can envision sophisticated and subtle forms of analysis not possible with systems available today, which rely on limited statistical techniques, domain specific training schemes, or large databases of manually constructed world knowledge and rules for inference.

Discourse processing concerns the location and identification of semantic relations at the discourse level and the clauses or sentences associated with them. It is centered around the idea that the production and interpretation of phrases and utterances depends on the discourse context, and that the meaning associated with these phrases is directly related to coherence in the passage. Modern approaches to research in discourse utilize the fact that clauses, sentences, and larger text spans are not read and interpreted in an isolated context, but rather are highly interrelated (Mann and Thomson 1988).

Discourse processing, and more generally, the semantic interpretation of coherent text spans, is a complex task. It does not suffice to describe coherence relations between each pair of sentences, as sentence ordering in a coherent passage is often non-linear. Instead, we seek to reformulate the text as a hierarchical structure, with locally coherent pairs of nodes (extracted text spans) connected by some semantic relation. Frequently, the selection of such relations will rely on processing schemes

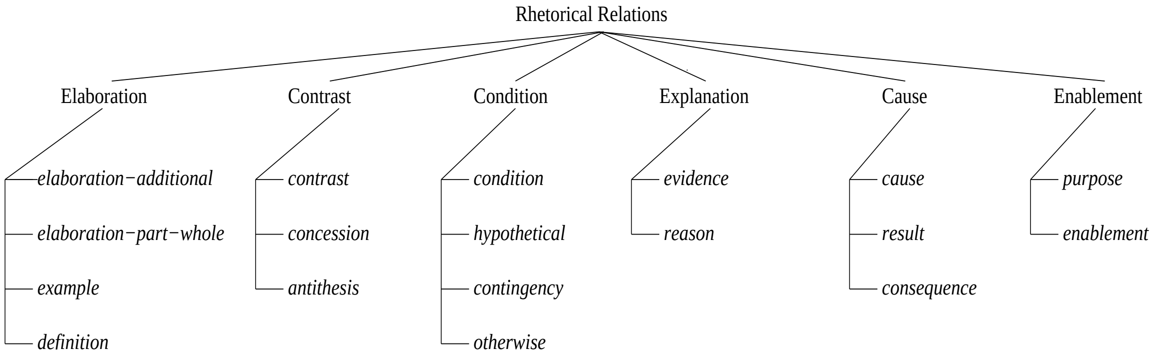


Figure 1.1. Some RST relation classes, according to Mann and Thomson

for particular syntactic constituents, such as in anaphoric or cataphoric phrases where a decision must be made regarding pronoun resolution.

The selection and organization of the set of discourse relations remains an area of active research in computational linguistics. One widely-recognized approach is that of Rhetorical Structure Theory, which describes a set of *rhetorical relations* that hold between text spans. An abbreviated set of the relations used in RST is given in Figure 1.1.

A discourse relation may be cued by a particular word or phrase, the presence of which is neither necessary nor sufficient in establishing coherence. Consider the following sentence: *Mary ate Anne's orange because she was hungry.* The cue *because* has been inserted to provide clarification of the explanation present in the passage (why Mary ate Anne's orange), but does not itself provide the semantic pairing. We can just as easily write: *Mary ate Anne's orange. She was hungry.*

The immediate interpretation is the same; that is, a coherent and cohesive passage in which the pronoun corefers with Mary. Likewise, we can write the sentence *Mary ate Anne's orange because she had a red car.* Here cohesion is maintained (due to pronoun coreference, although the reference is now ambiguous) but coherence is not. This simple example demonstrates why lexical and syntactic approaches to text analysis are freqently inadequate, even when the underlying semantic structure is unambiguous. The EXPLANATION relation in this example is encoded irrespective of the presence of the cue, and can be expressed in the following simple logical formula (Jurafsky and Martin 2000):

$$\forall e_i, e_j \; cause(e_i, e_j) \rightarrow Explanation(e_i, e_j) \tag{1.1}$$

This is, given events e_i and e_j, if e_i is the *cause* of e_j we can say that the EXPLANATION relation holds between the associated text spans.

Finally, the presence of a cue phrase may not necessarily constrain us to a single relation. As we begin to examine the semantic structure of the sentence in more detail, it may frequently be observed that a single cue appears in conjunction with multiple relations, introducing a further level of ambiguity.

While many theories of discourse structure provide sophisticated models for text analysis, the approaches used can differ significantly, particularly with respect to the decomposition and coverage of the semantic classes described. There is often a large degree of overlap in the relations considered, stemming from a shared recognition that classes such as CONDITION, ELABORATION, and EXPLANATION exist. However, we may find that in the application of two different theories, relation instances in open and unrestricted text will frequently be labeled differently depending on how a particular relation has been defined.

Systems for automatically assigning such labels are therefore limited, and the latest statistical and machine learning techniques provide only for the most shallow semantic approach to text analysis. This is in part due to the sheer complexity and breadth of the discourse relation class definitions, in part due to significant syntactic and semantic ambiguities inherent in open text, and in part due to the lack of algorithmic or heuristic methods for reliable logical inference. Such systems are capable, among other tasks, of recognizing and labeling with high accuracy limited subsets of ambiguously cued discourse relations (Marcu and Echihabi 2002), locating and annotating discourse connectives (cue words and phrases) and the arguments they link in syntactically and semantically annotated corpora (Miltsakaki 2002), probabilistically inferring single relation classes in large unannotated corpora (Lapata and Lascarides 2004), and performing sentence level discourse parsing on syntactically annotated corpora (Soricut and Marcu 2003).

In this thesis, we focus on the *contingency relations*, a set of six discourse relations encoded by adverbial clauses of contingency. These relations are Cause, Reason, Purpose, Result, Condition, and Concession. For our purposes, the *contingency relations* are superordinated by the semantic class Explanation, a hierarchical selection that differs significantly from that in theories such as RST, where these relations are split among a series of other classes. In fact, as we shall see later, the semantic label Contingency carries a separate meaning in RST. The justification for the grouping used here depends both on syntactic considerations (all six can be described in terms of an adverbial classification) and semantic analysis provided in Chapter 3.

Explanation discourse relations occupy a key role in the semantic structure of natural text. We observe them, for example, in arguments providing evidentiary support and descriptions of reason or purpose. As noted previously, they may be indicated by distinct cue words or phrases that link a main and subordinate argument within a sentence, or occur without explicit cues, particularly in extended or rhetorically complex text spans. Native speakers of English can readily identify such cues (when present) and the arguments they connect, and, with minimal training, are able to distinguish between relatively finely-grained classes of Explanation relations, including those examined in the experiments conducted here, the *contingency relations.*

From a computational perspective, advances in the construction of semantic and discourse annotated corpora have dramatically impacted the performance of statistical and machine learning models constructed for the automatic or semi-automatic classification of broad classes of semantic relations. Furthermore, recent work has shown that coarsely-grained discourse relations can be identified in massive, unannotated corpora using relatively simple lexical and syntactic features in a statistical classifier (Marcu and Echihabi 2002). As these relations are realized at finer levels of granularity, however, performance drops due to their semantic and syntactic similarities. More sophisticated mechanisms relying on semantically annotated corpora and revised feature sets constitute one potential solution to this problem.

Many important applications exist for systems capable of automatically identifying discourse relations in open text. These include Text Summarization, in which a document is compressed into a representation of logically ordered salient points, and Question Answering, which in many cases requires complex inferential processing. For example, traditional QA systems can readily process with a high degree of accuracy questions which require as an answer a single fact ("factoid"-based questions) such as *"When did the battle of Austerlitz take place?"*, and, to some degree, questions where extraction of the answer requires some level of semantic inference – inference achieved through the recognition of discourse relations. However, as these relations become more finely grained, as the ambiguity both in the relation classes and the phrases indicating those relations increases, performance in such systems typically drops dramatically. As an example, take the set of questions *"Why did X happen?"*, *"What was the purpose of X"?* and *"What are the effects of X?"*, classified respectively as the Explanation sub-relations CAUSE, PURPOSE, and RESULT.

We believe that the learning model presented in this paper may help to significantly improve the performance of such high level discourse processing systems by increasing the accuracy of discourse relation classification in cases where the relations are ambiguously cued or exhibit other forms of semantic and syntactic overlap. Our approach is novel, both in selection and use of particular features in the learning model, and in the examination of the *contingency relations*, which to our knowledge have not been addressed in other approaches.

1.2 Experimental Overview

We are interested in a generating a detailed account of the characteristics of the *contingency relations*, where our primary focus is the semantic similarities and differences among them. Consequently, we perform experiments aimed at automatically identifying and classifying contingency relations in text. To build the system, we train a set of Support Vector Machine classifiers using nine lexico-syntactic and semantic features.

The training and test examples are generated from the L.A. Times text collection, an unannotated (discounting HTML document tags) corpus drawn from a larger text collection used for a wide variety of text analysis tasks. The articles appearing in this collection date from 1989 to 1994. All examples are automatically tagged to provide parallel sets comprising syntactic, grammatical, and discourse parses, in addition to being manually tagged with argument boundaries.

Each sentence (or pair of sentences) extracted from the raw corpus encodes one of the six *contingency relations*, and is explicitly but ambiguously identified by a list of predefined cue phrases. Selection of the cue phrases is based on relative frequency in the raw text collection. As noted previously, we consider the set of *contingency relations* to be superordinated by the EXPLANATION semantic class.

Pairwise classification trials allow us to judge the relative performance of the features selected on both relations where there exist cross-relation cue-phrase ambiguities, and those where cue sets are disjoint. More generally, these trials provide insight into the semantic spaces occupied by the *contingency relations* and the effect of semantic overlap on classification accuracy. One-versus-all classification trials provide further insight into the performance of the system in instances where a single relation must be selected from a non-homogeneous group consisting of examples drawn from all remaining contingency relations, a task more representative of relation classification in open text.

Accuracy ranges from a low of 55.96% (Cause vs. Concession) to a high of 72.47% (Condition vs. Result) in the pairwise comparisons. All comparisons outperformed the respective baselines centered around 50% (baseline adjustments were performed due to differentials in the magnitudes of data extracted for the test sets).

1.3 Goals

Our primary goals are two-fold. First, to show that a carefully selected set of lexico-syntactic and semantic features can effectively be used to classify among a set of finely grained and highly ambiguous discourse relations, the contingency relations, when paired with a Support Vector Machine learning procedure. Second, to demonstrate that this classification can be performed even when the discourse context is inter-sentential – that is, between adjacent sentences.

The classification procedure presented here has been developed specifically for the improvement of semantic interpretation systems designed for tasks such as Text Summarization and Question Answering. Ultimately, the desired outcome is a system capable of locating and classifying semantic arguments in open texts of arbitrary length.

1.4 Thesis Outline

In Chapter 2, we discuss previous attempts to improve performance in similar classification problems through the use of lexico-syntactic and semantic information extracted from associated corpora. In addition, we revisit the representational schemas that have been used to encode this information, including but not limited to rhetorical structure theory and discourse annotated corpora, automatic semantic role labelling, approaches to discourse representation, grammar parsing tools, and statistical part-of-speech tagging.

In Chapter 3 we provide a detailed analysis of the class of contingency relations, the selection, content, and preprocessing of the selected text corpus, a brief introduction to supervised machine learninng, and a more specific account of the SVM learning model used in the experiments. In addition, the lexico-syntactic and semantic features extracted from the text are analyzed in detail.

Experimental results and additional procedural details are discussed in Chapter 4. We discuss the operation and tuning of the SVM, and compare the results from one-vs-one and one-vs-all style experimental trials.

Chapter 5 concludes the thesis, providing an overview of the relative success of the method and additional comments on the efficacy of learning models to this and more general tasks. Future work and potential applications of the procedures developed here are outlined.

Tables providing error analysis, trained parameter values, and other information are included for each experimental trial in Appendix A.

CHAPTER TWO

Related Work

The application of machine learning to semantic argument identification and classification is a relatively recent development, in part due to the prior lack of appropriately annotated corpora. In this chapter, we examine both the development of theories of semantic structure and the formalisms used to express them, and various applications of machine learning to related tasks.

2.1 Rhetorical Structure Theory

Rhetorical Structure Theory was developed by a research group headed by Bill Mann (Mann 1984), in response to a recognized lack of available theories for discourse structure and function - in particular for the task of computer generation of natural language text. RST is essentially a formalism that explains coherence in text. That is, given some set of text spans (in a typical case, adjacent sequences of words in a sentence) RST seeks to describe the relationship of one to the other in terms of a specific role.

Relations identified in RST can be mononuclear or multinuclear. In the former, there is a distinct "best" choice for the NUCLEUS in the relation. In the latter, there are multiple selection for the NUCLEUS role that can provide an adequate account of the associated semantic structure. The more peripheral relation is labelled the SATELLITE. Prior to the creation of a discourse structure using these relations, the text is separated (either manually or using a statistical procedure) into minimal text spans associated with a particular relation; these text spans are known as elementary discourse units, or EDUs. In some instances, this process is straightforward. For example, the ATTRIBUTION relation is frequently associated with attribution verbs, such as *said* and *stated*, or by phrases such as *according to*. However, syntactic ambiguities and exceptions occur frequently for many of the relations indicated below, and thus require more detailed treatments.

In the Discourse Tagging Reference Manual, Carlson and Marcu note that the 78 distinct relations recognized by RST can be partitioned into a compact framework of relations sharing semantic meaning. This list is given below:

Attribution: attribution, attribution-negative
Background: background, circumstance
Cause: cause, result, consequence
Comparison: comparison, preference, analogy, proportion
Condition: condition, hypothetical contingency, otherwise
Contrast: contrast, concession, antithesis
Elaboration: elaboration-additional, elaboration-general-specific, elaboration-part-whole, elaboration-process-step, elaboration-object-attribute, elaboration-set-member, example, definition
Enablement: purpose, enablement
Evaluation: evaluation, interpretation, conclusion, comment
Explanation: evidence, explanation-argumentative, reason
Joint: list, disjunction
Manner-Means: manner, means

As discourse structures are constructed hierarchically, any pair of text spans may serve as a single satellite or nucleus for some relation appearing further up in the discourse tree. Furthermore, a constituent EDU may be split by an embedded phrase (such as an appositive or parenthetical phrase) representing a separate EDU. Such constituents must be addressed using artificial relation tags in order to reconnect them under a single relation heading, further complicating the representational scheme.

In spite of these complications, and other known difficulties (Forbes 2001), (Joachims 1999), RST provides a relatively compact and well-defined set of rules for constructing shallow semantic parses of sentences.

2.2 Discourse Processing

In order to apply theories such as RST to practical tasks such as the automatic annotation of large corpora, we require both computational tools and a more complete account of the lexical and syntactic elements appearing in open text that cue for the relation sets given. In the following sections we describe one such tool and several related corpus studies.

2.2.1 Discourse Segmentation and Parsing

One approach to automated discourse processing involves the application of a series of probabilistic models to identify discourse units in individual sentences (discourse segmentation) and select the “best” discourse parse tree from the automatically generated discourse segmented lexicalized trees (discourse parsing) (Soricut and Marcu 2003). The model in this research is trained on a publicly available corpus consisting of approximately 7000 sentences drawn from 385 Wall Street Journal articles (RST-DT 2002). Every document in the corpus is associated with a manually-constructed discourse structure (in tree form) built in the style of Rhetorical Structure Theory (discussed in a later section).

The discourse segmenter adopts a statistical model that determines probabilities for the discourse boundary insertion at all possible locations in a sentence using both lexical (word-level) and syntactic (part-of-speech annotated syntactic trees) information derived from the RST Discourse Treebank (RST-DT 2002). For any given possible boundary, the segmenter subsequently inserts a boundary given a sufficiently high probability.

The initial corpus used to build the RST-DT corpus comprises 110 different rhetorical relations. For the SPADE tool (a series of scripts implementing both the automatic discourse segmenter and the discourse parser), Soricut and Marcu compressed these relations according to work performed by (Carlson et al., 2003) into a set of 18 superordinating labels. The RST-DT corpus consists of 385 Wall Street Journal articles drawn from the Penn Treebank, an ongoing project to provide a massive, wide-coverage text corpus manually annotated with syntactic trees. The use of such corpora is desirable, as they provide a "gold standard" benchmark for tagging accuracy (in this case, discourse structures in the same form as those generated by the software), and a common reference point for performance comparison between systems.

The discourse segmenter implemented in SPADE processes the Charniak parsed input text into a series of non-overlapping segments called *elementary discourse units*, or EDUs. These are in *all* cases categorized either as the NUCLEUS or SATELLITE of some rhetorical relation identified in the full sentence. Generally, the NUCLEUS consists of the central idea or statement, while the SATELLITE(S) provide peripheral information (Jurafsky and Martin 2000). This is a direct consequence of the asymmetry observed in most discourse relations (for example, the subordination of one argument by another).

It should be noted that in this automatic process, the rhetorical relation that SPADE identifies will in many cases not match in class or composition that we are interested in, as the class of *contingency relations* is split among a number of superordinating relations in RST. Nevertheless, the discourse structures produced by SPADE can provide valuable insight into the semantic organization of the text under consideration.

2.2.2 Discourse Connectives

(Creswell 2004) describe a corpus study in which discourse connectives are located and annotated for the purpose of developing algorithms to resolve anaphoric expressions (pronoun coreference). In this work, the authors explore the difference between *structural* and *discourse* connectives, using as a theoretical basis the Discourse Lexicalized Tree Adjoining Grammar, or DLTAG (Forbes 2001). The DLTAG itself is a formalism in which tree structures for predicate-argument dependancies are recursively modified by auxiliary trees encoding adjuncts to the initial structure.

The authors derive a series of heuristics for improving the decisions made by annotators in locating discourse connectives and labeling associated arguments. In doing so, they note the broad syntactic variety of potential arguments for a set of cues that overlaps with the ones examined in the experiments carried out in this work.

The authors note that while discourse connectives are easily identified, the actual associated discourse structure and relation involved in any particular case is not completely addressed in current linguistic theory, let alone related computational approaches. Expanded analysis of large-scale corpora is therefore essential to realistic progress in this area.

A method for the annotation of both discourse connectives and their arguments is presented in (Miltsakaki 2002), using as a theoretical basis previous work demonstrating the effectiveness of integrating sentence level structures with discourse level structures using tree-adjoining grammars such as DLTAG. They note that the use of such grammars allows for the separation of the compositional elements of discourse meaning (that is, those described by the trees constructed) from the non-compositional elements (such as those that are realized through inferential analysis and anaphora resolution).

This provides an elegant and effective method for attacking the problem of discourse structure automatically. It avoids in part the problems associated with making an initial ('arbitrary' according to the authors) selection of discourse relations to be examined, instead beginning with the task of locating discourse connectives and their associated arguments. This reasoning is mirrored in the approach presented in this thesis, where sentences containing the requisite discourse connectives are extracted and subsequently annotated with syntactic, semantic, and grammatical information.

2.2.3 Statistical Approaches to Role Labeling

A series of statistical classifiers are used in *Automatic Labeling of Semantic Roles* to identify semantic roles filled by constituents of a sentence (Gildea and Jurafsky 2000). The experiments include both identification of semantic roles in previously segmented sentences, and a more complex task in which the sentence is automatically segmented and subsequently analyzed for semantic roles.

Gildea and Jurafsky address a number of points directly related to the work presented here. First, they make a convincing case for the use of shallow semantic interpretation of sentences in the development of systems such as those used for Question Answering and information extraction.

A series of *target words* are identified as having meaning associated with the desired semantic frame, and for each of which examples from the corpus are drawn which represent a full range of existing lexico-syntactic patterns. Thus, as in the work presented here, the sentences extracted for the purpose of training and testing the system *do not* comprise a statistically representative spread of those sentences in open text, but rather provide a complete account of the specific relations that are of interest in the associated classification task.

The authors present a detailed treatment of features selected for use in their experimental systems, including the *Parse Tree Path* feature, defined as the path from the target word to the desired sentence constituent. As in the work presented here, this path, composed of a sequence of part-of-speech values, is treated as an atomic value.

In addition to providing a solid theoretical basis for the development of the features used in our system, the research of Gildea and Jurafsky suggests that data-driven techniques for semantic interpretation constitute an effective approach that may aid in generalizing systems that work with “shallow” semantic parses in limited domains to more sophisticated tasks. Prior to such methods, semantic parsing systems relied heavily on complex human-generated grammars linking semantic roles to specific syntactic structures. Such grammars are problematic, particularly in cases where, for example, ambiguity of coreference between pronouns and preceding noun phrases in the examined discourse results in the failure of the system to establish proper coherence in the passage. Data-driven approaches avoid the the problems associated with developing grammars that minimize such difficulties, while at the same time allowing us to make use of the connection between lexico-syntactic realizations and underlying semantic roles.

An unsupervised approach using Naive Bayesian classifiers to recognize the discourse relations Contrast, Explanation-Evidence, Condition, and Elaboration is presented in (Marcu and Echihabi 2002). This work is interesting due to the fact that these relations are located between arbitrary spans of text using a system trained on extremely large corpora of automatically extracted examples, a percentage of which are not explicitly marked by cue phrases.

