Modelling Entity Instantiations

by

Andrew James McKinlay

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy.

UNIVERSITY OF LEEDS
The University of Leeds
School of Computing

February 2013

Declarations

The candidate confirms that the work submitted is his own, except where work which has formed part of jointly-authored publications has been included. The contribution of the candidate and the other authors to this work has been explicitly indicated below. The candidate confirms that appropriate credit has been given within the thesis where reference has been made to the work of others.

- A. McKinlay and K. Markert. Modelling entity instantiations. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 268–274. RANLP 2011 Organising Committee, 2011

  Author contributions: Andrew McKinlay was principal author, performed all experiments and most of the analysis of results. Katja Markert provided general guidance, feedback and supervision as well as aiding with analysis of results.

  Chapters based on this work: The parts of Chapters 3 and 4 which relate to intersentential entity instantiations.

- A. McKinlay and K. Markert. Recognising sets and their elements: Tree kernels for entity instantiation identification. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, 2013

  Author contributions: Andrew McKinlay was principal author, performed all experiments and most of the analysis of results. Katja Markert provided general guidance, feedback and supervision as well as aiding with analysis of results.

  Chapters based on this work: The parts of Chapters 3 and 4 which relate to intrasentential entity instantiations.

Abstract

The problem of automatically extracting structured information from texts is an important, unsolved problem within the field of Natural Language Processing. The extraction of such information can facilitate activities such as the building of knowledge bases, automatic summarisation and sentiment analysis. A human reader can easily discern the events described in a text, along with the participants and the relationships between them, but using a computer to automatically discover the same information is much more challenging.

Particular focus has been given to extracting relations between the entities in a text, such as those representing geographical locations, personal and social relationships, and employment. In this thesis, we consider two closely related entity relationships, which are interesting, frequent and have not been tackled previously, which we refer to collectively as *entity instantiations*.

We define an entity instantiation as an entity relation in which a set of entities is introduced, and either a member or subset of this set is mentioned. In the example below, we see a set membership instantiation, between *'several EU countries'* and *'the UK'*, along with a subset instantiation, between the same set and *'the low countries'*.

> Inflation has increased sharply in **several EU countries**.
> In *the UK*, this has accompanied a drop in interest rates, but in *the low countries* rates have remained steady.

This thesis details the creation of the first corpus of entity instantiations. The final corpus consists of 4,521 instantiations, 2,118 of which are intersentential, and 2,403 of which are intrasentential, annotated over 75 Penn Treebank Wall Street Journal newswire texts. The subsequent annotation study shows high levels of inter-annotator agreement and our corpus study analyses the annotated entity instantiations in terms of their internal structure, the distance between arguments and their syntactic relationship, finding a particularly strong link between syntactic parent-child relationships and sentence-internal entity instantiations.

To establish that the accurate automatic identification of entity instantiations is possible, we develop the first instantiation identification algorithm, which uses a supervised machine learning approach. The feature set draws on surface, syntactic, contextual, salience and knowledge features to aid classification. We separately apply our classifier to intersentential and intrasentential entity instantiations and experiment with both balanced data, with a 50/50 positive/negative split, and the original unbalanced corpus. The classifier records highly significant performance increases over both unigram-based

and majority class baselines on the balanced data, and also on the original distribution of intrasentential instantiations.

In order to take advantage of the aforementioned link between syntax and intrasentential entity instantiations, tree kernels were employed to learn directly from the syntactic parse trees which contain the two potential participants in an intrasentential instantiation. The tree kernel features perform similarly to the unstructured feature set, with a much shorter development time. Combining tree kernels with unstructured features gives further improvements over both the baselines, and either method in isolation.

We also apply our entity instantiations to the difficult problem of implicit discourse relation classification, hypothesising that introducing features identifying the presence of an entity instantiation between the arguments of a discourse relation can improve classification performance. Our experiments show that an entity instantiation is a strong indicator of the presence of an *Expansion.Instantiation* discourse relation. We create a binary Expansion.Instantiation classifier, based on the feature set detailed in Sporleder and Lascarides (2008), but augment it by adding entity instantiation features based on gold standard annotations. The classifier which includes entity instantiation data performs significantly better than the same classifier without entity instantiation data. We also experiment with the incorporation of *machine-identified* entity instantiations. However, our entity instantiation classifier is not sufficiently accurate to impact on discourse relation classification.