This work serves to establish the validity of applying machine learning methods to disambiguating between semantic classes with common cues and syntactic structures, in addition to identifying performance increases when training on massive amounts of data.

CHAPTER THREE

Design and Implementation

3.1 Contingency Relations

Contingency relations are classified using six categories (Quirk 1985). These are Cause, Reason, Purpose, Result, Condition, and Concession. In the following sections, we define each relation and provide a selection of frequently appearing cue phrases associated with that relation. A number of these cue phrases are ambiguous; that is, they encode more than one relation, or encode semantic structured outside of the relations considered. The inclusion of ambiguous cue phrases is important in determining the relative effectiveness of the machine learning component in the experiments described here, as the distinction between examples appearing in more than one category may frequently be more subtle than those which are unambiguous.

Table 3.1 provides an overview of cue phrase distribution in the combined training and test corpora. While the cue phrases are predominantly concentrated within the relation class(es) for which they have been explicitly extracted, they may also appear *in addition to* another cue within corpus examples associated with other classes. Additionally, the cue distribution represented here does not necessarily reflect the distribution we might expect to find in open text, as this corpus was constructed to include *only* those sentences that contain a particular cue (or precede a sentence containing a particular cue).

3.1.1 Cause

Cause is "concerned with causation and motivation seen as established with some objectivity (Quirk 1985)." This is exemplified in the following example: *She*

Table 3.1. Cue distribution by semantic class

	Number of cues (Percentile)					
#	Cause	Concession	Condition	Purpose	Reason	Result
[Aa]ccordingly	93 (97%)	1 (1%)	1 (1%)	1 (1%)	0	0
[Bb]ecause	117 (43%)	32 (12%)	6 (2%)	4 (1%)	106 (39%)	4/(1%)
[Cc]onsequently	100 (98%)	1 (1%)	0	0	0	1 (1%)
[Tt]herefore	102 (97%)	1 (1%)	0	1 (1%)	1 (1%)	0
[Tt]hus	103 (95%)	1 (1%)	2 (2%)	2 (2%)	0	0
[Aa]lthough	14 (10%)	109 (81%)	5 (4%)	5 (4%)	1 (1%)	1 (1%)
[Ee]ven though	1 (1%)	103 (96%)	1 (1%)	1 (1%)	1 (1%)	0
[Hh]owever	21 (14%)	116 (65%)	17 (9%)	8 (4%)	14 (8%)	2 (1%)
[Nn]evertheless	0	101 (100%)	0	0	0	0
[Nn]onetheless	0	100 (98%)	1 (1%)	0	1 (1%)	0
[Aa]s long as	1 (1%)	3 (3%)	102 (95%)	0	0	1 (1%)
[Aa]ssuming that	0	0	69 (100%)	0	0	0
[Ii]n the event that	0	0	47 (100%)	0	0	0
[Pp]rovided that	0	0	54 (98%)	1 (2%)	0	0
[Ss]o long as	0	0	104 (100%)	0	0	0
[Ii]n order that	0	0	0	18 (100%)	0	0
[Ii]n order to	5 (4%)	2 (2%)	0	101 (92%)	0	2 (2%)
[Ll]est	4 (3%)	6 (4%)	4 (3%)	130 (87%)	3 (2%)	1 (1%)
[Ss]o as to	0	0	0	101 (100%)	0	0
[Ss]o that	2 (1%)	1 (0%)	1 (0%)	101 (49%)	0	101 (49%)
[Aa]s	7 (5%)	10 (7%)	6 (4%)	12 (8%)	107 (74%)	3 (2%)
[Ff]or	6 (5%)	11 (8%)	4 (3%)	5 (4%)	106 (79%)	2 (1%)
[Ss]ince	5 (4%)	13 (9%)	5 (4%)	4 (3%)	109 (79%)	2 (1)
[Ss]o	33 (14%)	36 (16%)	17 (7%)	18 (8%)	27 (12%)	100 (43%)

died because she had cancer. Causal connectives (discounting causative verbs) may be split into the following categories: adverbial causal links, prepositional causal links, subordination causal links, and clause-integrated links.

Certain categories, such as prepositional causal links (cued by phrases such as *because of* and *due to*), are generally used to link a noun phrase with a clause, or to link two noun phrases in an apposition (Girju and Moldovan 2002), rather than the main-subordinate argument pairs that we are interested in here. Others, such as adverbial causal links and clause- integrated links, may be associated with either the CAUSE or REASON semantic classes, depending on use context (Quirk 1985). CAUSE correlatives include *because*, *thus*, *therefore*, *seeing that*, *accordingly*, and *consequently.*

Cue phrases for the work presented here have been selected for relative frequency in the LA Times corpus, specifically by examining a randomly selected 32,500 sentence block for instances of a particular word/phrase. The five selected cues are: *because*, *thus*, *therefore*, *accordingly*, and *consequently.*

3.1.2 Reason

According to (Quirk 1985), Reason is a superordinating term for the following four categories: cause and effect (expressing the perception of an inherent objective connection in the real world), reason and consequence (expressing the speaker's inference of a connection), motivation and result (expressing the intention of an animate being that has a subsequent result), and circumstance and consequence (combining reason with a condition that is assumed to be fulfilled or is about to be fulfilled).

Reason cue phrases include *because*, *since*, *as*, and *for*. Note that in the majority of cases, Reason involves a relatively personal and subjective assessment of a situation. This is demonstrated in the sentence *"The flowers are growing well because I sprayed them."* The grammatical and syntactic structure of this sentence is virtually identical to that of the example given for the Cause relation; the difference is that this is semantically understood not to be a description of fact, but of opinion. The definition of Reason given here is particular to (Quirk 1985), and serves to distinguish at a finer granularity examples of this relation from those which are subsumed by Purpose under other definitions. We retain only this definition as it provides a basis upon which to distinguish the relative success or failure of different feature vectors in classifying instances from the test corpus that exhibit such semantic subtlety.

3.1.3 Purpose

Purpose clauses are adjunct (adverbial and attributive), more often infinitival than finite, and typically indicate a result that has yet to be achieved. As a consequence of this, Purpose clauses overlap with those of Result both in meaning and in subordinators.

Purpose clauses in the infinitive form are introduced by the cue phrases *to*, *so as to*, and *in order to*. Finite clause patterns include *so that*, *so*, *in order that*.

The more formal *lest* is also considered here, and is considered to be sufficiently representative of the constructions within which other unambiguous negative patterns appear. The following finite clause example illustrates the Purpose relation: *The school closes early so that the children can come home.*

The cue phrases selected for Purpose are as follows: *so that*, *in order that*, *in order to*, *so as to*, and *lest.* The *so that* cue overlaps with an identical cue appearing in the Result relation.

3.1.4 Result

Result clauses, as stated above, differ from those of purpose primarily in that they indicate a result that has already been achieved. Result clauses are introduced by the subordinators *so that* and *so.* It should be clear that the subtlety of the distinction between result and purpose, in terms of applying a machine learning algorithm, can provide valuable information about the usefulness of various features in the feature vector. An example using an identical cue to that provided in the example provided for Purpose is as follows: *We paid him immediately, so that he left contented.*

3.1.5 Condition

Condition clauses indicate a potential result as a consequence of a particular condition. Conditional clauses may be direct (*"If you put the baby down, she'll scream"*) or indirect (*"She's far too polite, if I may say so"*). There are a large number of conditional subordinators, including *as long as*, *so long as*, *assuming (that)*, *given (that*, *in case*, *in the event that*, *just so (that)*, *provided (that)*, and *supposing (that).*

Conditional clauses may also be introduced by temporal subordinators; in certain cases these overlap with the regular subordinators. Such instances include *before*, *as long as*, *when*, *whenever*, and *once*. For example, *He will leave as long as she does.* The cue phrases selected for this series of experiments are as follows: *as long as*, *assuming (that)*, *provided (that)*, *in the event that*, and *so long as.*

3.1.6 Concession

Concession is essentially the inverse of *condition*; it indicates circumstances in which a result would ensue irrespective of the concessive clause. This is illustrated succinctly in the following example: *"It was an exciting game, although no goals were scored"*.

Concession clauses, as with condition clauses, are introduced by a large number of cue phrases. These include *although*, *(even) though*, *while*, *granted (that)*, *even if*, *yet*, *still*, *however*, *nevertheless*, *nonetheless*, *notwithstanding*, *anyway*, *anyhow.*

The cue phrases selected for this series of experiments is as follows: *although*, *even though*, *nevertheless*, *nonetheless*, and *however.* It is important to note that while certain encodings of these cues are ambiguous, the ambiguity results from examples *outside* the set of contingency relations. That is, ambiguity in Concession never results from overlap with *condition*, for example.

3.2 The Corpus

We have assembled a corpus from the L.A. Times text collection. Stripped of HTML tags, header information, tables, and figure references, the corpus contains approximately 3.4 million sentences. For simpler handling, the corpus was split into 96 files, each consisting of 35,000 sentences. A sentence splitter (implementing a two-phase process for locating non-sentence boundary punctuation and subsequently writing out corrected paragraph splits) was used to rewrite the contents of these files for further processing, tagging sentence boundaries and discarding overtly malformed examples.

A simple regular expression based Perl script was used to extract all sentences containing or immediately following a particular discourse connective from the entire data set, yielding approximately 210,000 sentences for the full discourse connectives set.

From this set, we manually selected and annotated approximately 100 sentences for each predefined cue phrase encoding each of the six contingency relations. For certain cues, fewer than 100 examples were present in the corpus, as can be seen in Table 3.1.

Sentence selection was performed using contextual information within and prior to each sentence according to the guidelines from (Quirk 1985) outlined in the previous section. In some cases, the candidates for selection were obvious, particularly for cue phrases specific to a single contingency relation. In other cases, the selection was more difficult. For example, when selecting examples of the *because* cue shared by CAUSE and REASON, a sentence such as "*I had to stay over an additional night because America West had overbooked the return by six seats*" was classified as CAUSE due to the fact that the author was providing an objective assessment of the situation, while a sentence such as "*I believe in a strong country because people mistake gentility for weakness*" was classified as REASON because of the inherent subjectivity present in statements of *belief*.

The argument boundary annotations were performed based on guidelines provided for the RST-DT corpus (Carlson et al. 2003). These are discussed in further detail in Section 3.4.1. As this series of experiments is also concerned with relations spanning an inter-sentential context, sentences beginning with a particular cue phrase were grouped with the preceding sentence, and a *context tag* was added to ensure separation of each relation instance.

All 2,600 sentences were automatically parsed into syntactic trees (Charniak 1997). Subsequently, grammatical roles were inserted into these trees using the automatic grammatical role detector GRD. Additionally, sentence level discourse parse trees were constructed using SPADE (Soricut and Marcu 2003). These tools are described in Section 3.4.2.

To aid in feature extraction, the corpus was subsequently converted into an XML representation. As a preliminary step, all non-XML compliant tags were converted via a direct mapping scheme; in all cases, these conversions maintain a format similar enough to the original versions to be identified in a manual inspection. While various standards have been developed for the purpose of conversion and manipulation of corpora in XML, a number of the tools used in the procedure presented here do not conform to a general XML-compliant format.

As a final step, the pairs of contextually linked sentences were attached at a new head node inserted locally into the XML file. This was not intended as an explicit representation of the actual semantic link between these sentences; indeed, it is the most naive form of attachment possible. Rather, it was added for the purpose of linking those sentences in a way which did not alter the existing structures. This allows for some flexibility in later processing of paths to and between arguments in an inter-sentential context.

This attachment step may in future revisions be replaced by a semantically and syntactically rigorous tree-adjoining procedure. Such procedures are based on formalism known as a Tree Adjoining Grammar, or in cases where lexical tags are present, a lexicalized Tree Adjoining Grammar. Recently, the theory has been extended to discourse parsing with discourse lexicalized Tree Adjoining Grammars (Forbes 2001). The authors claim that such a system, which represents lexicalized elements with respect to both the source sentence and the extracted discourse structure, can more completely describe the contribution of such elements to both the underlying semantics and the syntactic realization. The construction of systems and concepts such as D-LTAG have been aided by on-going research initiatives such as XTAG (http://www.cis.upenn.edu/˜xtag/home.html), an attempt to provide a wide-coverage grammar for the English language using a lexicalized Tree Adjoining Grammar.

3.3 Supervised Machine Learning

Machine learning experiments may be categorized according to the mechanism via which the learned model is acquired. Learning procedures are referred to as *supervised*, *unsupervised*, or *semi-supervised* depending on this mechanism. A *supervised* approach is used in the experiments performed in this work. Briefly, *supervised learning* procedures require examples of both inputs and outputs to generate a learned model. In the problem of classifying *contingency relations*, the inputs consist of 9–element vectors of syntactic and semantic features generated from sentences belonging to these relation classes, and the outputs are simply labels indicating the relation encoded by a particular sentence. The learned model is constructed based on patterns of association between particular feature values and the six labeled classes. We use a supervised approach because these six classes (*cause*, *concession*, *condition*, *purpose*, *reason*, and *result*) are known, and fully represent the superordinating class *contingency*.

Following construction of the learned model (the *training* procedure), the system is tested on unseen data instances in which inputs but no outputs are provided. The performance of the system (in terms of accuracy, precision, recall, and various error measures) may then be computed based on known correct outputs for unseen instances. One disadvantage of this method is that known outputs (in our case, semantic class labels) must be generated without error for all of the training and test data prior to applying the machine learning procedure. For large corpora this can be a difficult and time-consuming task, and is typically performed by a group of independant annotators.

Conversely, in an unsupervised setting, outputs are not provided and the learning model is acquired purely based on patterns appearing in the input. Such an approach is inappropriate for our task, as we wish to locate patterns specific to the semantic relations being considered and not, for example, patterns of syntactic similarity that would cause examples from multiple relation categories to be grouped together. Were we not fully cognizant of the six *contingency relation* classes, we could adopt a semi-supervised approach in which the pattern classification is not forced, providing an opportunity for the location of more legitimate patterns (in our case, semantic class groupings).

3.4 Support Vector Machines

3.4.1 Background

The theoretical basis for Support Vector learning is outlined by Vladimir Vapnik in his seminal work (Vapnik 2000). This work appeared as a response to difficulties encountered in processing high-dimensional data efficiently in various classification problems. Support Vector methods have recently been adopted in the construction of classifiers for a wide variety of automatic text processing systems.

The popularity of Support Vector Machines has increased primarily due to the fact that they are universal classifiers and can build classification models efficiently irrespective of the dimensionality of the feature space, which may be extremely large when attempting to create, for example, systems for lexical classification. Additionally, while the most basic SVMs learn linear threshold functions corresponding to linear classifiers, they can be adapted via the insertion of a kernel function to learn a variety of other classifiers. The most popular of these are polynomial classifiers, radial basis function networks, and three-layer sigmoid neural nets. Most SVM implementations also make provisions for the construction of user-defined kernel functions for specialized tasks such as the manipulation of string subsequences (Lodhi 2002).

SVMs have recently been used in combination with an active learning procedure to perform several text classification tasks (Tong and Koller 2000). As is generally the case, the objective of this work is the creation of a classifier that performs well on unseen future instances. The distinguishing feature of this work, however, is the confirming evidence that active learning with SVMs can reduce the need for labeled training instances, in many cases by an order of magnitude with no reduction in classification performance.

A method for domain independent shallow semantic role parsing via a machine learning algorithm based on the SVM core is presented in (Pradhan 2003). They report a precision and recall of 84% and 75% respectively on assigning semantic labels to the PropBank corpus. The system presented by these authors uses a series of features identical to those introduced previously by Gildea and Jurafsky: Predicate, Path, Phrase Type, Position, Voice, Head-Word, and Sub-Categorization. All of these features (except for the predicate) are extracted from a syntactic parse of the sentence, both automatically generated using the Charniak parser and human-corrected. The experiments carried out in this research concern both argument identification and argument classification. For the classification procedure, the authors note that head word and predicate were the most salient features (Pradhan et al. (2004)).

SVMs can be applied to a wide variety of of generalized data classification problems (Chang and Lin 2001). The authors provide a simple overview of problems associated with kernel selection, data scaling, cross-validation, grid-search for parameter estimation when using Radial Basis Function kernels, and outline a proposed procedure for using the libSVM-2.5 software on real-world problems.

3.4.2 Learning Model

The simplest SVM implementation learns a linear threshold function (a linear classifier). However, the practicality of the SV algorithm for a wide range of problem sizes arises from the fact that it can use kernel functions which enable us to perform all necessary computations in the input space even though all data is transposed into a high-dimensional feature space via a non-linear mapping. This is a remarkable property, since independence of the learning procedure from the dimensionality of the feature space means that the system generalizes even when a large number of features are present, something that is not true of statistical procedures such as Naive Bayesian classifiers or decision-tree learners.

In order to build a learning model, salient patterns or *features* are extracted from the training data for each instance. Each set of patterns is associated with a particular classifying label that consititutes the desired output. As an example, our system extracts 9 features from each sentence in the corpus (described in 3.5.3) and associates that feature set with the appropriate semantic label.

In general, given some N-dimensional set of patterns x_i and associated classifying labels y_i (where $y_i \in \{+1, -1\}$ for binary classification problems), that is,

$$(\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_l) \in R^N \times \{\pm 1\} \tag{3.1}$$

we wish to describe a function $f : R^N \rightarrow \{\pm 1\}$ that will assign the correct class label y to some unseen vector $\mathbf{x}$.

Support Vector Machines map the original data into some high-dimensional feature space via the nonlinear map $\Phi : R^N \rightarrow F$ and evaluate the dot product

$$K(x, y) = (\Phi(x) \cdot \Phi(y)) \tag{3.2}$$

that corresponds to the computation of the following linear decision function:

$$f(\mathbf{x}) = sign((\mathbf{w} \cdot \mathbf{x}) + b) \tag{3.3}$$

In this space, an *optimal separating hyperplane* is constructed to separate the data. In the simplest case, where the data is linearly separable, the construction of this hyperplane is as follows. Given our N-dimensional set of patterns as above, we can describe a separating hyperplane of the form:

$$(w \cdot x) - b = 0, \quad w \in R^N \tag{3.4}$$

where $(w \cdot x)$ indicates the inner product between vectors w and x, and w is a vector of weight parameters.

The separating hyperplane is *optimal* if it bisects the space containing the training vectors without error and with the distance to the closest vector (the margin of separation) maximized. The optimal hyperplane is unique, and in the case of completely separable data may be extracted by solving the quadratic programming problem

$$W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{3.5}$$

whose solution takes the form of the weight vector w, a linear combination of the vectors in the training set:

$$w = \sum_{i=1}^{l} y_i \alpha_i^0 x_i, \quad \alpha_i^0 \geq 0 \tag{3.6}$$

In the case of binary classification, the margin of separation between the closest vectors can be maximized subject to the constraints $(w \cdot x) - b \leq -1$ for $y_i = -1$, and $(w \cdot x) - b \geq 1$ for $y_i = 1$. These can be derived in any case by simply rescaling w and the factor b such that the margin (equal on both sides) becomes 1. It can be shown that only those vectors that satisfy these inequalities (Vapnik, 1999) can have nonzero coefficients of α_i^0 in the expansion noted above. These are the *support vectors*, and by definition provide all information required to address the classification problem. See Figure 3.1 for a graphical representation of the optimal hyperplane in a simple linear classifier.

Linear classifiers are frequently inadequate, even for very simple classification tasks. Take as an example a 2-dimensional plane with axes x_1 and x_2, in which

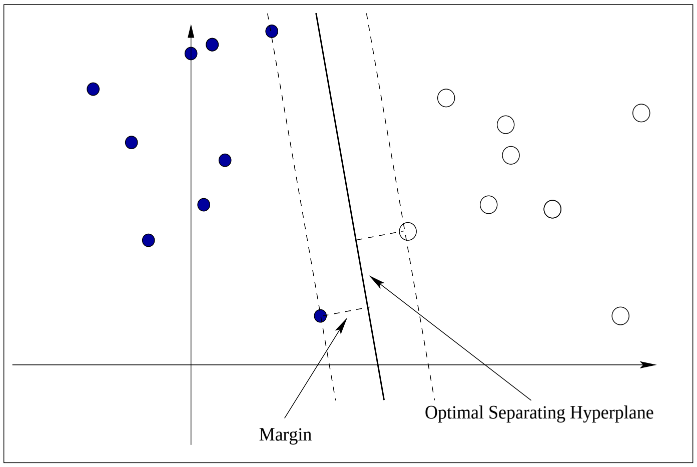


Figure 3.1: Optimal Separating Hyperplane (Linear SVM). The margin indicated represents the maximal margin of separation between the hyperplane and those support vectors (flattened to 2-dimensions here) with $\alpha = 0$.

we wish to isolate the points satisfying $x_1^2 + x_2^2 \leq 1$. As the separation boundary is circular, a linear classifier does not suffice. In order to address cases in which the data is linearly non-separable, (Vapnik 2000) provides a generalized form of the optimal hyperplane, uniquely determined by the vector w corresponding to the solution of the optimization problem

$$\Phi(w, \xi) = \frac{1}{2}(w \cdot w) + C\left(\sum_{i=1}^{l} \xi_i\right) \tag{3.7}$$

where C is a penalty parameter on the error term introduced through the non-negative variables $\xi_i \geq 0$. These variables modify the previous constraints used to determine the maximum margin of separation so that they now read $(w \cdot x) - b \leq -1 - \xi$ for $y_i = -1$, and $(w \cdot x) - b \geq 1 - \xi$ for $y_i = 1$ (where each y_i is an assigned label in our binary classification).

Recalling from before the basic form of the linear decision function (Equation 3.3), we can describe a set of decision functions that are both identical in function to those linear functions in the high-dimensional feature space and additionally fulfill our requirement for the location of a non-linear classification boundary in the input space. For each training vector $\mathbf{x_i}$, we substitute the left hand side of Equation 3.2

into our decision function to obtain the most general form associated with Support Vector classifiers in terms of some *kernel function* K:

$$f(x) = sign \sum_{i=1}^{n} y_i \alpha_i K(x, x_i) \tag{3.8}$$

For this series of experiments, we use a radial basis function (RBF) kernel $K = exp\{-\gamma|x-x_i|^2\}$, where C (a penalty parameter for the error term) and γ are typically tuned via cross-validation to the problem instance. In classical RBF machines (such as the type used here) determining the values of γ and C are based on heuristics; that is, they do not generalize to well-known problem instances, but are generated as needed (Vapnik, 1999).

The libSVM package includes a Python script, *grid.py*, to automatically determine the most appropriate values of C and γ for a particular problem instance. In the experiments performed here, it executes a 10-fold cross-validation for parameter estimation given a particular range. As per (Hsu, 2002), this method divides the training data into 10 subsets of equal size, training each of the 10 trained classifiers on the remaining 9 sets using all (C,γ) pairs in the range specified. It should be emphasized that this technique produces a "best-estimate", and does not always produce parameter values that maximize classification accuracy.

A graphical plot of cross-validation accuracy is constructed incrementally using training data during each parameter tuning run. Examples for the 9 feature, manual Argument Boundary Detection, Cause vs. Concession and the 9 feature, manual Argument Boundary Detection, Cause vs. All data sets are included in Figures 3.2 and 3.3.

In order to allow for salient analysis of the results, we performed a series of pairwise classification experiments in which all 15 possible pairings within the 6 classes were used to train a particular instance. All text data was mapped to numeric data (a constraint of the libSVM-2.5 implementation) via a hashing scheme in which text

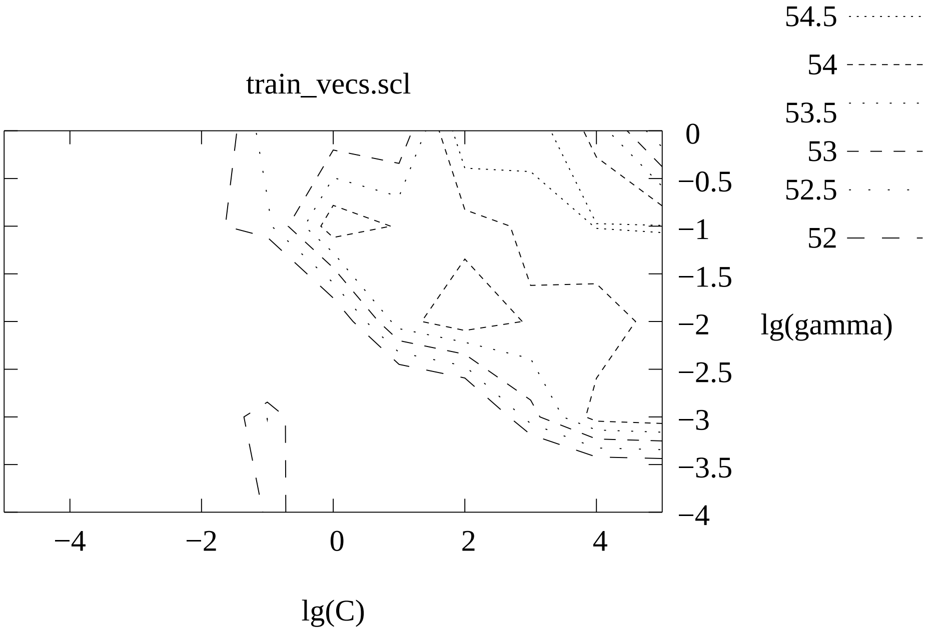


Figure 3.2: Grid training visualization (One-Vs-One): Dashed contours in this plot describe areas of cross-validation accuracy corresponding to a range of (C,γ) pairs.

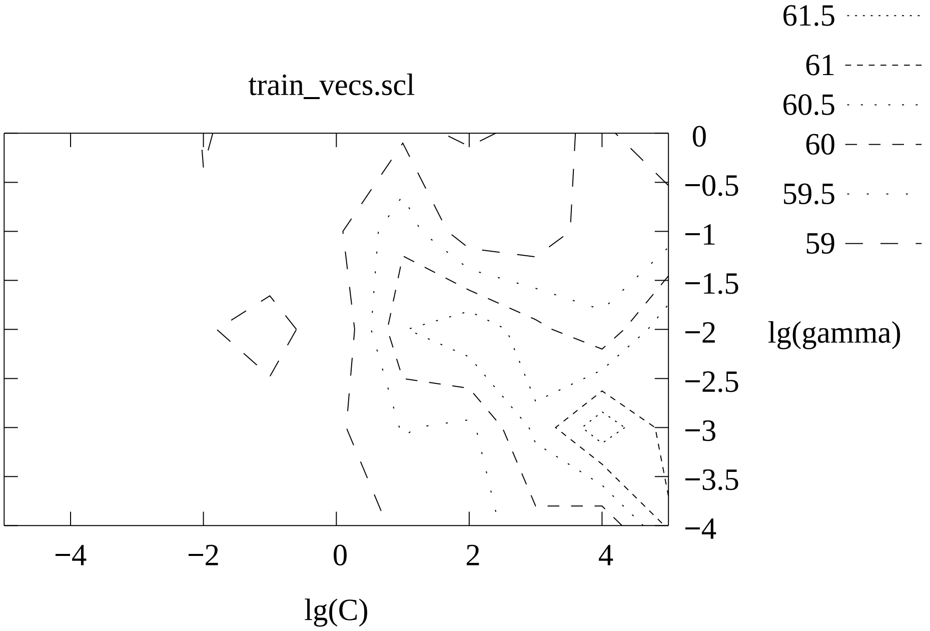


Figure 3.3. Grid training visualization (One-Vs-All)

strings corresponding to unique feature values were assigned real values and hashed into one of nine tables (one hash table for each feature). It is important to note that while more sophisticated methods exist for numerical feature representation (gray-code schemes, or more generally using n numbers to represent n-category features (Chang and Lin 2001)), the associated advantages due to numeric stability are only seen when the category magnitude is sufficiently low.

Finally, all data was scaled to a range from -1 to 1, both in order to avoid numeric difficulties due to over- and under-flow, and most critically to avoid biases due to groups of feature values located in larger numeric ranges.

The libSVM-2.5 machine learning tool, referenced above, was used in these experiments. It is freely available from *http://www.csie.ntu.edu.tw/˜cjlin/libsvm/*, and includes implementations of a variety of kernel functions.

3.5 Implementation

Machine learning approaches to discourse relation recognition and classification typically take advantage of large semantically annotated corpora such as FrameNet, a semantically annotated corpus based on the 100-million word British National Corpus, PropBank, a corpus annotated for basic semantic propositions, or the RST-DT corpus discussed in Chapter 2. In the procedure presented here, we show that a small group of carefully selected lexico-syntactic and semantic features can be applied to classify the set of contingency relations in a compact corpus automatically annotated with syntactic and semantic structures. This follows from the fact that while syntactic realizations are linked to underlying semantic arguments (the subject of *linking theory*), the syntactic information alone frequently cannot be used to identify the presence of a particular relation.

Since a number of the features rely directly on the position and boundaries of the main and subordinate arguments linked by the existing cue, we initially developed an automatic method for the detection of arguments and argument boundaries. An informal overview of the corpus suggested that the two most frequent argument patterns arising in the contingency relations are [(SUBORD)-(CUE)-(MAIN)] and [(CUE)-(MAIN)-(SUBORD)]. We used an heuristic that initially assumes canonical form, recursively parsing through XML subtrees until the maximal VP on either side of the cue word or phrase corresponding to the subordinate and main arguments is found. If a VP was is found for the subordinate argument, the remainder of the tree following the main argument is traversed to locate another VP.

During early trials of the feature extraction system, it became apparent that this heuristic was prone to error, particularly in cases where complex argument relationships resulted in syntactic overlap not appropriately addressed in the extraction. Additionally, the heuristic had no method for validating argument order, and forced the introduction of values to flag instances where no argument was found.

While an additional machine learning procedure for the location of such boundaries was the logical choice for solving this problem in a general manner, time constraints did not permit such an endeavor. Consequently, the entire set of extracted sentences was tagged with argument boundaries by hand. In particular, this resolved difficulties that were encountered with the automatic boundary detector in handling non-canonical sentences or sentence pairs marked by constructs such as appositions and anaphora.

3.5.1 Argument Boundary Detection

While multiple annotators were not available to tag argument boundaries for this data set, rules for tagging argument boundaries were developed for use during annotation. The theoretical basis for most of these rules is provided by examination of the Discourse Tagging Reference Manual provided by Carlson and Marcu. Much of the discussion of syntactic phenomena with relation to identifying and extracting elementary discourse units is relevant to the task of extracting arguments such as those being examined here; indeed, virtually all of our examples fall into a single syntactic device class, that of "Subordinate Clause with Discourse Cue" (DTRM 1999)

Phrases that (for our purposes) are semantically empty were excluded from the arguments. These included introductory comments such as *"In fact..."* and *"Remarkably..."* and any type of attribution such as *"..., said Mr. Doe"*. Attributions preceding or following the main argument body may legitimately be discarded as they will always be labelled as such in the discourse structures output by SPADE.

Phrase structures following a coordinating term such as *"and"* were excluded from the argument boundaries unless the remainder of the phrase appeared to corefer strongly with the preceding argument. An example of such an argument structure appearing in the extracted corpus follows:

> It is highly improbable that the policies towards unification will, being essentially nonpartisan, vary on either side of the dividing line. Therefore, appeals for gradualism are curiously irrelevant, and assurances to the Soviet Union that no effort will be made to accelerate the process could be dangerous.

In certain cases, the argument structure is implicit, and is not adequately captured by our tagging procedure. Consider the sentence "Museum officials said the stolen artifacts are irreplaceable and therefore priceless, but would be difficult to sell." 'Priceless' in the second argument is understood to stand for 'the stolen artifacts were priceless', a construct that would be grammatically awkward in the original sentence due to repetition of the noun phrase. As we will see later, such examples, while relatively rare, pose a problem for our feature extraction mechanisms which do not directly account for such implicit arguments. In the current system, the argument boundaries are placed around 'priceless' or given as 'not present' at the discretion of the human tagger.

Corpus examples where one argument was syntactically bounded by another were encountered with some frequency, as in the following example: "The United States, lest it be forgotten, is about to begin its ninth consecutive year of limits on Japanese car imports..."

Here the main argument, reconstructed, appears to be *"The United States is about to begin its ninth consecutive year of limits on Japanese car imports..."*. Disregarding the unwritten elaboration, the remaining argument inclusive of the cue is *"lest it be forgotten"* – where *it* corefers with the objectively stated fact in the previous argument. The tagging procedure here, and indeed, the functional methods for extracting meaningful paths and subtrees from the automatically generated trees, provide no direct method for explicitly representing such splits. Consequently, the subsumed argument is tagged as in canonical examples, while the boundaries of the split argument are placed around the full structure.

The complexities of syntax are such that it would be impractical to develop heuristics for addressing every such example. Consequently, we attempt to find simplifications that avoid overtly altering the semantic aspects of the passage we are interested in. However, we will examine one further example in which the tagging heuristics followed here result in a significant semantic alternation to the original meaning. Consider the following sentence, extracted with the cue *in order that*:

> "If we are being brought to our knees, so to speak, by the need for more priests, is it not in order that we may understand with greater humility and love who the Lord of the harvest really is," he asked?

For our purposes, the desired semantic relation is encoded even when the phrasal elements associated with the question and attribution are eliminated from the original sentence: "We are being brought to our knees, so to speak, by the need for more priests in order that we may understand with greater humility and love who the Lord of the harvest really is."

This is, of course, a dramatically different statement; were this part of a full-fledged discourse processor, we would want to avoid situations in which our boundary tags created the suggestion of fact from a rhetorical question, suggestion, allegory, or any other form of unqualified opinion.

3.5.2 Tools

Charniak's Parser is a tool developed at Brown University under the supervision of Dr. Eugene Charniak to automatically annotate sentences with syntactic parse trees. While tools have been developed which out-perform this tool when properly trained and used on sufficiently large corpora, we selected Charniak's parser as it presents a well-documented and highly accurate account of syntactic structure on a wide variety of corpora. A full description of the statistical method implemented in this parser is available in (Charniak 1997). Briefly, for a given sentence the tool returns the parse π for of a sentence s for which the probability that π is represents the correct parse is the highest. Initial training of the system (inference of grammar rules and probabilities) is performed using the Penn Treebank.

SPADE, a discourse segmenting and parsing tool developed at ISI/USC, automatically idenfities a list of 18 discourse relations in English text compressed from a comprehensive list of 110 as described in (Carlson et al. 2003). The relations include ATTRIBUTION, BACKGROUND, CAUSE, COMPARISON, CONDITION, CONTRAST, ELABORATION, ENABLEMENT, EVALUATION, EXPLANATION, JOINT, MANNER-MEANS, TOPIC-COMMENT, SUMMARY, TEMPORAL, AND TOPIC CHANGE. These relations are more coarse-grained than our list of contingency relations. The system claims a performance of 75.5% F-score using perfect syntactic trees and perfect discourse segmentation when the human annotation agreement is 75%.

In its parsing phase, SPADE implements a probabilistic procedure that examines all legal discourse parse trees for a particular sentence via dynamic programming, performing comparisons of legally derived semantic constituents for each text span encountered and eliminating those with lower probabilities.

SPADE outputs tagged text representations of discourse structures in the style of RST, modified to provide for adequate handling of discontinuous arguments (this is described in more detail in a later section).

GRD is a tool developed at the University of Texas at Dallas that automatically labels grammatical roles in unannotated text. These roles are *subject, direct object indirect object, oblique object, copular, complement, and adverbial.* It also provides verb voice information. The performance of GRD is 87% F-score. At the time of writing, this tool is not publicly available.

3.5.3 Features

We identified and experimented with a set of 9 syntactic and semantic features. Their values are determined with the help of several publicly available tools, an in-house grammatical role analyzer, and a series of Perl scripts for text filtering, XML conversion, and XML processing. A detailed description of each is provided in the following section.

To better exemplify each feature, we will consider the following short example: *"Crampton was able to identify the ball, so it was only a one-stroke penalty."* The syntactic tree of this example sentence enriched with grammatical role tags is shown in Figure 3.4. Figure 3.8 shows the corresponding discourse structure as provided by SPADE.

The *Cue-Main Path* feature (Feature 1) describes the path between the head node of the main argument and the head of the cue phrase. In each case, the head node is considered to be the immediate parent in the syntactic parse tree of the text node corresponding to the first word in the argument or cue. We selected this feature because of its importance in semantic role detection (Gildea and Jurafsky 2002). The

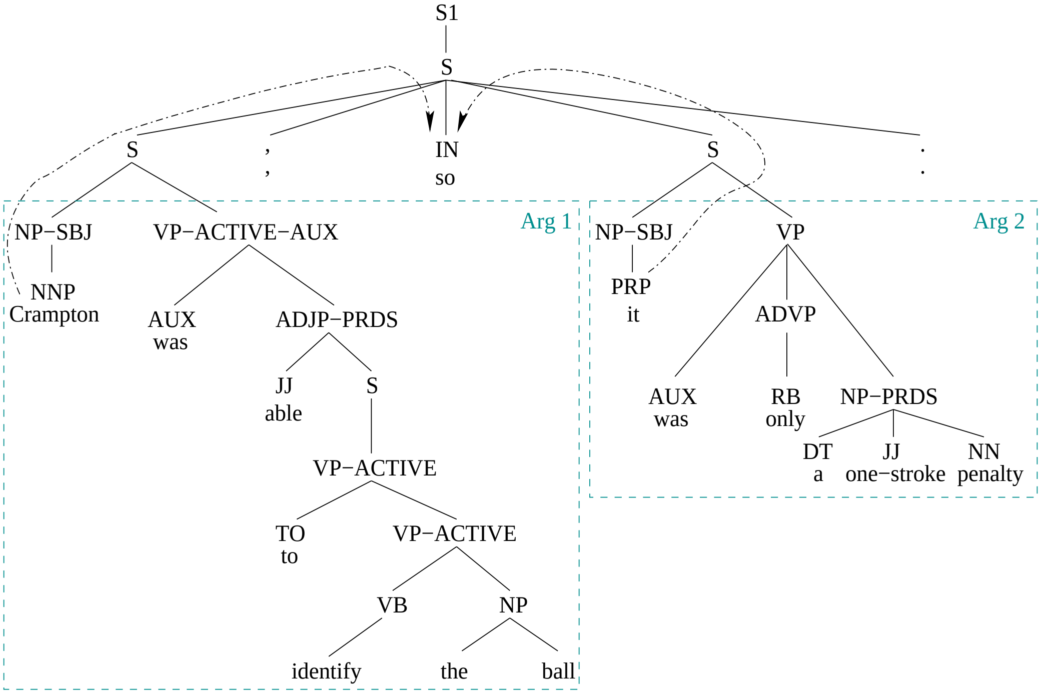


Figure 3.4. Features 1 and 2

feature is primarily designed to encode the syntactic relationship between a specific argument and the cue anchor which is, in this context, the most relevant link to the remainder of the sentence(s). This and all subsequent features are represented in the system as individual strings encoding the appropriate sequence of syntactic, grammatical, and grammatical nodes. Note that the star in the following string indicates the transition from moving “up” the path from the head of the main argument to moving “down” the path following the shared parent node in the tree structure. The encoding for the syntactically and grammatically annotated sentence in Figure 3.4 is *NNP-NP-S*IN*.

The *Cue-Subordinate Path* feature (Feature 2) describes the path between the first node of the subordinate argument and the head of the cue phrase. Functionally, the construction of this feature is performed as in Feature 1. We include this feature to further articulate the link between syntactic realization and the associated shallow semantic parse. The sample encoding is *PRP-NP-S*IN*.

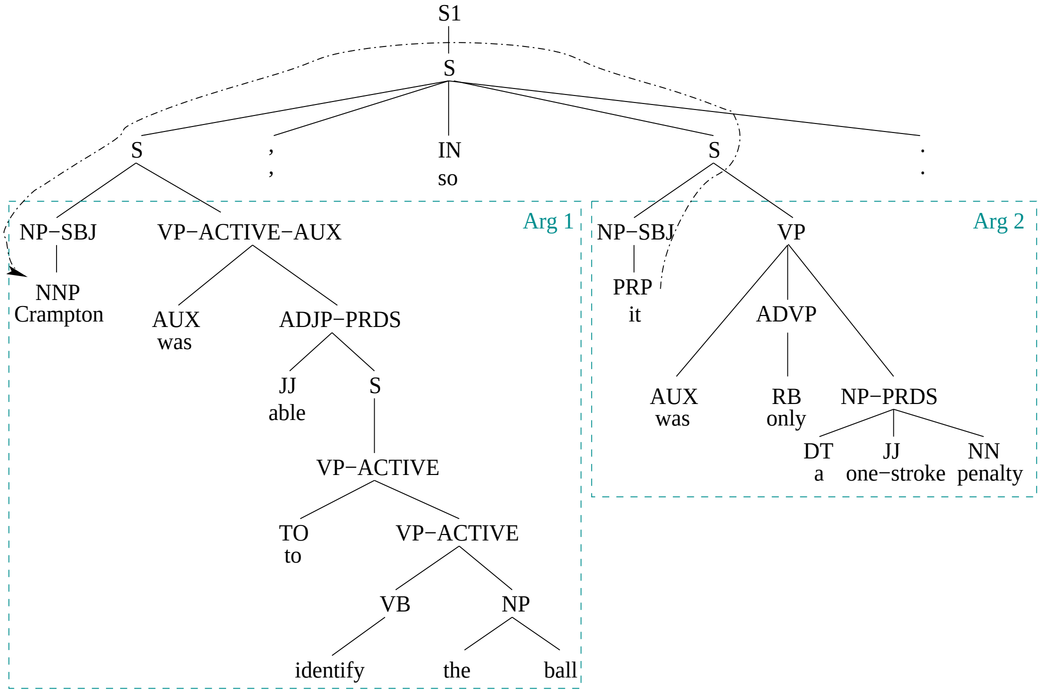


Figure 3.5. Feature 3

The *Main-Subordinate Path* feature (Feature 3) describes the path from the subordinate to the main argument. This feature captures syntactic information about the position of both arguments in relation to each other, a particularly important relationship when examining semantic relations in an inter-sentential context. Note that, in this example, the format of this feature (from left to right) begins at the first node in the main argument, moves up the parse tree path to the minimally shared syntactic node, and then down to the subordinate argument's first node. The sample encoding is *PRP-NP-S*S-NP-NNP*.