Acknowledgements

First of all, I'd like to thank my supervisor, Dr Katja Markert. Getting to this point was *hard*, and her support, insight and encouragement — in both carrot and stick form — were certainly necessary along the way. Her suggestions for ways round the many setbacks I encountered, and belief that all would work out right in the end, were invaluable in completing this thesis. There were plenty of times that I walked into her office for a meeting thoroughly fed up, and would come away with a plan, ideas and motivation.

Secondly, I'd like to thank my parents. Their love, support, generosity, not to mention their free board and lodgings, helped me immeasurably in completing of this thesis.

Thirdly, my huge, huge, huge thanks go to my girlfriend, Sarah. She has kept me fed, watered, motivated and sane over the last couple of years. I couldn't have done it without her love and support. Here's to being grown-ups soon!

I'd also like to thank my friends, even if most of your eyes did glaze over when I started talking about my research. Special thanks go to Chris Windsor, Phil Beedham, James Newsome, Mark Hartley and Ryan Ambrosen for providing welcome distractions and moral support over the years.

Finally, I'd like to take this opportunity to acknowledge anyone else who has helped me over the past four and a half years, including those people in the NLP group with whom I've discussed my work — particularly Amal Alsaif, Josiah Wang and Noushin Rezapour Asheghi. Thank you for you help.

Contents

1 Introduction **1**

1.1 Problem Outline 1
1.2 Importance of the Problem 3
1.3 Problem Definition 5
1.4 Research Questions and Hypotheses 10
1.4.1 Research questions 10
1.4.2 Hypotheses 11
1.5 Contributions 12
1.6 Thesis Overview 13

2 Literature Review **15**

2.1 Information Extraction 15
2.1.1 Early information extraction 17
2.1.2 Template filling 17
2.1.3 Relation extraction 24
2.1.3.1 Approaches using unstructured features 27
2.1.3.2 Kernel approaches 29
2.1.3.3 Comparison of research from Sections 2.1.3.1 and 2.1.3.2 32
2.1.3.4 Connection to entity instantiations 32
2.1.4 Unsupervised relation extraction 34
2.2 Context-independent Relation Extraction 37
2.2.1 Minimally supervised relation extraction 37
2.2.2 Minimally supervised set extraction 42
2.2.3 Differentiating entity instantiations from context-independent relation extraction 43
2.3 Bridging Anaphora 44
2.3.1 Anaphora and bridging 44
2.3.2 Linguistic bridging research 46

2.3.3 Bridging corpora . . . 47
2.3.4 Bridging anaphora resolution algorithms . . . 48
2.4 Conclusion . . . 50

3 Creating a Corpus of Entity Instantiations 51
3.1 Motivation . . . 51
3.2 Intuitive Definition and Examples . . . 53
3.2.1 Instantiation definition . . . 53
3.2.2 Variety in entity instantiations . . . 55
3.3 Exact Definition and Annotation Guidelines . . . 58
3.3.1 Annotation restrictions . . . 58
3.3.2 Definition of a mention . . . 60
3.3.2.1 Generic pronouns . . . 60
3.3.2.2 Idiomatic mentions . . . 60
3.3.2.3 Non-referential 'it' . . . 61
3.3.3 Specific annotation rules and special cases . . . 61
3.3.3.1 Generic mentions . . . 61
3.3.3.2 Indefinite pronouns . . . 62
3.3.3.3 Negated mentions . . . 62
3.3.3.4 Members implicitly excluded from sets . . . 62
3.3.3.5 Metonymic mentions . . . 63
3.3.3.6 Co-ordinations . . . 63
3.3.3.7 Nested mentions . . . 64
3.4 Potential Difficulties and Borderline Cases . . . 65
3.5 The Annotation Design and Process . . . 67
3.5.1 Choice of texts to annotate . . . 67
3.5.2 Pre-processing of noun phrases . . . 68
3.5.3 Classification of noun phrases into singular or plural . . . 69
3.5.4 Annotation tool . . . 72
3.5.4.1 Annotation tool requirements . . . 72
3.5.4.2 Annotation tool implementation . . . 73
3.6 Annotation Results . . . 74
3.6.1 Agreement study . . . 74
3.6.1.1 Intersentential agreement . . . 74
3.6.1.2 Intersentential disagreement analysis . . . 79
3.6.1.3 Intrasentential agreement study . . . 81