The *Subordinate In-order Traversal* feature (Feature 4) describes the in-order traversal of the subordinate clause. This feature is important because it provides a compact representation of the argument structure as it appears in the automatically generated syntactic parse tree. We believe that even at this level of semantic granularity, there may be distinct and consistent differences between argument structures from one relation to another. This feature includes the direct path from the sentence

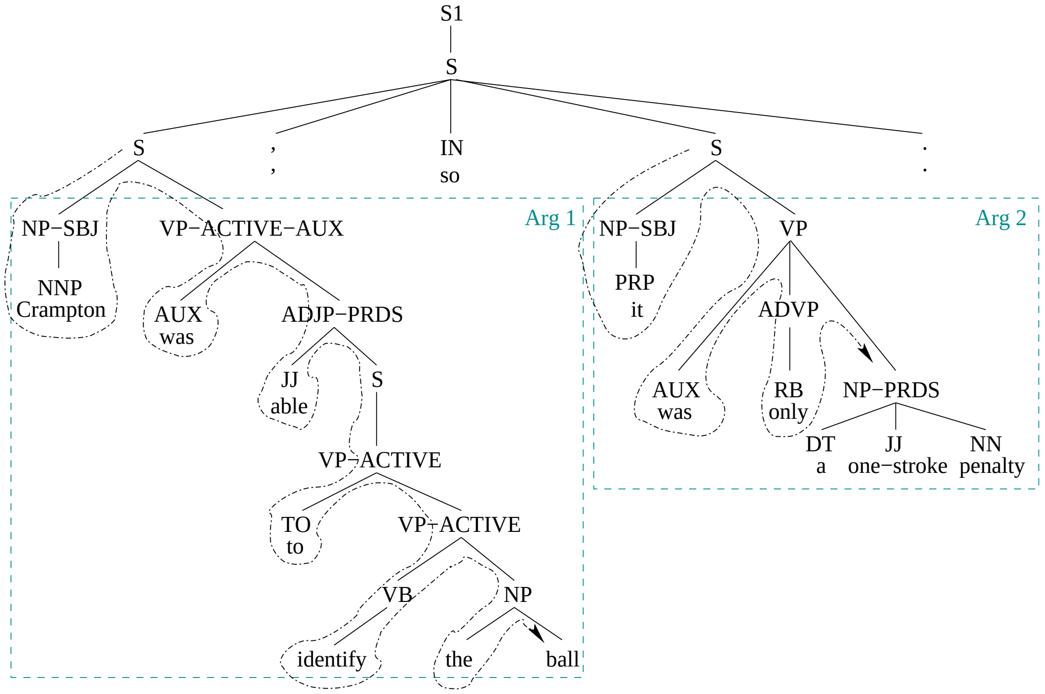


Figure 3.6. Features 4 and 5

root to the argument head in addition to the traversal of the argument itself. The sample encoding is *S-NP-PRP-VP-AUX-ADVP-RB-NP-DT-JJ-NN*.

The *Main In-order Traversal* feature (Feature 5) is the in-order traversal of the main clause syntactic tree. Functionally, this is constructed in the same manner as Feature 4. The sample encoding is *S-NP-NNP-VP-AUX-ADJP-JJ-S-VP-TO-VP-VB-NP-DT-NN*.

The *Flat XML Tree* feature (Feature 6) is a flat representation of the Charniak parse tree constructed via an in-order traversal of the XML-converted representation. The feature provides a complete and compact representation of the full syntactic structure of the sentence(s). The sample encoding is *SFC-S1-S-S-NP-NNP-VP-AUX-ADJP-JJ-S-VP-TO-VP-VB-NP-DT-NN-COMMA-IN-S-NP-PRP-VP-AUX-ADVP-RB-NP-DT-JJ-NN-PERIOD*.

The *Flat GRD Tree* feature (Feature 7) is a flat representation of the syntactic tree augmented with grammatical role labels using the GRD tool. This fea-

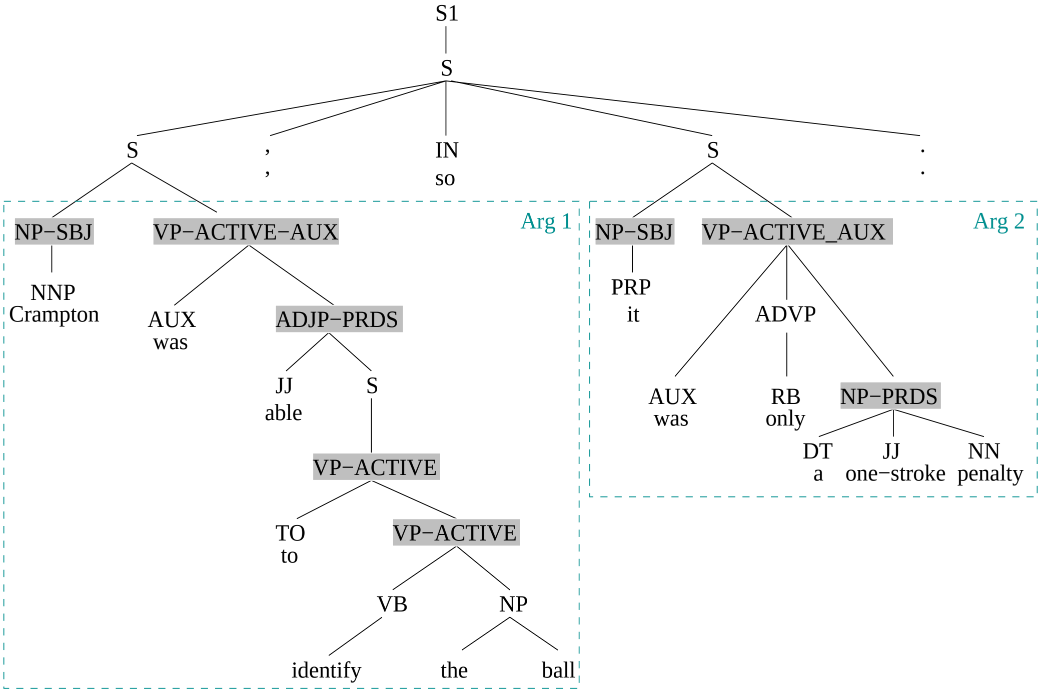
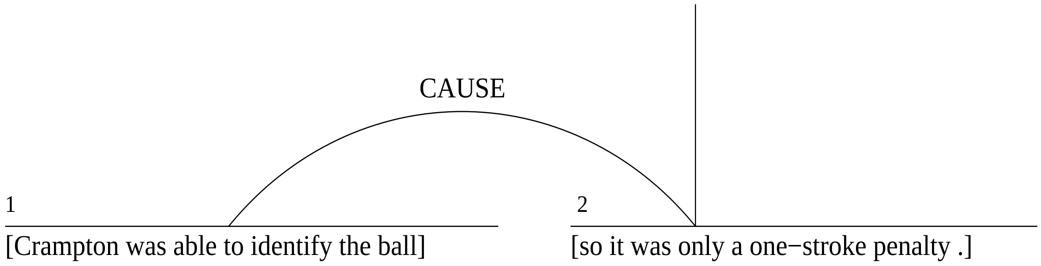


Figure 3.7. Feature 7

ture complements those generated using the Charniak and SPADE parsing tools by providing a complete account of the grammatical structure of the sentence. Additionally, we expect some correlation between subtrees located under particular grammatical roles and those text spans identified by SPADE. The sample encoding is *NP_SBJ VP_ACTIVE_AUX ADJP_PRDS VP_ACTIVE VP_ACTIVE NP_OBJD NP_SBJ VP_ACTIVE_AUX NP_PRDS.*

The *In-order Discourse Traversal* feature (Feature 8) is an ordered list of the values associated with the *rel2par* tag generated by SPADE. The *rel2par* values comprise the rhetorical relations that hold between the discourse units identified automatically by SPADE. The order in which they appear provides a compact account of the semantic organization of the sentence without an explicit description of structural information. The sample encoding is *Cause span.*

The *All In-order Discourse Traversal* feature (Feature 9) is an in-order traversal of *all* SPADE discourse tags with the text spans removed. This feature is essentially


CAUSE
1
2
[Crampton was able to identify the ball]
[so it was only a one−stroke penalty .]

```
(Root (span 1 2)
      (Satellite (leaf 1) (rel2par Cause)
(text _!Crampton was able to identify the ball_!))
      (Nucleus (leaf 2) (rel2par span)
(text _!so it was only a one−stroke penalty._!))
 )
```

Figure 3.8. Features 8-9

the SPADE equivalent of the in-order traversal of the Charniak parse tree. The actual text items have been removed, since given that there are virtually no identical sentences in the corpus, this would make the feature irrelevant. The sample encoding is *SFC Root span 1 2 Satellite leaf 1 rel2par Cause text Nucleus leaf 2 rel2par span text.*

As a final note, the traversals conducted in features 1 through 6 are made possible via consulting the manually tagged argument boundaries. These are kept in files corresponding exactly in form and content to those containing the XML-converted syntactic trees, but tagged in a simple numeric format (Arg1-head Arg1-tail Arg2-head Arg2 tail) corresponding to the beginning and end of each of the arguments. In the following example, each numeric tag represents a linear position in the sentence. The final set of four numbers indicate the head and tail of the main and subordinate arguments as selected by the human annotator according to the guidelines listed in Section 3.5.1.

<0> Crampton <1> was <2> able <3> to <4> identify <5> the <6> ball <7> , <8> so <9> it <10> was <11> only <12> a <13> one-stroke <14> penalty <15>
0 6 9 14

A formal account of argument boundaries generally includes the discourse connective (the associated cue) with the argument before which it appears (Gildea and Jurafsky 2002). As cues do not consistently appear immediately preceding the subordinate argument (for example, sentences in non-canonical form or those in which some movement rule has been invoked) and due to the necessity of keeping the argument boundary representation simple, the cue is not included in this argument boundary representation.

3.5.4 Overview

Figure 3.9 illustrates the implementation steps described in the preceding section. Individual arrows within the *Feature Extraction* module indicate unique parses of the data.

The features described in the preceding section constitute our hypothesis for those syntactic and semantic attributes we expect to be most relevant for learning a "good" classifier, one which will provide for high-accuracy separation of the CONTINGENCY relations. In features 1 and 2, we address structural information related to cue placement relative to the respective arguments, without resorting to using lexical information provided by the cue itself. In feature three, we provide a compact syntactic representation directly related to the syntactic realization of the encoded argument. As it is well established in the literature that the argument informs this realization (albeit not unambiguously), this is an important inclusion. For the same reason, we include representations of the syntactic subtrees corresponding to each argument in features 4 and 5. As these representations are necessarily more complex, future revisions of the feature vector may perform lemmatization of verb forms and compacting of terminal noun phrases to limit the number of unique feature values. Feature 6 provides similar syntactic information (and may be subject to the same future constraints). Feature 7 adds information lacking in the syntactic parse by

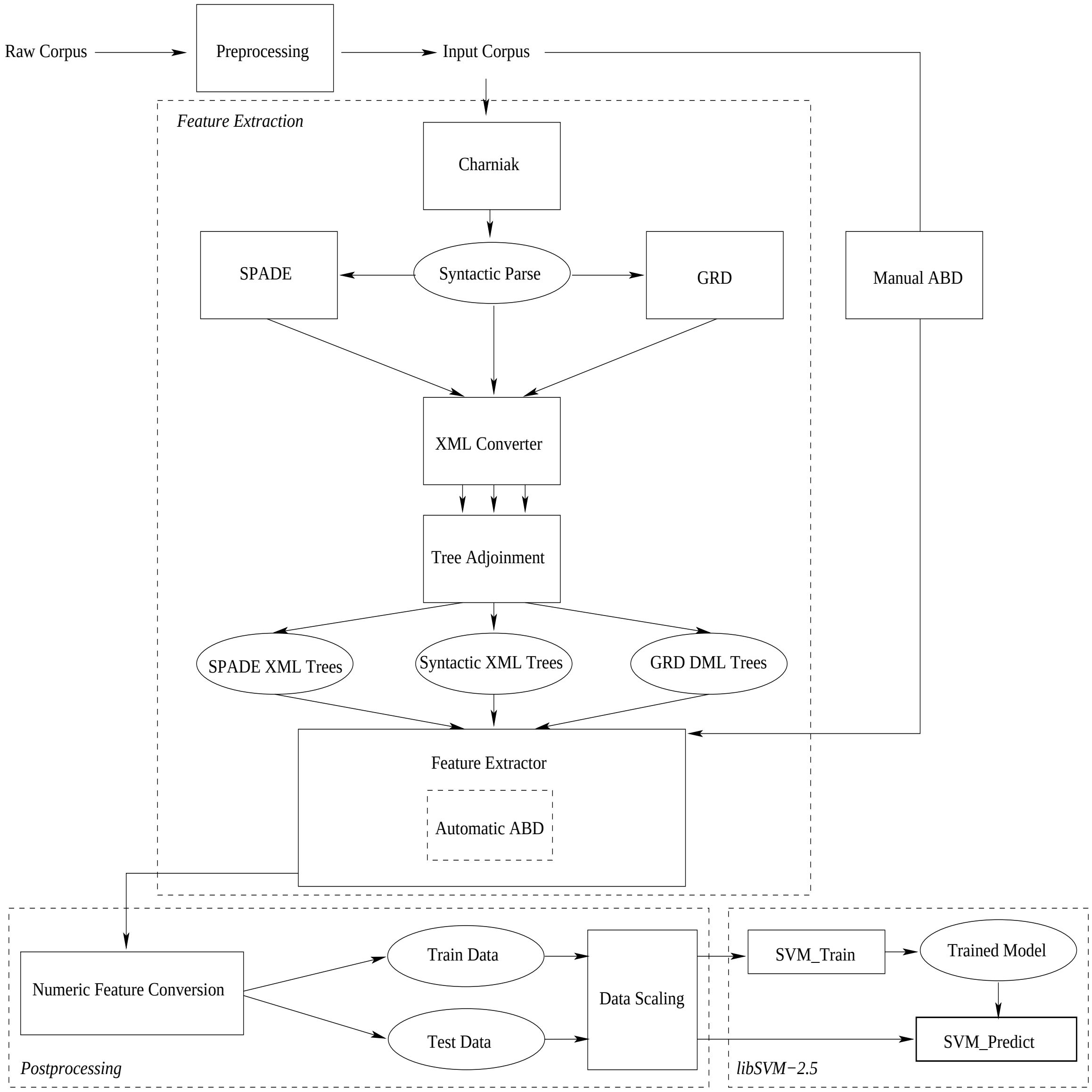


Figure 3.9: System Architecture: Arrows within the dashed-line Feature Extraction box correspond to individual syntactic, semantic, and grammatical parses. Note that Automatic ABD is performed in tandem with the actual feature extraction, while manual argument boundaries are passed in as a separate parse of the data.

modifying the syntactic tree with grammatical roles, providing a more complete description of the syntactic behavior of each argument. Finally, features 8 and 9 provide broad semantic coverage for each data instance.

CHAPTER FOUR

Experimental Results

4.1 Overview

We performed a variety of experiments designed to provide insight into the contribution of both lexico-syntactic features and semantic features to classification accuracy when considering the Contingency relations. Additionally, experiments using both the automatic heuristic for location of argument boundaries and the manually tagged argument boundaries were performed on identical data sets.

For all experiments, the corpus described in Chapter 3 was split into training and testing using a 70/30 ratio. On this data, we performed a series of one-vs-one and one-vs-all classification experiments. Vectors in the experiment consisted of nine numeric values in a regular (non-sparse) format. Unique feature values were recorded as follows, listed here in the order of the features previously described: 519, 1220, 1848, 2357, 2394, 2418, 2320, 1387, and 1504. The total number of unique training and test examples (sentences or attached sentence pairs) was 2433.

Baselines for all classification results are centered around 50% (we provide different baseline values as the magnitude of examples available for each relation instance varies). The baselines are computed from the perspective of the class listed in the left-hand column in each trial. For example, in "Cause vs. Concession", where there are 147 examples of Cause and 150 examples of Concession in the test set, were we to randomly select a class for an unseen test item we would expect to have that selection result in a "true positive" 49.4% of the time. These baselines are provided in Table 4.1.

Results in each case (except for those indicating feature contribution) are given both for the SVM operating with default values for C and γ, and for approximated

Table 4.1. One-vs-One Classification Baselines (Accuracy)

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	49.49	56.98	52.50	55.06	71.01
Concession		-	57.47	53.00	55.56	71.43
Condition			-	45.49	48.05	64.91
Purpose				-	52.57	68.91
Reason					-	66.67
Result						-

optimal values derived using the grid-search procedure. As we will see, the grid-search procedure, even when run at a fine granularity, does not always yield parameter values that result in a performance increase.

4.2 One-Vs-One (7 Features, Automatic Argument Boundaries)

The results presented in Tables 4.2 through 4.5 were obtained using an early version of the feature extraction system incorporating the previously described automatic argument boundary detection heuristic and lexico-syntactic features only (that is, no SPADE data). Both results from the SVM learning procedure with an RBF kernel using default parameters for C and γ and results using kernel parameters trained using the aforementioned grid-search procedure are included to demonstrate the benefits of tuning the system.

4.2.1 Default Parameters

Table 4.2. One-vs-One, 7 Features, Automatic ABD (Accuracy)

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	58.61	78.49	60.98	54.58	70.62
Concession		-	83.27	66.32	59.57	71.63
Condition			-	83.07	74.17	70.79
Purpose				-	60.31	69.00
Reason					-	66.67
Result						-

Table 4.3. One-vs-One, 7 Features, Automatic ABD (Precision/Recall (f-measure))

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	58.45/55.70 (57.04)	83.82/76.51 (79.98)	58.94/81.88 (68.54)	54.75/96.64 (69.90)	70.62/100.00 (82.78)
Concession		-	88.03/81.69 (84.74)	63.03/86.93 (73.08)	57.85/98.69 (72.94)	71.69/99.35 (83.22)
Condition			-	82.30/80.17 (81.22)	68.75/85.34 (76.16)	69.51/98.28 (81.42)
Purpose				-	61.33/66.67 (63.88)	69.00/100.00 (81.66)
Reason					-	66.67/100.00 (80.00)
Result						-

4.2.2 *Grid-Search Parameters*

Consistent accuracy gains are achieved through parameter training. Note in particular the significant improvement (14%) in the pairwise comparison of Condition vs. Reason. Even when considering highly ambiguous pairings such as Purpose vs. Reason, consistent accuracy improvements (due primarily to a classification boundary that extends recall performance) are apparent.

Table 4.4: One-vs-One, 7 Features, Automatic ABD, Trained Parameters (Accuracy)

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	59.60	84.15	70.38	58.60	73.46
Concession		-	88.48	68.38	60.29	76.74
Condition			-	84.65	88.75	79.78
Purpose				-	64.12	70.50
Reason					-	62.90
Result						-

4.3 *One-Vs-One (9 Features, Automatic Argument Boundaries)*

Features 8 and 9 (extracted from the SPADE discourse structures) were included in a system revision to determine the contribution of RST-style discourse semantic tags towards system performance. The addition of these features resulted in small but consistent decrease in system performance for almost all classification pairs. However, this does not serve the conclusion that the semantic features introduced in the SPADE data have a negative contribution. In a later section we will see that, at least when

Table 4.5: One-vs-One, 7 Features, Auto. ABD, Trained Params (Precision/Recall (f-measure))

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	59.71/55.70	85.91/85.91	70.78/73.15	59.68/74.49	75.40/92.62
	-	(57.64)	(85.92)	(71.94)	(66.26)	(83.12)
Concession		-	90.13/89.54	70.47/68.63	58.85/93.46	76.68/96.73
			(89.84)	(69.54)	(72.22)	(85.54)
Condition			-	82.91/83.62	86.78/90.52	79.85/92.24
				(83.26)	(88.62)	(85.58)
Purpose				-	63.25/76.09	72.57/92.03
					(69.08)	(81.14)
Reason					-	72.00/72.58
						(72.28)
Result						-

using the "gold standard" manually tagged argument boundaries, this is not the case.

A more consistent explanation may be that the lexico-syntactic and semantic feature values are interfering with each other during the search for the optimal separating hyperplane performed by the SVM. This is most likely due to limitations of the argument boundary detection heuristic. For example, non-canonical form sentences frequently defeat the heuristic, which performs a relatively naive search for maximally bounding verb phrases on either side of the cue phrase. If such a phrase is not found for one argument or the other, a default tag is introduced. Distributions of default tags in the resulting data can create an undesired bias. In addition, some semantic arguments have consistent syntactic structures that do not support the assumption of simple VPs that describe the main body of the argument. In such cases, the heuristic may find VPs that describe some syntactic pattern unrelated to the semantic argument we are interested in.

Results from the experimental trials with 9 features and automatically detected argument boundaries are given in Tables 4.6 through 4.9.

4.3.1 Default Parameters

Table 4.6. One-vs-One, 9 Features, Automatic ABD (Accuracy)

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	57.28	69.81	63.41	59.71	70.62
Concession		-	73.98	64.26	63.18	71.64
Condition			-	62.59	70.42	66.29
Purpose				-	63.74	69.00
Reason					-	65.59
Result						-

Table 4.7. One-vs-One, 9 Features, Automatic ABD (Precision/Recall (f-measure))

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	61.90/34.89 (44.62)	72.84/73.83 (73.34)	64.67/65.10 (64.88)	58.52/89.93 (70.90)	70.62/100.00 (82.78)
Concession		-	74.56/82.35 (78.26)	62.69/79.08 (69.94)	62.69/82.35 (71.18)	71.49/100.00 (83.38)
Condition			-	60.61/51.72 (55.81)	66.67/77.59 (71.72)	67.28/93.97 (78.42)
Purpose				-	63.19/74.64 (68.44)	69.00/100.00 (81.66)
Reason					-	66.85/99.19 (79.88)
Result						-

4.3.2 Grid-Search Parameters

While classification accuracy drops when using trained parameters (Table 4.8, this does not necessarily indicate that the learned model is “worse”; frequently, the new split of the text data results in more evenly balanced classification performance across both classes. For example, while we may have fewer “true positives” for the class listed in the left-hand column, we may see a parallel increase in “true negatives” or examples of the other class involved in the trial that were classified correctly.

Table 4.8: One-vs-One, 9 Features, Automatic ABD, Trained Parameters (Accuracy)

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	56.95	71.30	70.73	61.54	68.72
Concession		-	76.95	67.69	63.18	71.63
Condition			-	76.38	78.33	70.79
Purpose				-	67.18	70.50
Reason					-	65.59
Result						-

Table 4.9: One-vs-One, 9 Features, Auto. ABD, Trained Parameters (Precision/Recall (f-measure))

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	62.03/32.89 (42.98)	71.86/80.54 (75.96)	74.44/66.44 (70.22)	61.96/76.51 (68.48)	70.44/95.97 (81.24)
Concession		-	84.21/73.20 (78.32)	69.79/67.97 (68.86)	63.78/77.12 (69.82)	75.84/88.24 (81.58)
Condition			-	75.00/72.41 (73.68)	77.12/78.45 (77.78)	75.81/81.03 (78.34)
Purpose				-	68.31/70.29 (69.28)	73.09/90.58 (80.90)
Reason					-	67.24/94.35 (78.52)
Result						-

4.4 One-Vs-One (7 Features, Manual Argument Boundaries)

For the following trials, the feature extraction system based on automatic argument boundary detection heuristic was modified to use the manually annotated boundaries described in Chapter 3.

In almost every trial listed in Table 4.10, the classification accuracies here are lower than those recorded for the trials using the automated argument boundary detection heuristic (although still above the associated baseline). Potential sources of this discrepancy may include the previously described possibility of a bias introduced by the heuristic. A direct way to address this hypothesis would include application of the system to a massive corpus providing a wider selection and distribution of syntactic forms and examine the resulting classification accuracies to determine whether such a bias existed. Errors or unidentified biases in the manual tagging may also account for the decrease in accuracy. Manual annotations are typically performed by a group of trained individuals, with resulting corpora adjusted for inter-annotator agreement and vetted for tagging errors. This was not possible here due to time constraints and the fact that external annotators were not available.

4.4.1 Default Parameters

Table 4.10. One-vs-One, 7 Features, Manual ABD (Accuracy)

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	55.29	65.66	58.89	54.95	70.62
Concession		-	69.89	56.01	55.23	71.16
Condition			-	60.63	64.17	72.47
Purpose				-	59.54	69.00
Reason					-	66.13
Result						-

Table 4.11. One-vs-One, 7 Features, Manual ABD (Precision/Recall (f-measure))

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	54.86/53.02 (53.84)	84.25/47.65 (60.88)	56.59/83.26 (67.38)	54.89/97.97 (70.36)	70.62/100.00 (82.78)
Concession		-	87.50/54.90 (67.46)	55.02/89.54 (68.16)	58.10/67.97 (62.64)	71.16/100.00 (83.16)
Condition			-	54.08/91.38 (67.94)	58.52/88.79 (70.54)	70.30/100.00 (82.56)
Purpose				-	61.26/63.04 (62.14)	69.00/100.00 (81.66)
Reason					-	66.85/97.58 (79.34)
Result						-

4.4.2 Grid-Search Parameters

When classification trials are performed using trained parameters (Tables 4.12 and 4.13), moderate improvements in accuracy are observed for a number of class pairs, including Concession vs. Purpose and Purpose vs. Reason. As in previous trials, note that all trials involving Result perform at or near the baseline (final column in Table 4.12), due to the lack of sufficient training instances. This will be addressed in future revisions of the system through the inclusion of additional discourse cues.

Table 4.12. One-vs-One, 7 Features, Manual ABD, Trained Parameters (Accuracy)

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	53.64	66.42	58.19	54.21	69.19
Concession		-	69.52	61.17	56.32	71.16
Condition			-	60.24	67.50	72.47
Purpose				-	64.50	68.00
Reason					-	67.74
Result						-

Table 4.13: One-vs-One, 7 Features, Manual ABD, Trained Parameters (Precision/Recall (f-measure))

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	52.87/55.70 (54.24)	82.61/51.00 (63.06)	58.38/67.79 (62.74)	55.94/75.84 (64.38)	70.19/97.99 (81.78)
Concession		-	82.57/58.82 (68.70)	60.64/74.51 (66.86)	58.42/72.55 (64.72)	71.98/97.39 (82.78)
Condition			-	53.85/90.52 (67.52)	61.18/89.66 (72.74)	70.30/100.00 (82.56)
Purpose				-	64.90/71.01 (67.82)	69.07/97.10 (80.72)
Reason					-	68.18/96.77 (79.98)
Result						-

In most trials, the accuracy gain attributed to features 8 and 9 is moderate, and in only two cases is the contribution negative. In considering these results, it is important to remember that there is not a 1-to-1 relationship between the semantic roles defined in RST (and by association, SPADE) and the contingency relations considered in these experiments. It is therefore difficult to make qualitative statements regarding the contribution of these features. Additionally, many of the cues used in these experiments are not recognized by SPADE as potential candidates for the encoding of a particular relation, and thus will never appear in a text span produced by SPADE following a desired semantic relation.

4.5 One-Vs-One (9 Features, Manual Argument Boundaries)

Table 4.14 shows the performance results computed for each pair of discourse relations when using both SPADE features and manually detected argument boundaries. These results correspond to the latest revision of the system for one-vs-one trials.

4.5.1 Default Parameters

As discussed previously, we observe improvement in classification results due to the addition of features extracted from SPADE data, even when the SVM procedure is run using default parameters. Note, in addition, the consistently lower classification accuracies (when compared to other trials in the same block of experiments) on semantic argument pairs where shared ambiguous cues are present, such as Cause vs. Reason.

Table 4.14. One-vs-One, 9 Features, Manual ABD (Accuracy)

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	55.96	66.04	60.98	59.34	70.62
Concession		-	68.03	63.23	64.26	71.16
Condition			-	60.63	63.33	72.47
Purpose				-	61.45	69.00
Reason					-	67.20
Result						-

Table 4.15. One-vs-One, 9 Features, Manual ABD (Precision/Recall (f-measure))

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	55.00/59.06 (56.96)	86.42/46.98 (60.86)	58.15/88.59 (70.22)	58.41/88.59 (70.40)	70.62/100.00 (81.78)
Concession		-	87.64/50.98 (64.46)	59.91/90.85 (72.20)	67.31/68.63 (67.96)	71.16/100.00 (83.16)
Condition			-	54.08/91.38 (67.94)	57.78/89.66 (70.28)	70.30/100.00 (82.56)
Purpose				-	63.12/64.49 (63.78)	69.00/100.00 (81.66)
Reason					-	67.21/99.19 (80.12)
Result						-

4.5.2 Grid-Search Parameters

When trained parameters are used, we see one significant drop in classification accuracy (Cause vs. Concession, which still remains above the baseline), but both a general trend towards both better classification accuracy and improved consistency in the balance between precision and recall.

Table 4.16. One-vs-One, 9 Features, Manual ABD, Trained Parameters (Accuracy)

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	48.68	65.66	64.11	60.44	70.14
Concession		-	70.63	64.95	64.26	71.16
Condition			-	61.42	66.67	73.03
Purpose				-	60.31	67.00
Reason					-	66.67
Result						-

Table 4.17: One-vs-One, 9 Features, Manual ABD, Trained Params (Precision/Recall (f-measure))

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	50.27/61.07 (55.14)	81.52/50.34 (62.24)	64.94/67.11 (66.00)	59.28/87.92 (70.82)	70.48/99.34 (81.46)
Concession		-	88.54/55.56 (68.28)	65.27/71.24 (68.12)	66.88/69.93 (68.38)	71.16/100.00 (83.16)
Condition			-	54.69/90.52 (68.18)	60.59/88.79 (72.02)	70.73/100.00 (82.86)
Purpose				-	61.18/67.39 (64.14)	68.94/94.93 (79.88)
Reason					-	68.24/93.55 (78.92)
Result						-

4.6 Evaluation of Classification Results

In this section we provide a more detailed analysis of the classification results obtained from the one-vs-one trials using 9 features, manually tagged boundaries, and grid-search trained parameters C and γ. This set was selected for examination as it corresponds to the latest revision of the system.

4.6.1 Cause vs. Concession

Cause and Concession are semantically distinct, sharing no cues in the extractions used here. This holds both in examining these relations from the perspective of contingency (fine granularity) and from the perspective of RST, where cause forms a super ordinating class including result, and concession is grouped with a number of other relations as Contrast.

It appears that the comparatively low performance of the system on this comparison may be due to a combination of insufficiently precise syntactic features, and, more importantly, the fact that the two features derived from the automatic SPADE tagging may frequently encode information about a different relation that holds between the text segments in the sentences - a relation that we do not consider in our analysis. For example, the SPADE features may correctly identify the sentences as encoding (respectively) the relations Cause and Contrast, but may also indicate that both sentences also encode some *other* relation, such as the Temporal relation, simultaneously. This is a key point, as it provides some insight into why the system on occasion underperforms on comparisons of relations that do not appear semantically close. The learned model may in fact be selecting for some *third* relation indicated by the feature data.

4.6.2 Cause vs. Condition

Cause and Condition share no ambiguous cues in our schema. It should be noted that in RST, Contingency is considered to be a semantic subclass of Condition, but that Contingency in this sense does not refer to the set of semantic relations that we consider to be unique classes in this set of experiments. Additionally, the high performance of this comparison may be due to the high frequency of non-canonical form sentences in the set of extracted Condition sentences. For example, when considering the cues "provided that" and "so long as", we see a large number of examples where the form is "*So long as* Arg1, Arg2" as opposed to the canonical "Arg2 *so long as* Arg1".

4.6.3 Cause vs. Purpose

The same reasoning as in Cause vs. Condition applies here. The distribution (and order) of the semantic labels produced by SPADE (in addition to the other syntactic features) is frequently inverted (due to frequent non-canonical form sentences in Purpose). The relation we call Purpose is tagged as Enablement by SPADE. These are almost certainly the primary factors contributing to the high accuracy here.

4.6.4 Cause vs. Reason

Here we see the second lowest accuracy in this row in the accuracy table. Somewhat lower accuracy is expected here due to the shared ambiguous cue 'because', which can also encode relations not considered in this set. This result is somewhat better than we might expect, given the semantic similarity between these two relations. (Quirk 1985) note that for sentences tagged by 'because' in particular, there is a subtlety of distinction between cause and reason that can depend on such factors as the relative authority of the source of the information.

We should therefore look elsewhere to explain this high degree of accuracy. One probable explanation has to do with the extraction of sentences with the 'as' and 'for' cues for the relation Reason. These cues are interesting because they can encode this relation, encode other relations outside of the set considered, or be essentially semantically empty depending on their use. When we were preparing this corpus, we noted that in the majority of cases where 'as' and 'for' encoded reason, they were preceded by a comma - a mark which in most cases served to emphasize the use of that cue as a semantic marker. Therefore, a large number of the training and test examples for the relation Reason have this syntactic form, which may serve to differentiate the class as a whole given the proportion of syntactic features in our feature vector.

4.6.5 Cause vs. Result

These relations do not share any ambiguous cues. The accuracy observed here would probably be greater if we included more cues for the relation RESULT (something of an anomaly in our data, as it is cued by only two phrases, 'so' and 'so that'). Note that in RST, CAUSE and RESULT are both semantic subclasses of the superordinating relation CAUSE.

We will observe this in all experimental comparisons involving RESULT. Accuracies which appear initially to be significantly higher than those seen in other class comparisons can be directly attributed to the unbalanced data; as a result, those experiments in which RESULT is involved frequently fail to out-perform the baseline.

4.6.6 Concession vs. Condition

Accuracy (and precision and recall) are all unusually high here. While these relations are semantically distinct both in our contingency scheme and in RST, this may be due to some peculiarity of syntactic form in the sentences extracted for the corpus used here. However, a preliminary examination of the SPADE features extracted for CONCESSION indicates that SPADE consistently tags those relations we consider to be CONCESSION as CONTRAST (the superordinating term in RST), so this may be an example of a pair of relations that exhibits sufficient semantic distance to make the classification task "easy".

4.6.7 Concession vs. Purpose

Here we observe relatively high (71.24%) recall of the class CONCESSION in the classification task with PURPOSE, but somewhat lower precision. This overclassification of examples of PURPOSE as CONCESSION is difficult to explain. Even in RST, these relations occupy semantically remote classes (CONTRAST and ENABLEMENT, respectively) and share no ambiguous cues. One potential explanation is that in the compact corpus used here, the syntactic complexity of the sentences extracted leads to the simultaneous learning (as described before) of some third, common relation not considered as part of this set.

4.6.8 Concession vs. Reason

As in the comparison of Concession to Purpose, we see adequate recall paired with somewhat lower precision. Since these classes, again, exhibit no semantic overlap (when encoded by the cues used here), this speaks to a tendency of the learning model to classify based on syntactic similarities; that is, the SPADE features do not adequately address the difference between these two relations, even though they are separately classed in RST (respectively under Contrast and Elaboration).

4.6.9 Concession vs. Result

As described previously, the lower performance of the system in this case is best explained by the relatively small number of training and test examples available for Result. This will be addressed in a future version of the system.

4.6.10 Condition vs. Purpose, Condition vs. Reason, and Condition vs. Result

The system exhibits remarkable accuracy in each of these comparisons. Even in the comparison to Result, the lack of an equivalently sized set of data for the result cues does not seem to result in misclassifications. It appears that both at the level of contingency relations, and in RST, where Purpose, Reason, and Result are respectively classified under Enablement, Explanation, and Cause, there is sufficient semantic distance between these classes to make classification relatively simple. It is not immediately clear why this should be the case. One explanation may be that the syntactic features extracted for this relation differ consistently from those in the other classes considered due to the relatively large size of the cue phrases selected for Condition (such as 'as long as', 'in the event that', and 'so long as'). In turn, this could result in consistent differences in the semantic trees automatically derived by SPADE.

4.6.11 Purpose vs. Reason

Purpose and Reason (respectively superordinated by Enablement and Explanation in RST) share no ambiguous cues. However, a manual examination of the extracted SPADE feature indicates that the Enablement relation frequently appears in the SPADE features extracted for sentences appearing in the Reason section of the corpus. This is, as noted previously, due to the fact that in the majority of sentences encountered in the corpus more than one relation is encoded (due to the length and complexity of the points involved). Semantic separation of these relations in the classification task will certainly improve as more refined methods for interpreting the SPADE data are developed.

4.6.12 Purpose vs. Result

Here again we see the "high recall, relatively low precision" phenomenon. As established previously, this has less to do with the semantic distance between the relations than with the inadequate size of the Result section of the corpus.

4.6.13 Reason vs. Result

Low performance here is most readily explained, again, by the lack of sufficient test and training data (generated from only 4 and 2 cues for each relation, respectively). In addition, a preliminary examination of the SPADE features in the Result feature files seems to indicate that the cues used in this experiment, 'so' and 'so that', rarely if ever trigger the correct superordinating RST semantic tag (Cause) in SPADE. Addressing this will require further work.

4.6.14 Additional Comments

Providing an adequate account of the semantic structure and distribution of the set of contingency relations is a complex task. Initially, it was not obvious how to process the RST data from the trees generated by SPADE to aid in the classification task. However, we have made some general observations on the process.

First, there is not a direct mapping from the contingency relations as defined in (Quirk 1985) to those sets of semantic relations defined in RST. Nevertheless, the 16-class partitioning noted by Marcu in the RST tagging manual provides a good account of the cases in which these relations can be semantically separated even which shifted into a classification scheme of coarser granularity. Due to this, it might be useful to examine *only* those rel2par values that appear immediately prior to the cue for the relation we are looking for, either for inclusion as a feature in the system or simply to provide an account of the consistency of the mapping between our relations and the superordinating relations in RST.

Second, certain of the cue phrases we utilize will never trigger the appropriate RST semantic class into which the particular contingency relation we are examining should fall. This is a major issue, as it means that, effectively, the SPADE features cannot provide us with a reasonable account of the semantic structures that we are interested in *when using those cues.* Future work will include a more detailed analysis of corpus cue distribution for the *contingency relations*, in addition to the selection of multiple cue sets for each relation. Various methodologies have been developed for the study of discourse cue usage (Moser and Moore 1995), with applications towards both generative and interpretive systems.

4.7 One-Vs-One Feature Contribution (9 Features, Manual ABD)

In experimental machine learning procedures, it is desirable to extract feature contribution values in order to determine which features are most salient to the task at hand. A formal account of feature contribution would necessitate a prohibitive amount of experimental trials (that is, a trial for each set in the power set of features present), as pairs or groups of features may yield patterns of interference. However, we can still extract useful information about feature contribution by running each trial from the one-vs-one experiments with each feature removed individually.

Intuitively, we expect that features with unique feature value magnitudes approaching the size of our full corpus will have a negligible contribution to the accuracy of the trials, as such features will not aid in the construction of classification rules in the learned model. The converse, however, does not necessarily hold. That is, features with relatively few unique feature values in the corpus used here will not necessarily contribute to higher classification accuracies. The distribution of such feature values may in fact result from a pattern or series of patterns in the data that are not being considered here, and consequently may result in a reduction in classification accuracy.

In Table 4.18, the classification accuracies are listed in order of features removed. For each row element, the topmost value indicates the classification accuracy with features 2-9 present, the next with features 1 and 3-9 present, and so on. Differentials between these and the original classification accuracies are given in parentheses.

Table 4.18: One-vs-One, 9 Features, Manual ABD (Feature Contribution): Difference measures are given in parentheses.

#	Cause	Concession	Condition	Purpose	Reason	Result
Cause	-	53.97 (-1.99)	66.04 (0.00)	60.28 (-0.70)	60.07 (+0.73)	70.62 (0.00)
		55.96 (0.00)	58.49 (-7.55)	56.79 (-4.19)	59.34 (0.00)	70.62 (0.00)
		56.62 (+0.71)	66.04 (0.00)	59.93 (-1.05)	57.51 (-1.83)	70.62 (0.00)
		55.96 (0.00)	66.04 (0.00)	60.98 (0.00)	58.61 (-0.73)	70.62 (0.00)
		55.62 (-0.34)	66.04 (0.00)	63.07 (+2.09)	58.61 (-0.73)	70.62 (0.00)
		55.96 (0.00)	66.04 (0.00)	63.07 (+2.09)	59.34 (0.00)	70.62 (0.00)
		55.96 (0.00)	66.42 (+0.40)	62.72 (+1.74)	57.88 (-1.46)	70.62 (0.00)
		54.97 (-0.99)	66.42 (+0.40)	62.02 (+1.04)	58.24 (-1.10)	70.62 (0.00)
		54.30 (-1.66)	66.42 (+0.40)	61.34 (+0.36)	53.85 (-5.49)	70.62 (0.00)
Concession		-	67.66 (-0.37)	63.91 (+0.68)	64.62 (+0.36)	71.16 (0.00)
			64.31 (-3.72)	61.16 (-2.07)	63.54 (-0.72)	71.16 (0.00)
			69.89 (+1.86)	62.88 (-0.35)	63.18 (-1.08)	71.16 (0.00)
			68.77 (+0.74)	62.54 (-0.69)	64.26 (0.00)	71.16 (0.00)
			69.15 (+0.26)	62.89 (-0.34)	64.62 (+0.36)	71.16 (0.00)
			68.77 (+0.74)	63.23 (0.00)	63.89 (-0.34)	71.16 (0.00)
			68.77 (+0.74)	63.92 (+0.69)	64.62 (+0.36)	71.16 (0.00)
			68.77 (+0.74)	62.19 (-1.04)	64.26 (0.00)	71.16 (0.00)
			69.52 (+0.63)	58.08 (-5.15)	55.59 (-8.67)	71.16 (0.00)
Condition			-	60.23 (-0.40)	64.17 (+0.84)	72.47 (0.00)
				60.63 (0.00)	60.83 (-2.50)	72.47 (0.00)
				61.81 (+1.18)	64.58 (+1.25)	70.78 (-1.69)
				61.47 (+0.84)	63.33 (0.00)	71.35 (-1.12)
				61.81 (+1.18)	63.75 (+0.42)	71.91 (-0.56)
				61.42 (+0.79)	63.75 (+0.42)	71.91 (-0.56)
				62.20 (+1.57)	64.58 (+1.25)	73.59 (+1.12)
				60.63 (0.00)	63.33 (0.00)	72.47 (0.00)
				60.63 (0.00)	63.75 (+0.42)	72.47 (0.00)
Purpose				-	61.07 (-0.38)	69.00 (0.00)
					56.87 (-4.58)	69.00 (0.00)
					60.69 (-0.76)	69.00 (0.00)
					60.31 (-1.14)	69.00 (0.00)
					60.69 (-0.76)	69.00 (0.00)
					61.45 (0.00)	69.00 (0.00)
					61.07 (-0.38)	69.00 (0.00)
					61.07 (-0.38)	69.00 (0.00)
					61.07 (-0.38)	69.00 (0.00)
Reason					-	67.20 (0.00)
						66.67 (-0.53)
						67.20 (0.00)
						67.20 (0.00)
						67.20 (0.00)
						67.20 (0.00)
						67.20 (0.00)
						67.20 (0.00)
						67.20 (0.00)
Result						-

4.8 One-Vs-All (7 Features, Automatic Argument Boundaries)

In addition to the one-vs-one experimental series discussed in the previous section, the system was modified to produce test and training data for classification trials involving selection of a single semantic relation within non-homogeneous set comprising all remaining relations.