3.6.1.4 Intrasentential disagreement analysis . . . 81
3.7 Restriction of Annotations . . . 82
3.8 Gold Standard Corpus . . . 84
3.8.1 Corpus dimensions . . . 84
3.8.2 Entity instantiation distribution . . . 86
3.8.3 Intrasentential analysis: syntactic arrangement of noun phrases . . . 88
3.8.4 Intersentential analysis . . . 91
3.8.4.1 Ordering of and distance between noun phrases . . . 91
3.8.4.2 Noun phrase categorisation . . . 94
3.9 Conclusion . . . 97
3.9.1 Summary . . . 97
3.9.2 Future work . . . 98

4 Machine Learning of Entity Instantiations 101
4.1 Feature Design . . . 101
4.1.1 Surface features . . . 102
4.1.1.1 N-grams . . . 102
4.1.1.2 Part-of-speech . . . 102
4.1.1.3 Head words . . . 102
4.1.1.4 Lemmas . . . 103
4.1.1.5 Head word lemmas . . . 103
4.1.1.6 Levenshtein's distance . . . 103
4.1.1.7 Distance . . . 104
4.1.1.8 Ordering . . . 104
4.1.2 Salience features . . . 104
4.1.2.1 Grammatical role . . . 104
4.1.2.2 Mention count . . . 104
4.1.3 Syntactic features . . . 105
4.1.3.1 Syntactic parallelism . . . 105
4.1.3.2 Modification . . . 105
4.1.4 Context features . . . 105
4.1.4.1 Verb semantics . . . 105
4.1.4.2 Quotations . . . 106
4.1.4.3 Discourse relations . . . 106
4.1.5 Knowledge-based features . . . 107
4.1.5.1 WordNet . . . 108

4.1.5.2 Pattern-based hyponyms . . . 108
4.1.5.3 Animacy . . . 109
4.1.6 Full feature list . . . 109
4.2 Intrasentential Features — Tree Kernels . . . 112
4.2.1 Tree representation . . . 114
4.2.2 Tree kernel algorithms . . . 120
4.2.3 Application of tree kernels to intrasentential entity instantiations . 121
4.3 Experimental Setup . . . 126
4.4 Evaluation . . . 129
4.4.1 Evaluation measures . . . 129
4.4.2 Intrasentential evaluation . . . 132
4.4.2.1 Evaluation of flat feature performance on balanced data set . . . 132
4.4.2.2 Evaluation of tree kernel performance on balanced data set . . . 133
4.4.2.3 Evaluation of combination kernels on balanced data set 136
4.4.2.4 Evaluation of flat feature performance on unbalanced data set . . . 137
4.4.2.5 Evaluation of tree kernel performance on unbalanced data set . . . 138
4.4.2.6 Evaluation of combined kernel performance on unbalanced data set . . . 138
4.4.2.7 Intrasentential summary results . . . 141
4.4.2.8 Error analysis . . . 141
4.4.3 Intersentential evaluation . . . 146
4.4.3.1 Evaluation of performance on balanced data set . . . 146
4.4.3.2 Evaluation of performance on unbalanced data set . . . 147
4.4.3.3 Error analysis . . . 150
4.5 Conclusion . . . 155
4.6 Future Work . . . 156
4.6.1 Feature selection . . . 156
4.6.2 Additional features . . . 157
4.6.3 Kernel methods . . . 157
4.6.4 Machine learning techniques . . . 158

5 Entity Instantiations and Discourse Relations **161**
5.1 Introduction . . . 161
5.2 Background . . . 164
5.2.1 Theories of discourse . . . 164
5.2.2 Corpora . . . 166
5.2.3 Automatic approaches . . . 167
5.3 The Interaction between Discourse Relations and Entity Instantiations . . 172
5.3.1 Entity instantiation and discourse relation co-occurrence . . . 172
5.3.2 Annotating discourse relations for the presence of entity instantiations . . . 180
5.4 Baseline Discourse Relation Classification . . . 182
5.4.1 Feature set . . . 182
5.4.2 Full feature list . . . 184
5.4.3 Experimental set-up . . . 187
5.4.4 Results . . . 187
5.5 Discourse Relation Classification with Gold Standard Entity Instantiations 187
5.6 Discourse Relation Classification with Machine Identified Entity Instantiations . . . 189
5.6.1 Machine-identified entity instantiation features . . . 189
5.6.2 Results . . . 190
5.7 Conclusion . . . 191
5.7.1 Summary . . . 191
5.7.2 Future work . . . 192