The majority of SVM implementations currently available treat multi-class problems as a series of pairwise comparisons. For the one-vs-all series of experiments presented here, the data was reformulated so that examples from the desired class appeared with the original label and all other class examples were grouped together under a single label.

Additionally, early trials confirmed that while the SVM procedure is relatively immune to discrepancies in the magnitudes of the data available for different classes, the system would become overtrained on the non-class data (comprised of all remaining classes), resulting in almost no "true positives" in the trial results, if all examples of the test and training data were used for those classes. Consequently, the size of this data set was attenuated to match that of the desired class, resulting in more uniform results. This does not conform to a distribution we might expect to see in open text, but represents an acceptable compromise while we seek to improve the system. As in previous trials, all baselines for the following results are centered around 50%.

4.8.1 Default Parameters

Using default parameters, classification accuracy is highest relative to the baseline for CONCESSION (remembering that the relatively small number of training and test examples available for RESULT leads to high accuracies that are discounted when we observe that recall is almost nonexistent). Lower classification accuracies for CAUSE and PURPOSE reflect the presence of overtly ambiguous cues, or cues sharing syntactic similarity with those from other classes.

Table 4.19. One-vs-All, 7 Features, Automatic ABD (Accuracy)

Cause	Concession	Condition	Purpose	Reason	Result
57.51	65.34	63.87	57.09	58.97	69.42

Table 4.20. One-vs-All, 7 Features, Automatic ABD (Precision/Recall (f-measure))

Cause	Concession	Condition	Purpose	Reason	Result
59.54/69.13 (63.98)	65.08/80.39 (71.92)	62.09/66.38 (64.16)	62.75/46.38 (53.34)	65.91/46.77 (54.72)	33.33/1.61 (3.08)

Recall for PURPOSE and REASON appears significantly lower than for the preceding three classes. Both of these classes share cues (in the former, *so that* with RESULT and in the latter, *because* with CAUSE) with other relations, so in addition to narrower semantic distance between these and other relations in the CONTINGENCY set, we can expect similarity of syntactic patterns across these classes to be a factor here as well.

4.8.2 Grid-Search Parameters

As in the one-vs-one trials with automatic boundary detection, we see a significant improvement in classification accuracy when using trained parameters C and γ. A significant departure is the accuracy for the class REASON, which decreases slightly. One difficulty we have in evaluating these trends is in determining what is and what is not statistically significant, due to the complexity of a feature extraction process which uses a variety of tools that may not perform at published error rates on a corpus of extremely limited size. We must therefore exercise care in separating anomalous cases from semantically interesting ones.

Table 4.21: One-vs-All, 7 Features, Automatic ABD, Trained Parameters (Accuracy)

Cause	Concession	Condition	Purpose	Reason	Result
66.30	67.15	74.79	60.92	57.69	69.42

Table 4.22: One-vs-All, 7 Features, Auto. ABD, Trained Params (Precision/Recall (f-measure))

Cause	Concession	Condition	Purpose	Reason	Result
68.87/69.79	68.02/76.47	72.58/77.59	65.52/55.07	59.84/61.29	33.33/30.65
(69.32)	(71.98)	(75.00)	(59.84)	(60.56)	(31.94)

Note that recall for Purpose and Reason remain low (trailing by roughly 20% as compared to the three preceding classes), and that while the overall accuracy for the class Reason drops, recall improves dramatically over the default parameter trial.

4.9 One-Vs-All (9 Features, Automatic Argument Boundaries)

When adding the semantic features extracted from the SPADE discourse structures, we observe a small increase in classification accuracy for the classes Concession and Condition, and slight decreases for the remaining classes (again discounting the figures for Result). The trend here mirrors that of the one-vs-one trials when using automatic argument boundaries. That is, a strong indication of interference between distributional patterns introduced by the heuristic and those patterns inferred by the SPADE semantic tags.

4.9.1 Default Parameters

Table 4.23. One-vs-All, 9 Features, Automatic ABD (Accuracy)

Cause	Concession	Condition	Purpose	Reason	Result
57.88	66.07	65.13	56.32	55.13	69.90

Table 4.24. One-vs-All, 9 Features, Automatic ABD (Precision/Recall (f-measure))

Cause	Concession	Condition	Purpose	Reason	Result
61.49/61.07	66.67/77.12	64.86/62.07	63.04/42.03	63.77/35.48	0.00/0.00
(61.28)	(71.52)	(63.44)	(50.42)	(45.58)	(0.00)

As in the trials lacking SPADE data, we observe a dramatic drop in recall for Purpose and Reason. However, we would not expect adding SPADE data, which tags relations at a much coarser semantic level, to improve this. Indeed, Contengency relations with lower semantic distance from other classes are *more* likely to be subsumed by a superordinating relation in the RST subset used by SPADE.

4.9.2 Grid-Search Parameters

As in previous trials, tuned parameters lead to higher classification accuracies for most classes. This time (again discounting Result) there is a decrease in accuracy for Concession, providing some further incidental support for the idea that such results are a consequence of the limited size of the test data being used, and not a manifestation of some inherent feature associated with the semantic argument.

Table 4.25: One-vs-All, 9 Features, Automatic ABD, Trained Parameters (Accuracy)

Cause	Concession	Condition	Purpose	Reason	Result
62.27	63.54	69.75	57.47	58.12	66.02

Table 4.26: One-vs-All, 9 Features, Auto. ABD, Trained Params (Precision/Recall (f-measure))

Cause	Concession	Condition	Purpose	Reason	Result
64.95/67.11	66.05/69.93	66.92/75.00	63.11/47.10	59.84/61.29	40.91/29.03
(66.02)	(67.94)	(70.72)	(53.94)	(60.56)	(33.96)

Here again we see that the cross-validated parameter training procedure favors balance of precision and recall over accuracy, noting a similar increase in the class Reason as in the previous experiments. The continued inability of the classification procedure to select a majority of the test examples associated with Purpose (for which recall has been below or near 50% in all experiments) suggests poor separability resulting from either an imbalance in the syntactic patterns of the test data (unlikely, due to the randomized shuffle of the original corpus prior to the train/test split) or (more reasonably) the lowest semantic distance between this and the other Contingency classes when labeled with SPADE discourse parses.

4.10 One-Vs-All (7 Features, Manual Argument Boundaries)

We continue to observe parallels with the one-vs-one trials when examining the one-vs-all trials using the "gold standard" manual argument boundaries. Again, accuracies are noticeably lower than those associated with the trials using features generated using the automatic heuristic, but remaining above the baseline.

4.10.1 Default Parameters

Table 4.27. One-vs-All, 7 Features, Manual ABD (Accuracy)

Cause	Concession	Condition	Purpose	Reason	Result
52.75	55.59	63.45	53.26	50.85	70.39

Table 4.28. One-vs-All, 7 Features, Manual ABD (Precision/Recall (f-measure))

Cause	Concession	Condition	Purpose	Reason	Result
55.10/78.48	56.94/80.39	57.84/92.24	64.29/26.09	56.00/33.87	100.00/1.16
(64.74)	(66.66)	(71.55)	(37.12)	(42.22)	(2.30)

To understand why these results appear to underperform their automatic ABD counterparts, we must examine the associated precision/recall data in more detail. As before, we see a distinct pattern, in which recall is much higher for the classes Cause, Concession, and Condition than for Purpose and Result, which have correspondingly fallen. We notice an interesting phenomenon: for those classes with few or no ambiguous cues (and, presumably, the largest semantic distance from other classes), recall approaches near-human levels, indicating a closer grouping of the examples in the feature space. However, as this grouping is still non-separable from all other class data existing in that space, precision drops accordingly as non-class examples are misclassified.

For those remaining classes exhibiting higher semantic and syntactic ambiguity, recall drops noticeably, while precision remains relatively high; this speaks to a classifier which selects a relatively small subset of data in test set, those which appear unambiguously defined, while remaining examples remain indistinguishable from other classes.

4.10.2 Grid-Search Parameters

Tuned parameters appear to have a negligible effect on this series of trials, with classification accuracies effectively remaining the same. Precision and recall for the highly ambiguous classes remain low, in contrast to previous experiments where recall was boosted significantly in many cases by the tuning process.

Table 4.29. One-vs-All, 7 Features, Manual ABD, Trained Parameters (Accuracy)

Cause	Concession	Condition	Purpose	Reason	Result
52.01	55.23	63.03	54.02	51.71	69.90

Table 4.30: One-vs-All, 7 Features, Manual ABD, Trained Params (Precision/Recall (f-measure))

Cause	Concession	Condition	Purpose	Reason	Result
54.95/67.11 (60.42)	57.36/73.86 (64.58)	57.87/88.79 (70.08)	58.65/44.20 (50.40)	58.21/31.45 (40.84)	50.00/9.68 (16.22)

4.11 One-Vs-All (9 Features, Manual Argument Boundaries)

The following results represent classification trials conducted using the latest version of the feature extraction system. Initially, we had assumed that the use of lexico-syntactic *and* semantic features along with manually detected argument boundaries would yield the highest classification accuracies. As can be seen below, this is not the case. However, as noted in the one-vs-one trials, the reason for this may not be trivial. While the automatically detected argument boundaries lead to higher classification accuracies using the corpus we developed, this performance may be a consequence of patterns introduced by the heuristic itself, patterns that may not hold in other corpora, or more generally in open text.

4.11.1 Default Parameters

Table 4.31. One-vs-All, 9 Features, Manual ABD (Accuracy)

Cause	Concession	Condition	Purpose	Reason	Result
52.38	59.21	62.61	53.26	53.42	69.42

Table 4.32. One-vs-All, 9 Features, Manual ABD (Precision/Recall (f-measure))

Cause	Concession	Condition	Purpose	Reason	Result
55.62/63.09	60.99/72.55	57.22/92.24	57.27/45.65	61.19/33.06	0/0
(59.12)	(66.26)	(70.62)	(50.80)	(42.92)	(N/A)

4.11.2 Grid-Search Parameters

While the accuracy of a particular classification may not change significantly between the default parameter trial and the tuned parameter trial, the actual classification results can vary significantly, as evidenced by the often substantial shifts in precision and recall.

Table 4.33. One-vs-All, 9 Features, Manual ABD, Trained Parameters (Accuracy)

Cause	Concession	Condition	Purpose	Reason	Result
54.21	56.32	63.03	54.41	54.27	69.90

Such examination can help guide future trials, as preliminary evidence can be misleading, particularly when dealing with multi-class data; for example, note the high classification accuracy for RESULT in the previous trial, in spite of the fact

Table 4.34: One-vs-All, 9 Features, Manual ABD, Trained Parameters (Precision/Recall (f-measure))

Cause	Concession	Condition	Purpose	Reason	Result
57.14/64.43 (60.56)	58.16/74.51 (65.32)	57.69/90.52 (70.46)	56.74/57.97 (57.34)	61.97/35.48 (45.12)	50.00/6.45 (11.42)

that no “true positives” were selected. Grid-search parameter selection improves this result (albeit marginally).

4.12 Discussion

In the preceding experiments, we see a number of interesting phenomena related both to the lexico-syntactic and semantic feature set selected for the SVM trials, and to the argument boundary selection procedures that provide a basis for the extracted feature values.

The results obtained from the trials conducted using features extracted with the automatic boundary detection heuristic are not conclusive, even though they demonstrate consistently higher accuracies than those obtained with the manual argument boundaries. Examination of the corpus and the extracted features themselves, which contain numerous instances of missing or incorrectly intepreted features, suggest that this somewhat naive approach cannot provide a legitimate account of the semantic spaces occupied by the contingency relations, but rather imposes patterns derived from its own encoding on the data.

An examination of the trials in which manual argument boundaries were used indicates that the introduction of “gold standard” arguments eliminates this flaw. However, it also serves to further illustrate the difficulty of classifying these relations. Trials using manual ABD consistently result in classification accuracies marginally above the associated baseline. This is a consequence both of the lack of adequate semantic features in the feature vector and the limited size of the training corpus. As per the results given in Table 4.18, it appears the provided features which encode the full discourse structure associated with a sentence may be too broad an approach; the discourse structures themselves can vary significantly based on the inclusion of arguments not relevant to our task, such as attributions, even when the syntactic and semantic organization related to the contingency are similar or identical.

Classification trials involve a balance of precision and recall depending on the desired output (for example, to what degree we are willing to tolerate false positives). As the default parameter trials using automatic ABD show, relatively high accuracies can be achieved when we build classifiers using feature data that depends directly on syntactic patterns (as the automatic ABD heuristic has no method for ensuring that the verb phrases located indeed correspond to the actual semantic arguments and argument order). When we introduce manually tagged argument boundaries, classification accuracies seem to better reflect our understanding of the semantic spaces occupied by the arguments.

The *contingency relations* are highly ambiguous, with the interpretation of some adverbial cues depending largely on our perception of the situation being described in a particular passage. In the immediate context of an isolated pair of phrases, we may in some cases assign one interpretation (and, subsequently, relation) that is not borne out in the remaining discourse. When the semantics of a particular argument are unambiguous, and when the associated syntactic patterns correspond to a direct realization of those arguments, classical syntactic approaches may suffice. However, as we move into more finely grained classes, and semantic and syntactic boundaries become blurred (due, in the former, to overlap in the associated semantic spaces, and in the latter, to overlap in cue phrases), we must look to more sophisticated methods for generating and processing hierarchical semantic constructs within the discourse (in addition to the associated syntactic realizations) to perform class selection.

For now, such an examination is limited by the availability of large, appropriately tagged corpora, and by the relative dearth of tools for the creation of semantic parses. However, as these tools become available, and as learning procedures are expanded to include steps such as generalized argument boundary detection (giving us the ability to construct large corpora of paired arguments across unlimited text spans), we will have the facilities to build more complete classification tools relying less on purely syntactic processing and more on shallow semantic features.

CHAPTER FIVE

Conclusion

5.1 Overview

In this thesis we describe a unique approach to the task of classifying a finely grained and highly ambiguous set of discourse relations, the *contingency relations.* This investigation demonstrates that providing an adequate account of semantic structure and distribution of the set of contingency relations for application in a supervised machine learning environment is a feasible but complex task. Even when examining the *contingency relations* in a context limited to one or two sentences drawn from a given discourse where each relation is explicitly cued, achieving acceptable levels of classification can be difficult. The lexico-syntactic and semantic features developed and applied here represent an initial step toward adequately addressing the semantic spaces occupied by these relations.

Learning procedures in general (applied to tasks not related to semantic or other forms of text processing) typically utilize a large number of features, both manually selected and automatically generated, to enhance classification accuracy. Frequently, such features are generated algorithmically or based on an expected statistical distribution of certain lexical factors in the text. In these experiments, a detailed account of the reasoning for the inclusion of each feature is provided, in addition to a preliminary exploration of feature contribution. Second, learning procedures – even those utilizing Support Vectors – are typically trained and tested on massive corpora to avoid problems due to irregularities in distribution and matching patterns that can lead to overfitting of the learned model. We have shown that SVM models trained on carefully annotated data drawn from a compact corpus and tuned via cross-validation consistently out-perform the baseline on the highly ambiguous set of relations used.

The variability in the results obtained is not unexpected, as the current system remains relatively semantically shallow and operates on an extremely limited corpus. Additionally, the creation of large corpora where manual annotations (such as those for argument boundaries) are needed can require prohibitive amounts of time to create. As the corpus was selected to represent a wide variety of syntactic forms (including but not limited to canonical and non-canonical structures, intra- and inter-sentential argument structures), argument structures (including representative example of lexically separate arguments, arguments with lexical overlap, and argument inclusion), and a broad range of cue phrases expected to typically appear with high frequency in open text, we expect that the performance of this system will generalize to other corpora, in certain cases improving due to better distributions of arguments underrepresented in these experiments.

5.2 Future Work

Future work will include applications of Support Vector learning to argument boundary detection and more intelligent methods for adjoining syntactic trees. Moreover, we intend to expand the feature set with semantic and pragmatic knowledge to capture intentions, beliefs, and contextual information such as discourse relations in a large context window surrounding the target instance. The intuition behind this idea is that some discourse relations may depend on previously occurring relations in discourse.

Argument boundary detection proved a significant obstacle in the research presented here. Ideally, we desire a system where trained models for argument boundary detection could be applied to domain specific corpora without significant loss of accuracy. In the current model, we are limited to a boundary detection procedure which relies on heuristics (a non-adaptive and semantically empty approach), or one which depends on massive expenditures of time on the part of human annotators to tag the corpus being used. The use of machine learning procedures would provide a good balance of accuracy and efficiency. A suitably general training corpus, once annotated with argument boundaries for the purpose of constructing a trained model, would allow us to quickly and efficiently run the ABD procedure on any available corpus with near-human levels of accuracy. A corpus manually annotated with discourse structures (such as RST-DT) would further enhance the procedure, as we could use manually verified text spans corresponding to satellites of a relation to provide further support to the procedure.

Experiments suggest that when attempting semantic argument classification in open text, a small number of carefully selected features can be sufficient for many types of data (Joachims 1999). In practice, such feature selection can be very difficult, as novel features frequently require adjustments to the available data representation and increase the size of the battery of tests required to appropriately measure feature contribution.

The features described and used in this thesis were developed partly based on prior experimental work in the extraction of syntactic features from automatically annotated trees and the associations between syntactic realizations and underlying semantic arguments (Gildea and Jurafsky 2002). Additional elements of the feature vector were developed by examining patterns present within and across parallel annotations of the available corpus with parts of speech, grammatical roles, and discourse structures. However, choosing "good" feature sets can require many iterations and extensive examination of the available data, both for developing heuristics to aid feature selection in the future, and for providing appropriate justifications for the present features. The current system, having undergone four major feature (or feature-related) revisions (respectively, for feature representation format, argument subtree traversal, addition of semantic features 8 and 9, and replacement of the automatic argument boundary detection procedure), requires further examination and modification.

Examination of the feature magnitude values indicates that in their present form, a number of the features (in particular, features 4-7) extract values that appear to be too specific to yield useful patterns for developing learned rules. Feature 7 (the in-order traversal of grammatical roles) could be updated to provide a more specific account of the ordering of particular roles, for example passive and active verb phrases. Feature 6 will most likely be removed in future revisions of the system, as even in large corpora there is a relative dearth of sentence repetition, which is all this feature appears to capture. Features 4 and 5 (respectively, the subordinate and main argument in-order traversals) could each be separated into two features, one representing the path from the sentence head to the feature head, and one representing the traversal of the lone argument. Furthermore, these features would benefit from a generalization procedure that would take constructs such as noun compounds, tailing noun phrases, and in certain cases adjectival phrases, and aggregating these into single tag representations for insertion into the feature. This would have the effect of reducing the number of unique feature values without eliminating relevant parts of the syntactic realization with respect to the associated semantic class.

A major aspect of this research was the construction of the corpus. Eugene Charniak notes that the creation of a corpus for test purposes of a specific system can be a prohibitive task, one which researchers will avoid both due to time constraints and the lack of a comparative measure with existing systems (Charniak 1997). We came to appreciate this several months into the creation of the corpus used here. Even with the aid of Perl scripts for cue pattern matching, the location of relevant examples encoding the requisite semantic arguments and subsequent manual tagging of argument boundaries was a laborious and time-intensive task. A benefit of this approach was that the resulting data set remained virtually noise-free.

In the future, we intend to train and test the system on the RST-DT corpus used for a variety of other projects, including that which motivated the development of SPADE (Carlson et al. 2003). Benefits of using such a corpus include the use of discourse structures manually annotated by multiple sources, along with a measure of inter-annotator agreement. Such a corpus allows for more consistent methods of argument boundary detection, expansion of the classification procedure to other semantic classes, and provides for performance measurements that can be directly compared to similar semantic classification tools.