6 Conclusion **193**
6.1 Summary . . . 193
6.2 Impact of Limitations . . . 195
6.3 Future Work . . . 198
6.3.1 Corpus extensions . . . 198
6.3.2 Machine learning improvements . . . 200
6.3.3 Applications . . . 204

Bibliography **206**

A Inter-sentential Annotation Scheme **233**

B Intra-sentential Annotation Scheme **241**

C Psuedocode of Singular/Plural NP Classifier **251**

D Hierarchy of relations in the PDTB **257**

List of Figures

2.1 An example paragraph which might be subject to slot filling. Taken from Jurafsky and Martin (2009), pp 759. 18
2.2 An example of a filled template. Taken from Jurafsky and Martin (2009), pp 759. 18
2.3 An augmented parse tree from Miller et al. (1998). 27
2.4 The full dependency tree for the sentence containing the relation Protesters AT stations, as presented to the learner by Culotta and Sorensen (2004) . . 29
2.5 The shortest path dependency tree for the sentence considered in 2.4, as presented to the learner by Bunescu and Mooney (2005) 30
2.6 A sentence parse tree with a SOCIAL.Other-Personal relation between *partners* and *workers*, and the path-enclosed tree for the relation. 31

3.1 A constituency parse tree representation of the set noun phrase from Example 3.6. 55
3.2 The completed annotation tool. 73
3.3 The annotation tool with numbered functions. 75
3.4 The annotation tool showing the Subset panel. 76
3.5 The Show Further Context function of the annotation tool. 77
3.6 The intrasentential annotation tool. 78

4.1 A vector of features, as presented to the ICSIBoost machine learner. . . . 112
4.2 An example of large sentence parse tree with a relatively local entity instantiation. 116
4.3 The five tree representations used in Zhang et al. (2006), based on the sentence "... provide benefits to 200 domestic partners of their own workers in New York". "Partners" and "workers" are the two entities in question. . 117
4.4 The SPT representation of the sentence from Figure 4.3. 118
4.5 The SPET representation derived from Example 4.13. 119
4.6 The SPT representation derived from Example 4.13. 119

4.7 Examples of the substructures extracted and compared by three tree kernel algorithms: the ST, SST and PT, from Moschitti (2006a). 122
4.8 Example constituency parse trees of simple conjunctive phrases. 123
4.9 Example constituency parse trees comprising noun phrases with prepositional phrases. 124
4.10 Example constituency parse trees of nested sets. 125
4.11 A tree similar to those in Figure 4.9, but that does not represent an entity instantiation. 126
4.12 An example of a distant entity instantiation in a syntactically complex sentence, which is unlikely to be identified by tree kernel learning. 127
4.13 ROC curve: Set members 151
4.14 ROC curve: Subsets 151

6.1 A graphical representation of the instantiations in Example 6.5. 203

List of Tables

2.1 Details of the tasks of the Message Understanding Conferences 19
2.2 Some example heuristics from Riloff (1993). 20
2.3 A summary of the highest scoring algorithms in MUCs 3–7, from Chinchor (1998) 21
2.4 The list of ACE Event Types, reproduced from Grishman (2012) 23
2.5 The full ACE-2004 relation schema, as described in Grishman (2012). . . 26
2.6 An overview of the RDR portions of each ACE evaluation. 26
2.7 Comparison of relation extraction results on the ACE-2003 corpus. 32
2.8 Comparison of relation extraction results on the ACE-2004 corpus. 33
2.9 Comparison of relation extraction results on corpora other than ACE-2003 and ACE-2004. 33
2.10 Example patterns from Hearst (1992) 38

3.1 Positive and negative examples of textual entailment from Dagan et al. (2006). 52
3.2 Instantiations present in Example 3.17. 59
3.3 The NPs present in Example 3.52, and whether they are included after pre-processing. 70
3.4 The NPs present in Example 3.53, and whether they are included after pre-processing. 70
3.5 Results of the intersentential agreement study 79
3.6 Results of the intersentential agreement study 81
3.7 Frequency of intrasentential, adjacent intersentential and non-adjacent intersentential entity instantiations in 3 sample texts 83
3.8 Total corpus size in words, sentences and texts 85
3.9 Distribution of text lengths in corpus by genre, measured in words. 86
3.10 Distribution of text lengths in corpus by genre, measured in sentences. . . 86
3.11 Frequency of entity instantiations in 75 texts 86

3.12 Distribution of entity instantiations in corpus per text, by genre. 87
3.13 Distribution of intrasentential entity instantiations in corpus per sentence, by genre. 89
3.14 Distribution of intersentential entity instantiations in corpus per sentence pair, by genre. 90
3.15 Frequency of intersentential entity instantiations and non-instantiation NP pairs in 75 texts 91
3.16 Frequency of intrasentential entity instantiations and non-instantiation NP pairs in 75 texts 91
3.17 Frequency of syntactic relationships between NPs in intrasentential set member entity instantiations. 92
3.18 Frequency of syntactic relationships between NPs in intrasentential subset entity instantiations. 92
3.19 Ordering of NPs in intersentential set member entity instantiations 92
3.20 Ordering of NPs in intersentential subset entity instantiations 93
3.21 Distribution of intersentential set member entity instantiations by normalised distance in words between noun phrases. 93
3.22 Distribution of intersentential subset entity instantiations by normalised distance in words between noun phrases. 93
3.23 Intersentential set member NP categorisation. 95
3.24 Intersentential subset NP categorisation. 95
3.25 Intersentential set member NP modification. 96
3.26 Intersentential subset NP modification. 96
3.27 Intersentential set member set NP types 96
3.28 Intersentential subset set NP types 97

4.2 Statistical analysis of the numerical features included in the full feature set. 113
4.3 Results from Zhang et al. (2006). 117
4.4 The size and distribution of intrasentential machine learning data sets. . . 128
4.5 The size and distribution of intersentential machine learning data sets. . . 128
4.6 Confusion matrix 129
4.7 Classification with changing thresholds 131
4.8 Results of the intrasentential classifier on the balanced data set — flat features 134
4.9 Results of the intrasentential classifier on the balanced data set — tree kernels and combined kernels 135

4.10 Results of the intrasentential classifier on the original data set — flat features 139
4.11 Results of the intrasentential classifier on the original data set — tree and combination kernels . . . 140
4.12 Intrasentential Summary Results. . . . 142
4.13 Breakdown of best performing intrasentential set member algorithm by syntactic relationship. . . . 143
4.14 Breakdown of best performing intrasentential subset algorithm by syntactic relationship. . . . 143
4.15 Results of the intersentential classifier on the balanced data set . . . 148
4.16 Results of the intersentential classifier on the original data set . . . 149
4.17 Breakdown of best performing intersentential set member algorithm by set member NP type. . . . 150
4.18 Breakdown of best performing intersentential subset algorithm by subset NP type. . . . 152

5.1 Distribution of discourse relations in 75 entity instantiation texts. . . . 173
5.2 Entity instantiation and discourse relation co-occurrence: all instantiations, all relations. . . . 175
5.3 Entity instantiation and discourse relation co-occurrence: all relations, inter vs intra breakdown. . . . 176
5.4 Entity instantiation and discourse relation co-occurrence: all relations, set member vs subset breakdown. . . . 176
5.5 Entity instantiation and discourse relation co-occurrence: all relations, implicit vs explicit breakdown. . . . 177
5.6 Entity instantiation and discourse relation co-occurrence: implicit relations, inter vs intra breakdown. . . . 178
5.7 Entity instantiation and discourse relation co-occurrence: implicit relations, set member vs subset breakdown. . . . 178
5.8 Entity instantiation and discourse relation co-occurrence: implicit relations, relation type breakdown. . . . 179
5.9 Entity instantiation and discourse relation co-occurrence: Expansion.Instantiation . . . 180
5.10 Entity instantiation and discourse relation co-occurrence: Expansion.Instantiation, inter vs intra breakdown. . . . 181
5.11 Entity instantiation and discourse relation co-occurrence: Expansion.Instantiation, set member vs subset breakdown. . . . 181