5.3 Summary

This work represents a preliminary examination of the semantic spaces occupied by the *contingency relations*, a set of relations previously unexplored in the literature. It provides a novel approach to the problem of classifying these relations, and has a strong theoretical basis in previous work on syntactic and semantic feature extraction. Future applications of research in computational semantics, including tasks such as Question Answering and Text Summarization, will depend directly on the availability of such accounts.

APPENDIX

APPENDIX A

Full Experimental Trial Data

A.1 One-Vs-One, 7 Features, Automatic ABD

Cause vs. Concession	Default Parameters	Trained Parameters c1,g0.125
Accuracy	58.6093% (177/302)	59.6026% (180/302)
Mean squared error	0.413907	0.403974
Squared correlation coefficient	0.0294865	0.0367195
Confusion matrix	tp=83 fp=59 fn=66 tn=94	tp=83 fp=56 fn=66 tn=97
Precision	0.584507042253521	0.597122302158273
Recall	0.557046979865772	0.557046979865772

Cause vs. Condition	Default Parameters	Trained Parameters c32,g0.5
Accuracy	78.4906% (208/265)	84.1509% (223/265)
Mean squared error	0.860377	0.633962
Squared correlation coefficient	0.32623	0.459719
Confusion matrix	tp=114 fp=22 fn=35 tn=94	tp=128 fp=21 fn=21 tn=95
Precision	0.838235294117647	0.859060402684564
Recall	0.76510067114094	0.859060402684564

Cause vs. Purpose	Default Parameters	Trained Parameters c32,g1.0
Accuracy	60.9756% (175/287)	70.3833% (202/287)
Mean squared error	3.5122	2.66551
Squared correlation coefficient	0.0510922	0.165037
Confusion matrix	tp=122 fp=85 fn=27 tn=53	tp=109 fp=45 fn=40 tn=93
Precision	0.589371980676328	0.707792207792208
Recall	0.818791946308725	0.731543624161074

Cause vs. Reason	Default Parameters	Trained Parameters c32,g1.0
Accuracy	54.5788% (149/273)	58.6081% (160/273)
Mean squared error	7.2674	6.62271
Squared correlation coefficient	0.000321556	0.0224194
Confusion matrix	tp=144 fp=119 fn=5 tn=5	tp=111 fp=75 fn=38 tn=49
Precision	0.547528517110266	0.596774193548387
Recall	0.966442953020134	0.74496644295302

Cause vs. Result	Default Parameters	Trained Parameters c32,g0.25
Accuracy	70.6161% (149/211)	73.4597% (155/211)
Mean squared error	7.34597	6.63507
Squared correlation coefficient	nan	0.0723812
Confusion matrix	tp=149 fp=62 fn=0 tn=0	tp=138 fp=45 fn=11 tn=17
Precision	0.706161137440758	0.754098360655738
Recall	1	0.926174496644295

Concession vs. Condition	Default Parameters	Trained Parameters c32,g0.5
Accuracy	83.2714% (224/269)	88.4758% (238/269)
Mean squared error	0.167286	0.115242
Squared correlation coefficient	0.442364	0.585743
Confusion matrix	tp=125 fp=17 fn=28 tn=99	tp=137 fp=15 fn=16 tn=101
Precision	0.880281690140845	0.901315789473684
Recall	0.816993464052288	0.895424836601307

Concession vs. Purpose	Default Parameters	Trained Parameters c32,g1.0
Accuracy	66.323% (193/291)	68.3849% (199/291)
Mean squared error	1.34708	1.2646
Squared correlation coefficient	0.115645	0.134727
Confusion matrix	tp=133 fp=78 fn=20 tn=60	tp=105 fp=44 fn=48 tn=94
Precision	0.630331753554502	0.704697986577181
Recall	0.869281045751634	0.686274509803922

Concession vs. Reason	Default Parameters	Trained Parameters c4,g0.5
Accuracy	59.5668% (165/277)	60.2888% (167/277)
Mean squared error	3.63899	3.57401
Squared correlation coefficient	0.0452779	0.0377337
Confusion matrix	tp=151 fp=110 fn=2 tn=14	tp=143 fp=100 fn=10 tn=24
Precision	0.578544061302682	0.588477366255144
Recall	0.986928104575163	0.934640522875817

Concession vs. Result	Default Parameters	Trained Parameters c32,g0.25
Accuracy	71.6279% (154/215)	76.7442% (165/215)
Mean squared error	4.53953	3.72093
Squared correlation coefficient	0.00986823	0.130313
Confusion matrix	tp=152 fp=60 fn=1 tn=2	tp=148 fp=45 fn=5 tn=17
Precision	0.716981132075472	0.766839378238342
Recall	0.993464052287582	0.967320261437909

Condition vs. Purpose	Default Parameters	Trained Parameters c8,g0.25
Accuracy	83.0709% (211/254)	84.6457% (215/254)
Mean squared error	0.169291	0.153543
Squared correlation coefficient	0.433412	0.477241
Confusion matrix	tp=93 fp=20 fn=23 tn=118	tp=97 fp=20 fn=19 tn=118
Precision	0.823008849557522	0.829059829059829
Recall	0.801724137931034	0.836206896551724

Condition vs. Reason	Default Parameters	Trained Parameters c32,g1.0
Accuracy	74.1667% (178/240)	88.75% (213/240)
Mean squared error	1.03333	0.45
Squared correlation coefficient	0.250382	0.601766
Confusion matrix	tp=99 fp=45 fn=17 tn=79	tp=105 fp=16 fn=11 tn=108
Precision	0.6875	0.867768595041322
Recall	0.853448275862069	0.905172413793103

Condition vs. Result	Default Parameters	Trained Parameters c8,g0.5
Accuracy	70.7865% (126/178)	79.7753% (142/178)
Mean squared error	2.62921	1.82022
Squared correlation coefficient	0.0973682	0.289218
Confusion matrix	tp=114 fp=50 fn=2 tn=12	tp=107 fp=27 fn=9 tn=35
Precision	0.695121951219512	0.798507462686567
Recall	0.982758620689655	0.922413793103448

Purpose vs. Reason	Default Parameters	Trained Parameters c32,g1.0
Accuracy	60.3053% (158/262)	64.1221% (168/262)
Mean squared error	0.396947	0.358779
Squared correlation coefficient	0.0403059	0.0776628
Confusion matrix	tp=92 fp=58 fn=46 tn=66	tp=105 fp=61 fn=33 tn=63
Precision	0.613333333333333	0.632530120481928
Recall	0.666666666666667	0.760869565217391

Purpose vs. Result	Default Parameters	Trained Parameters c16,g0.5
Accuracy	69% (138/200)	70.5% (141/200)
Mean squared error	1.24	1.18
Squared correlation coefficient	nan	0.0417418
Confusion matrix	tp=138 fp=62 fn=0 tn=0	tp=127 fp=48 fn=11 tn=14
Precision	0.69	0.725714285714286
Recall	1	0.920289855072464

Reason vs. Result	Default Parameters	Trained Parameters c32,g1.0
Accuracy	66.6667% (124/186)	62.9032% (117/186)
Mean squared error	0.333333	0.370968
Squared correlation coefficient	nan	0.0262295
Confusion matrix	tp=124 fp=62 fn=0 tn=0	tp=90 fp=35 fn=34 tn=27
Precision	0.666666666666667	0.72
Recall	1	0.725806451612903

A.2 One-Vs-One, 7 Features, Manual ABD

Cause vs. Concession	Default Parameters	Trained Params c32,g0.125
Accuracy	55.298% (167/302)	53.6424% (162/302)
Mean squared error	0.44702	0.463576
Squared correlation coefficient	0.0111237	0.0053932
Confusion matrix	tp=79 fp=65 fn=70 tn=88	tp=83 fp=74 fn=66 tn=79
Precision	0.548611111111111	0.528662420382166
Recall	0.530201342281879	0.557046979865772

Cause vs. Condition	Default Parameters	Trained Parameters c16,g1
Accuracy	65.6604% (174/265)	66.4151% (176/265)
Mean squared error	1.37358	1.3434
Squared correlation coefficient	0.150988	0.150388
Confusion matrix	tp=71 fp=13 fn=78 tn=103	tp=76 fp=16 fn=73 tn=100
Precision	0.845238095238095	0.826086956521739
Recall	0.476510067114094	0.51006711409396

Cause vs. Purpose	Default Parameters	Trained Parameters c16,g1
Accuracy	58.885% (169/287)	58.1882% (167/287)
Mean squared error	3.70035	3.76307
Squared correlation coefficient	0.0396403	0.0254094
Confusion matrix	tp=133 fp=102 fn=16 tn=36	tp=101 fp=72 fn=48 tn=66
Precision	0.565957446808511	0.583815028901734
Recall	0.89261744966443	0.677852348993289

Cause vs. Reason	Default Parameters	Trained Parameters c32,g0.25
Accuracy	54.9451% (150/273)	54.2125% (148/273)
Mean squared error	7.20879	7.32601
Squared correlation coefficient	0.00145851	0.00212844
Confusion matrix	tp=146 fp=120 fn=3 tn=4	tp=113 fp=89 fn=36 tn=35
Precision	0.548872180451128	0.559405940594059
Recall	0.979865771812081	0.758389261744966

Cause vs. Result	Default Parameters	Trained Parameters c8,g0.125
Accuracy	70.6161% (149/211)	69.1943% (146/211)
Mean squared error	7.34597	7.70142
Squared correlation coefficient	nan	0.00600155
Confusion matrix	tp=149 fp=62 fn=0 tn=0	tp=146 fp=62 fn=3 tn=0
Precision	0.706161137440758	0.701923076923077
Recall	1	0.979865771812081

Concession vs. Condition	Default Parameters	Trained Parameters c8,g0.5
Accuracy	69.8885% (188/269)	69.5167% (187/269)
Mean squared error	0.301115	0.304833
Squared correlation coefficient	0.212161	0.183333
Confusion matrix	tp=84 fp=12 fn=69 tn=104	tp=90 fp=19 fn=63 tn=97
Precision	0.875	0.825688073394495
Recall	0.549019607843137	0.588235294117647

Concession vs. Purpose	Default Parameters	Trained Parameters c8,g0.5
Accuracy	56.0137% (163/291)	61.1684% (178/291)
Mean squared error	1.75945	1.55326
Squared correlation coefficient	0.0141882	0.0475676
Confusion matrix	tp=137 fp=112 fn=16 tn=26	tp=114 fp=74 fn=39 tn=64
Precision	0.550200803212851	0.606382978723404
Recall	0.895424836601307	0.745098039215686

Concession vs. Reason	Default Parameters	Trained Params c32,g0.0625
Accuracy	55.2347% (153/277)	56.3177% (156/277)
Mean squared error	4.02888	3.93141
Squared correlation coefficient	0.00606731	0.00896768
Confusion matrix	tp=104 fp=75 fn=49 tn=49	tp=111 fp=79 fn=42 tn=45
Precision	0.581005586592179	0.58421052631579
Recall	0.679738562091503	0.725490196078431

Concession vs. Result	Default Parameters	Trained Parameters c16,g0.5
Accuracy	71.1628% (153/215)	71.1628% (153/215)
Mean squared error	4.61395	4.61395
Squared correlation coefficient	nan	0.0084345
Confusion matrix	tp=153 fp=62 fn=0 tn=0	tp=149 fp=58 fn=4 tn=4
Precision	0.711627906976744	0.719806763285024
Recall	1	0.973856209150327

Condition vs. Purpose	Default Parameters	Trained Parameters c1,g0.25
Accuracy	60.6299% (154/254)	60.2362% (153/254)
Mean squared error	0.393701	0.397638
Squared correlation coefficient	0.0963812	0.0890607
Confusion matrix	tp=106 fp=90 fn=10 tn=48	tp=105 fp=90 fn=11 tn=48
Precision	0.540816326530612	0.538461538461538
Recall	0.913793103448276	0.905172413793103

Condition vs. Reason	Default Parameters	Trained Parameters c2,g0.5
Accuracy	64.1667% (154/240)	67.5% (162/240)
Mean squared error	1.43333	1.3
Squared correlation coefficient	0.114333	0.160412
Confusion matrix	tp=103 fp=73 fn=13 tn=51	tp=104 fp=66 fn=12 tn=58
Precision	0.585227272727273	0.611764705882353
Recall	0.887931034482759	0.896551724137931

Condition vs. Result	Default Parameters	Trained Params c0.5,g0.25
Accuracy	72.4719% (129/178)	72.4719% (129/178)
Mean squared error	2.47753	2.47753
Squared correlation coefficient	0.14741	0.14741
Confusion matrix	tp=116 fp=49 fn=0 tn=13	tp=116 fp=49 fn=0 tn=13
Precision	0.703030303030303	0.703030303030303
Recall	1	1

Purpose vs. Reason	Default Parameters	Trained Parameters c16,g0.25
Accuracy	59.542% (156/262)	64.5038% (169/262)
Mean squared error	0.40458	0.354962
Squared correlation coefficient	0.0350741	0.0816077
Confusion matrix	tp=87 fp=55 fn=51 tn=69	tp=98 fp=53 fn=40 tn=71
Precision	0.612676056338028	0.649006622516556
Recall	0.630434782608696	0.710144927536232

Purpose vs. Result	Default Parameters	Trained Parameters c32,g0.25
Accuracy	69% (138/200)	68% (136/200)
Mean squared error	1.24	1.28
Squared correlation coefficient	nan	7.87213e-05
Confusion matrix	tp=138 fp=62 fn=0 tn=0	tp=134 fp=60 fn=4 tn=2
Precision	0.69	0.690721649484536
Recall	1	0.971014492753623

Reason vs. Result	Default Parameters	Trained Parameters c8,g0.125
Accuracy	66.129% (123/186)	67.7419% (126/186)
Mean squared error	0.33871	0.322581
Squared correlation coefficient	0.000552486	0.0181818
Confusion matrix	tp=121 fp=60 fn=3 tn=2	tp=120 fp=56 fn=4 tn=6
Precision	0.668508287292818	0.681818181818182
Recall	0.975806451612903	0.967741935483871

A.3 One-Vs-One, 9 Features, Automatic ABD

Cause vs. Concession	Default Parameters	Trained Params c0.5,g0.25
Accuracy	57.2848% (173/302)	56.9536% (172/302)
Mean squared error	0.427152	0.430464
Squared correlation coefficient	0.0243457	0.0228147
Confusion matrix	tp=52 fp=32 fn=97 tn=121	tp=49 fp=30 fn=100 tn=123
Precision	0.619047619047619	0.620253164556962
Recall	0.348993288590604	0.328859060402685

Cause vs. Condition	Default Parameters	Trained Parameters c32,g1.0
Accuracy	69.8113% (185/265)	71.3208% (189/265)
Mean squared error	1.20755	1.14717
Squared correlation coefficient	0.148678	0.169141
Confusion matrix	tp=110 fp=41 fn=39 tn=75	tp=120 fp=47 fn=29 tn=69
Precision	0.728476821192053	0.718562874251497
Recall	0.738255033557047	0.805369127516778

Cause vs. Purpose	Default Parameters	Trained Parameters c16,g0.25
Accuracy	63.4146% (182/287)	70.7317% (203/287)
Mean squared error	3.29268	2.63415
Squared correlation coefficient	0.0713032	0.175451
Confusion matrix	tp=97 fp=53 fn=52 tn=85	tp=99 fp=34 fn=50 tn=104
Precision	0.646666666666667	0.744360902255639
Recall	0.651006711409396	0.664429530201342

Cause vs. Reason	Default Parameters	Trained Parameters c32,g0.5
Accuracy	59.707% (163/273)	61.5385% (168/273)
Mean squared error	6.44689	6.15385
Squared correlation coefficient	0.0325332	0.0453936
Confusion matrix	tp=134 fp=95 fn=15 tn=29	tp=114 fp=70 fn=35 tn=54
Precision	0.585152838427948	0.619565217391304
Recall	0.899328859060403	0.76510067114094

Cause vs. Result	Default Parameters	Trained Parameters c4,g0.125
Accuracy	70.6161% (149/211)	68.7204% (145/211)
Mean squared error	7.34597	7.81991
Squared correlation coefficient	nan	0.000365006
Confusion matrix	tp=149 fp=62 fn=0 tn=0	tp=143 fp=60 fn=6 tn=2
Precision	0.706161137440758	0
Recall	1	0

Concession vs. Condition	Default Parameters	Trained Parameters c32,g0.5
Accuracy	73.9777% (199/269)	76.9517% (207/269)
Mean squared error	0.260223	0.230483
Squared correlation coefficient	0.215353	0.297885
Confusion matrix	tp=126 fp=43 fn=27 tn=73	tp=112 fp=21 fn=41 tn=95
Precision	0.745562130177515	0.842105263157895
Recall	0.823529411764706	0.73202614379085

Concession vs. Purpose	Default Parameters	Trained Parameters c32,g0.5
Accuracy	64.2612% (187/291)	67.6976% (197/291)
Mean squared error	1.42955	1.2921
Squared correlation coefficient	0.0808441	0.124809
Confusion matrix	tp=121 fp=72 fn=32 tn=66	tp=104 fp=45 fn=49 tn=93
Precision	0.626943005181347	0.697986577181208
Recall	0.790849673202614	0.679738562091503

Concession vs. Reason	Default Parameters	Trained Parameters c16,g1.0
Accuracy	63.1769% (175/277)	63.1769% (175/277)
Mean squared error	3.31408	3.31408
Squared correlation coefficient	0.0593969	0.0594393
Confusion matrix	tp=126 fp=75 fn=27 tn=49	tp=118 fp=67 fn=35 tn=57
Precision	0.626865671641791	0.637837837837838
Recall	0.823529411764706	0.77124183006536

Concession vs. Result	Default Parameters	Trained Parameters c16,g0.25
Accuracy	71.6279% (154/215)	71.6279% (154/215)
Mean squared error	4.53953	4.53953
Squared correlation coefficient	0.0115315	0.0513436
Confusion matrix	tp=153 fp=61 fn=0 tn=1	tp=135 fp=43 fn=18 tn=19
Precision	0.714953271028037	0.758426966292135
Recall	1	0.882352941176471

Condition vs. Purpose	Default Parameters	Trained Parameters c32,g1.0
Accuracy	62.5984% (159/254)	76.378% (194/254)
Mean squared error	0.374016	0.23622
Squared correlation coefficient	0.0574311	0.273467
Confusion matrix	tp=60 fp=39 fn=56 tn=99	tp=84 fp=28 fn=32 tn=110
Precision	0.606060606060606	0.75
Recall	0.517241379310345	0.724137931034483

Condition vs. Reason	Default Parameters	Trained Parameters c32,g1.0
Accuracy	70.4167% (169/240)	78.3333% (188/240)
Mean squared error	1.18333	0.866667
Squared correlation coefficient	0.173049	0.320927
Confusion matrix	tp=90 fp=45 fn=26 tn=79	tp=91 fp=27 fn=25 tn=97
Precision	0.666666666666667	0.771186440677966
Recall	0.775862068965517	0.78448275862069

Condition vs. Result	Default Parameters	Trained Parameters c16,g0.5
Accuracy	66.2921% (118/178)	70.7865% (126/178)
Mean squared error	3.03371	2.62921
Squared correlation coefficient	0.0199607	0.11448
Confusion matrix	tp=109 fp=53 fn=7 tn=9	tp=94 fp=30 fn=22 tn=32
Precision	0.672839506172839	0.758064516129032
Recall	0.939655172413793	0.810344827586207

Purpose vs. Reason	Default Parameters	Trained Params c32,g0.125
Accuracy	63.7405% (167/262)	67.1756% (176/262)
Mean squared error	0.362595	0.328244
Squared correlation coefficient	0.0730728	0.116085
Confusion matrix	tp=103 fp=60 fn=35 tn=64	tp=97 fp=45 fn=41 tn=79
Precision	0.631901840490798	0.683098591549296
Recall	0.746376811594203	0.702898550724638

Purpose vs. Result	Default Parameters	Trained Parameters c32,g0.5
Accuracy	69% (138/200)	70.5% (141/200)
Mean squared error	1.24	1.18
Squared correlation coefficient	nan	0.0463267
Confusion matrix	tp=138 fp=62 fn=0 tn=0	tp=125 fp=46 fn=13 tn=16
Precision	0.69	0.730994152046784
Recall	1	0.905797101449275

Reason vs. Result	Default Parameters	Trained Parameters c8,g0.25
Accuracy	66.6667% (124/186)	65.5914% (122/186)
Mean squared error	0.333333	0.344086
Squared correlation coefficient	0.0013587	0.00215517
Confusion matrix	tp=123 fp=61 fn=1 tn=1	tp=117 fp=57 fn=7 tn=5
Precision	0.668478260869565	0.672413793103448
Recall	0.991935483870968	0.943548387096774

A.4 One-Vs-One, 9 Features, Manual ABD

Cause vs. Concession	Default Parameters	Trained Parameters c4,g1.0
Accuracy	55.9603% (169/302)	48.6755% (147/302)
Mean squared error	0.440397	0.513245
Squared correlation coefficient	0.0144526	0.000601182
Confusion matrix	tp=88 fp=72 fn=61 tn=81	tp=91 fp=90 fn=58 tn=63
Precision	0.55	0.502762430939227
Recall	0.590604026845638	0.610738255033557

Cause vs. Condition	Default Parameters	Trained Parameters c32,g0.25
Accuracy	66.0377% (175/265)	65.6604% (174/265)
Mean squared error	1.35849	1.37358
Squared correlation coefficient	0.163056	0.138251
Confusion matrix	tp=70 fp=11 fn=79 tn=105	tp=75 fp=17 fn=74 tn=99
Precision	0.864197530864197	0.815217391304348
Recall	0.469798657718121	0.503355704697987

Cause vs. Purpose	Default Parameters	Trained Params c32,g0.0625
Accuracy	60.9756% (175/287)	64.1115% (184/287)
Mean squared error	3.5122	3.22997
Squared correlation coefficient	0.0588876	0.0786144
Confusion matrix	tp=132 fp=95 fn=17 tn=43	tp=100 fp=54 fn=49 tn=84
Precision	0.581497797356828	0.649350649350649
Recall	0.885906040268456	0.671140939597315

Cause vs. Reason	Default Parameters	Trained Parameters c1,g0.25
Accuracy	59.3407% (162/273)	60.4396% (165/273)
Mean squared error	6.50549	6.32967
Squared correlation coefficient	0.0284279	0.0378265
Confusion matrix	tp=132 fp=94 fn=17 tn=30	tp=131 fp=90 fn=18 tn=34
Precision	0.584070796460177	0.592760180995475
Recall	0.885906040268456	0.879194630872483

Cause vs. Result	Default Parameters	Trained Params c8,g0.0625
Accuracy	70.6161% (149/211)	70.1422% (148/211)
Mean squared error	7.34597	7.46445
Squared correlation coefficient	nan	0.00198146
Confusion matrix	tp=149 fp=62 fn=0 tn=0	tp=148 fp=62 fn=1 tn=0
Precision	0.706161137440758	0.704761904761905
Recall	1	0.993288590604027

Concession vs. Condition	Default Parameters	Trained Params c8,g0.0625
Accuracy	68.0297% (183/269)	70.632% (190/269)
Mean squared error	0.319703	0.29368
Squared correlation coefficient	0.19078	0.226841
Confusion matrix	tp=78 fp=11 fn=75 tn=105	tp=85 fp=11 fn=68 tn=105
Precision	0.876404494382023	0.885416666666667
Recall	0.509803921568627	0.555555555555556

Concession vs. Purpose	Default Parameters	Trained Parameters c8,g0.125
Accuracy	63.2302% (184/291)	64.9485% (189/291)
Mean squared error	1.47079	1.40206
Squared correlation coefficient	0.0848841	0.0870122
Confusion matrix	tp=139 fp=93 fn=14 tn=45	tp=109 fp=58 fn=44 tn=80
Precision	0.599137931034483	0.652694610778443
Recall	0.908496732026144	0.712418300653595

Concession vs. Reason	Default Parameters	Trained Parameters c8,g0.125
Accuracy	64.2599% (178/277)	64.2599% (178/277)
Mean squared error	3.21661	3.21661
Squared correlation coefficient	0.0760009	0.0749397
Confusion matrix	tp=105 fp=51 fn=48 tn=73	tp=107 fp=53 fn=46 tn=71
Precision	0.673076923076923	0.66875
Recall	0.686274509803922	0.699346405228758

Concession vs. Result	Default Parameters	Trained Params c0.125,g0.25
Accuracy	71.1628% (153/215)	71.1628% (153/215)
Mean squared error	4.61395	4.61395
Squared correlation coefficient	nan	nan
Confusion matrix	tp=153 fp=62 fn=0 tn=0	tp=153 fp=62 fn=0 tn=0
Precision	0.711627906976744	0.711627906976744
Recall	1	1

Condition vs. Purpose	Default Parameters	Trained Parameters c1,g0.25
Accuracy	60.6299% (154/254)	61.4173% (156/254)
Mean squared error	0.393701	0.385827
Squared correlation coefficient	0.0963812	0.101503
Confusion matrix	tp=106 fp=90 fn=10 tn=48	tp=105 fp=87 fn=11 tn=51
Precision	0.540816326530612	0.546875
Recall	0.913793103448276	0.905172413793103

Condition vs. Reason	Default Parameters	Trained Params c16,g0.125
Accuracy	63.3333% (152/240)	66.6667% (160/240)
Mean squared error	1.46667	1.33333
Squared correlation coefficient	0.107156	0.146054
Confusion matrix	tp=104 fp=76 fn=12 tn=48	tp=103 fp=67 fn=13 tn=57
Precision	0.577777777777778	0.605882352941176
Recall	0.896551724137931	0.887931034482759

Condition vs. Result	Default Parameters	Trained Params c4,g0.0625
Accuracy	72.4719% (129/178)	73.0337% (130/178)
Mean squared error	2.47753	2.42697
Squared correlation coefficient	0.14741	0.159717
Confusion matrix	tp=116 fp=49 fn=0 tn=13	tp=116 fp=48 fn=0 tn=14
Precision	0.703030303030303	0.707317073170732
Recall	1	1

Purpose vs. Reason	Default Parameters	Trained Parameters c8,g0.125
Accuracy	61.4504% (161/262)	60.3053% (158/262)
Mean squared error	0.385496	0.396947
Squared correlation coefficient	0.0510351	0.0401663
Confusion matrix	tp=89 fp=52 fn=49 tn=72	tp=93 fp=59 fn=45 tn=65
Precision	0.631205673758865	0.611842105263158
Recall	0.644927536231884	0.673913043478261

Purpose vs. Result	Default Parameters	Trained Params c32,g0.125
Accuracy	69% (138/200)	67% (134/200)
Mean squared error	1.24	1.32
Squared correlation coefficient	nan	2.46057e-05
Confusion matrix	tp=138 fp=62 fn=0 tn=0	tp=131 fp=59 fn=7 tn=3
Precision	0.69	0.689473684210526
Recall	1	0.949275362318841

Reason vs. Result	Default Parameters	Trained Params c32,g0.0625
Accuracy	67.2043% (125/186)	66.6667% (124/186)
Mean squared error	0.327957	0.333333
Squared correlation coefficient	0.00819672	0.0117647
Confusion matrix	tp=123 fp=60 fn=1 tn=2	tp=116 fp=54 fn=8 tn=8
Precision	0.672131147540984	0.682352941176471
Recall	0.991935483870968	0.935483870967742

A.5 One-Vs-All, 7 Features, Automatic ABD

Cause	Default Parameters	Trained Parameters c32,g0.5
Accuracy	57.5092% (157/273)	66.3004% (181/273)
Mean squared error	0.424908	0.336996
Squared correlation coefficient	0.0171601	0.10203
Confusion matrix	tp=103 fp=70 fn=46 tn=54	tp=104 fp=47 fn=45 tn=77
Precision	0.595375722543353	0.688741721854305
Recall	0.691275167785235	0.697986577181208

Concession	Default Parameters	Trained Parameters c8,g1
Accuracy	65.343% (181/277)	67.148% (186/277)
Mean squared error	1.38628	1.31408
Squared correlation coefficient	0.0841843	0.108351
Confusion matrix	tp=123 fp=66 fn=30 tn=58	tp=117 fp=55 fn=36 tn=69
Precision	0.650793650793651	0.680232558139535
Recall	0.803921568627451	0.764705882352941

Condition	Default Parameters	Trained Parameters c32,g0.5
Accuracy	63.8655% (152/238)	74.7899% (178/238)
Mean squared error	3.2521	2.26891
Squared correlation coefficient	0.0776764	0.247461
Confusion matrix	tp=77 fp=47 fn=39 tn=75	tp=90 fp=34 fn=26 tn=88
Precision	0.620967741935484	0.725806451612903
Recall	0.663793103448276	0.775862068965517

Purpose	Default Parameters	Trained Parameters c16,g1
Accuracy	57.0881% (149/261)	60.9195% (159/261)
Mean squared error	6.8659	6.25287
Squared correlation coefficient	0.0250882	0.0513256
Confusion matrix	tp=64 fp=38 fn=74 tn=85	tp=76 fp=40 fn=62 tn=83
Precision	0.627450980392157	0.655172413793103
Recall	0.463768115942029	0.550724637681159

Reason	Default Parameters	Trained Parameters c32,g0.5
Accuracy	58.9744% (138/234)	57.6923% (135/234)
Mean squared error	10.2564	10.5769
Squared correlation coefficient	0.0403751	0.0223642
Confusion matrix	tp=58 fp=30 fn=66 tn=80	tp=76 fp=51 fn=48 tn=59
Precision	0.659090909090909	0.598425196850394
Recall	0.467741935483871	0.612903225806452

Result	Default Parameters	Trained Parameters c32,g0.5
Accuracy	69.4175% (143/206)	60.6796% (125/206)
Mean squared error	11.0097	14.1553
Squared correlation coefficient	7.35679e-05	0.00190437
Confusion matrix	tp=1 fp=2 fn=61 tn=142	tp=19 fp=38 fn=43 tn=106
Precision	0.333333333333333	0.333333333333333
Recall	0.0161290322580645	0.306451612903226

A.6 One-Vs-All, 7 Features, Manual ABD

Cause	Default Parameters	Trained Parameters c16,g1
Accuracy	52.7473% (144/273)	52.0147% (142/273)
Mean squared error	0.472527	0.479853
Squared correlation coefficient	0.000281165	0.000108249
Confusion matrix	tp=108 fp=88 fn=41 tn=36	tp=100 fp=82 fn=49 tn=42
Precision	0.551020408163265	0.549450549450549
Recall	0.724832214765101	0.671140939597315

Concession	Default Parameters	Trained Parameters c2,g0.25
Accuracy	55.5957% (154/277)	55.2347% (153/277)
Mean squared error	1.77617	1.79061
Squared correlation coefficient	0.00418653	0.00450035
Confusion matrix	tp=123 fp=93 fn=30 tn=31	tp=113 fp=84 fn=40 tn=40
Precision	0.569444444444444	0.573604060913706
Recall	0.803921568627451	0.738562091503268

Condition	Default Parameters	Trained Parameters c16,g0.5
Accuracy	63.4454% (151/238)	63.0252% (150/238)
Mean squared error	3.28992	3.32773
Squared correlation coefficient	0.115653	0.098886
Confusion matrix	tp=107 fp=78 fn=9 tn=44	tp=103 fp=75 fn=13 tn=47
Precision	0.578378378378378	0.578651685393258
Recall	0.922413793103448	0.887931034482759

Purpose	Default Parameters	Trained Parameters c2,g1
Accuracy	53.2567% (139/261)	54.023% (141/261)
Mean squared error	7.47893	7.35632
Squared correlation coefficient	0.014278	0.00888237
Confusion matrix	tp=36 fp=20 fn=102 tn=103	tp=61 fp=43 fn=77 tn=80
Precision	0.642857142857143	0.586538461538462
Recall	0.260869565217391	0.442028985507246

Reason	Default Parameters	Trained Parameters c2,g0.25
Accuracy	50.8547% (119/234)	51.7094% (121/234)
Mean squared error	12.2863	12.0726
Squared correlation coefficient	0.00171394	0.00438431
Confusion matrix	tp=42 fp=33 fn=82 tn=77	tp=39 fp=28 fn=85 tn=82
Precision	0.56	0.582089552238806
Recall	0.338709677419355	0.314516129032258

Result	Default Parameters	Trained Parameters c4,g0.5
Accuracy	70.3883% (145/206)	69.9029% (144/206)
Mean squared error	10.6602	10.835
Squared correlation coefficient	0.0113297	0.0116464
Confusion matrix	tp=1 fp=0 fn=61 tn=144	tp=6 fp=6 fn=56 tn=138
Precision	1	0.5
Recall	0.0161290322580645	0.0967741935483871

A.7 One-Vs-All, 9 Features, Automatic ABD

Cause	Default Parameters	Trained Params c32,g0.125
Accuracy	57.8755% (158/273)	62.2711% (170/273)
Mean squared error	0.421245	0.377289
Squared correlation coefficient	0.0227898	0.0559888
Confusion matrix	tp=91 fp=57 fn=58 tn=67	tp=100 fp=54 fn=49 tn=70
Precision	0.614864864864865	0.649350649350649
Recall	0.610738255033557	0.671140939597315

Concession	Default Parameters	Trained Parameters c2,g1
Accuracy	66.065% (183/277)	63.5379% (176/277)
Mean squared error	1.3574	1.45848
Squared correlation coefficient	0.0935545	0.0666338
Confusion matrix	tp=118 fp=59 fn=35 tn=65	tp=107 fp=55 fn=46 tn=69
Precision	0.666666666666667	0.660493827160494
Recall	0.77124183006536	0.699346405228758

Condition	Default Parameters	Trained Parameters c32,g0.25
Accuracy	65.1261% (155/238)	69.7479% (166/238)
Mean squared error	3.13866	2.72269
Squared correlation coefficient	0.0909651	0.1593
Confusion matrix	tp=72 fp=39 fn=44 tn=83	tp=87 fp=43 fn=29 tn=79
Precision	0.648648648648649	0.669230769230769
Recall	0.620689655172414	0.75

Purpose	Default Parameters	Trained Params c0.5,g0.25
Accuracy	56.3218% (147/261)	57.4713% (150/261)
Mean squared error	6.98851	6.8046
Squared correlation coefficient	0.0225961	0.027397
Confusion matrix	tp=58 fp=34 fn=80 tn=89	tp=65 fp=38 fn=73 tn=85
Precision	0.630434782608696	0.631067961165049
Recall	0.420289855072464	0.471014492753623

Reason	Default Parameters	Trained Parameters c32,g0.5
Accuracy	55.1282% (129/234)	58.1197% (136/234)
Mean squared error	11.2179	10.4701
Squared correlation coefficient	0.0194963	0.0266352
Confusion matrix	tp=44 fp=25 fn=80 tn=85	tp=NA fp=NA fn=NA tn=NA
Precision	0.63768115942029	NA
Recall	0.354838709677419	NA

Result	Default Parameters	Trained Parameters c32,g0.5
Accuracy	69.9029% (144/206)	66.0194% (136/206)
Mean squared error	10.835	12.233
Squared correlation coefficient	nan	0.0150914
Confusion matrix	tp=0 fp=0 fn=62 tn=144	tp=18 fp=26 fn=44 tn=118
Precision	NA	0.409090909090909
Recall	NA	0.290322580645161

A.8 One-Vs-All, 9 Features, Manual ABD

Cause	Default Parameters	Trained Params c16,g0.125
Accuracy	52.381% (143/273)	54.2125% (148/273)
Mean squared error	0.47619	0.457875
Squared correlation coefficient	0.000712463	0.00424334
Confusion matrix	tp=94 fp=75 fn=55 tn=49	tp=96 fp=72 fn=53 tn=52
Precision	0.556213017751479	0.571428571428571
Recall	0.630872483221476	0.644295302013423

Concession	Default Parameters	Trained Params c16,g0.125
Accuracy	59.2058% (164/277)	56.3177% (156/277)
Mean squared error	1.63177	1.74729
Squared correlation coefficient	0.0256559	0.00839344
Confusion matrix	tp=111 fp=71 fn=42 tn=53	tp=114 fp=82 fn=39 tn=42
Precision	0.60989010989011	0.581632653061224
Recall	0.725490196078431	0.745098039215686

Condition	Default Parameters	Trained Parameters c0.5,g0.5
Accuracy	62.605% (149/238)	63.0252% (150/238)
Mean squared error	3.36555	3.32773
Squared correlation coefficient	0.10553	0.104265
Confusion matrix	tp=107 fp=80 fn=9 tn=42	tp=105 fp=77 fn=11 tn=45
Precision	0.572192513368984	0.576923076923077
Recall	0.922413793103448	0.905172413793103

Purpose	Default Parameters	Trained Parameters c8,g0.25
Accuracy	53.2567% (139/261)	54.4061% (142/261)
Mean squared error	7.47893	7.29502
Squared correlation coefficient	0.00565787	0.00704068
Confusion matrix	tp=63 fp=47 fn=75 tn=76	tp=80 fp=61 fn=58 tn=62
Precision	0.572727272727273	0.567375886524823
Recall	0.456521739130435	0.579710144927536

Reason	Default Parameters	Trained Params c32,g0.0625
Accuracy	53.4188% (125/234)	54.2735% (127/234)
Mean squared error	11.6453	11.4316
Squared correlation coefficient	0.0108362	0.0141019
Confusion matrix	tp=41 fp=26 fn=83 tn=84	tp=44 fp=27 fn=80 tn=83
Precision	0.611940298507463	0.619718309859155
Recall	0.330645161290323	0.354838709677419

Result	Default Parameters	Trained Parameters c1,g1
Accuracy	69.4175% (143/206)	69.9029% (144/206)
Mean squared error	11.0097	10.835
Squared correlation coefficient	0.00210027	0.00760744
Confusion matrix	tp=0 fp=1 fn=62 tn=143	tp=4 fp=4 fn=58 tn=140
Precision	0	0.5
Recall	0	0.0645161290322581

BIBLIOGRAPHY

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

Charniak, E. (1997). Statistical techniques for natural language parsing. *AI Magazine 18*(4), 33–44.

Creswell, C. (2004). The predicate-argument structure of discourse connectives: A corpus-based study. In *Anaphora Processing: Linguisticc, cognitive, and computational modelling (To Appear).*

Forbes, K. (2001). D-LTAG system - discourse parsing with a lexicalized tree adjoining grammar. In *ESSLLI 2001 Workshop on Information Structure, Discourse Structure and Discourse Semantics*, Helsinki, Finland.

Gildea, D. and D. Jurafsky (2002). Automatic labeling of semantic roles. *Computational Linguistics 28*(3), 245–299.

Girju, R. (2001). Answer fusion with on-line ontology development. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL) Student Research Workshop*, Pittsburgh, PA.

Girju, R. and D. Moldovan (2004). Semantic verb clustering for automatic semantic role classification.

Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In C. Nédellec and C. Rouveirol (Eds.), *Proceedings of ECML-98, 10th European Conference on Machine Learning*, Number 1398, Chemnitz, DE, pp. 137–142. Springer Verlag, Heidelberg, DE.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In I. Bratko and S. Dzeroski (Eds.), *Proceedings of ICML-99, 16th International Conference on Machine Learning*, Bled, SL, pp. 200–209. Morgan Kaufmann Publishers, San Francisco, US.

Jurafsky, D. and J. H. Martin (2000). *Speech and Language Processing.* Prentice-Hall, Inc.

Knott, A. and T. Sanders (1998). The classification of coherence relations and their linguistic markers: An exploration of two languages. *Journal of Pragmatics 30*(2), 135–175.

Lapata, M. and A. Lascarides (2004). Inferring sentence-internal temporal relations. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, Edmonton, Canada.

Lodhi, H. (2002). Text classification using string kernels. *Journal of Machine Learning Research 2*.

Mann, W. C. (1984, June). Discourse structures for text generation. In *Proceedings of the 22nd Annual Meeting of the Association for Computational Linguistics.*

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

Manning, C. and H. Schutze (1999). *Foundations of Statistical Natural Language Processing.* The MIT Press.

Marcu, D. and A. Echihabi (2002). An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA.

Miltsakaki, E. (2002). Annotating discourse connectives and their arguments. *Proceedings of the NAACL/HLT Workshop Frontiers in Corpus Annotation.*

Mitchell, T. (1997). *Machine Learning.* McGraw-Hill.

Moser, M. and J. Moore (1995). Using discourse analysis and automatic text generation to study discourse cue usage. In *AAAI 1995 Spring Symposium Series: Empirical Methods in Discourse Interpretation and Generation, pages 92- 98.*, Stanford, CA, pp. 92–98.

Pradhan, S. (2003). Support vector learning for semantic argument classification. *Technical Report, TR-CSLR-2003-03, Center for Spoken Language Research.*

Quirk, R. (1985). *A Comprehensive Grammar of the English Language.* Longman.

Resnik, P. (1997). Selectional preference and sense disambiguation. In *Proceedings of ACL Siglex Workshop on Tagging Text with Lexical Semantics, Why, What and How?, Washington, April 4-5, 1997.*

RST-DT (2002). RST *Discourse Treebank.* Linguistic Data Consortium, http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T07.

Sameer Pradhan, Kadri Hacioglu, W. W. J. M. and D. Jurafsky (2003). Semantic role parsing: Adding semantic structure to unstructured text. In *Proceedings of the International Conference on Data Mining (ICDM 2003)*, Melbourne, Florida.

Soricut, R. and D. Marcu (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, Edmonton, Canada.

Tong, S. and D. Koller (2000). Support vector machine active learning with applications to text classification. In P. Langley (Ed.), *Proceedings of ICML-00, 17th International Conference on Machine Learning*, Stanford, US, pp. 999–1006. Morgan Kaufmann Publishers, San Francisco, US.

Vapnik, V. (1982). *Estimation of Dependencies Based on Empirical Data.* Springer-Verlag.

Vapnik, V. (2000). *The Nature of Statistical Learning Theory.* Springer-Verlag.