5.12 Distribution of entity instantiations over discourse relations 181
5.13 Presence of entity instantiations in discourse relations 182
5.15 Baseline classification of Expansion.Instantiation relations. 187
5.16 Classification of Expansion.Instantiation relations with gold standard entity instantiation features. 188
5.17 Classification of Expansion.Instantiation relations with machine-identified entity instantiation features. 191

List of abbreviations

Abbreviation	Definition	Explanation
ACE	Automatic Content Extraction	A set of shared IE tasks, including RE, Coreference Resolution and Event Detection.
API	Application Programming Interface	A way of accessing a resource programmatically.
AUC	Area Under Curve	The area under an ROC curve.
GPE	Geo-Political Entity	A type of named entity, which represents nations, regions, or governments.
IE	Information Extraction	The task of extracting structured information from free text.
MUC	Message Understanding Conference	One of a series of IE conferences which ran from 1987–1997
NE	Named Entity	Text representing the name of a person or organisation, or a location.
NP	Noun Phrase	A phrase with a noun (or pronoun) as the head.
PDTB	Penn Discourse Treebank	Corpus of WSJ articles annotated with discourse relations.
PMI	Pointwise Mutual Information	A measure of the similarity of two concepts.
POS	Part-of-speech	The lexical category of a word.
PTB	Penn Treebank	Corpus of WSJ articles annotated with POS and syntax trees.
RE	Relation Extraction	The IE sub-problem of identifying binary relationships between entities, such as *Part-Of, Employed-By* or *Located-In*.
ROC	Reciever Operating Characteristic	A method of displaying the performance of a classifier as its decision boundary is changed. It is a plot of the True Positive Rate against the False Positive Rate.
SVM	Support Vector Machine	A type of machine learning classifier.
WSJ	Wall Street Journal	American financial newspaper.

Chapter 1

Introduction

1.1 Problem Outline

This thesis is concerned with *entity instantiations*. An entity instantiation is an entity relationship in a text, where a *set* of entities is mentioned, and then a *member* or *subset*[1] of this set is introduced. Example 1.1 shows a pair of sentences with the set in bold and set member in italics.[2] Examples 1.2 and 1.3 show a pair of sentences with a set in bold and subset in italics[3].

(1.1) a. **Some European funds** recently have skyrocketed.

b. *Spain Fund* has surged to a startling 120% premium.

(1.2) a. **Bids totalling $515 million** were submitted.

b. *Accepted offers* ranged from 8.38% to 8.395%

(1.3) a. In the aftermath of the downturn **many manufacturers** have struggled.

b. *Those relying on foreign imports* have had the most difficulty.

[1] When we refer to a subset, we mean a *proper* subset. We consider two equal sets to be coreferent, and not participating in an Entity Instantiation.

[2] Examples 1.1, 1.2, 1.3, 1.5 and 1.12 are from the Penn Treebank Wall Street Journal Corpus (Marcus et al., 1993).

[3] This convention of displaying the set in bold and the member or subset in italics is used throughout the thesis.

(1.4) a. **Footballers** are vastly overpaid.

b. Manchester United pay *Wayne Rooney* £200,000 per week.

The recognition of entity instantiations can often be difficult. Entity instantiations occur in a variety of forms. Participating noun phrases (NPs) include common nouns, pronouns and proper nouns and can also have missing head nouns (see Example 1.3). The participating NPs can also fulfil various grammatical roles in a sentence. Examples 1.1, 1.2 and 1.6 show entity instantiations where both the set and member are subjects of their respective sentences, in contrast to Example 1.4 (between a subject and direct object), and Example 1.5 (between nested NPs in a more complex sentence).

The two participants in an entity instantiation can have word overlap (see Example 1.1) or synonymous head nouns (see Example 1.2), but are often not related in such a simple manner. For instance, in Example 1.4, knowledge that Wayne Rooney is a footballer is helpful in identifying the entity instantiation. Additionally, accurate recognition of an entity instantiation often needs contextual knowledge. In Examples 1.6 and 1.7, the contextual information about the attitudes of the workers is necessary to establish whether an entity instantiation exists.

(1.5) a. Already, scientists are developing tests based on **the newly identified genes** that, for the first time, can predict whether an otherwise healthy individual is likely to get cancer.

b. "It's a super-exciting set of discoveries," says Bert Vogelstein, a Johns Hopkins University researcher who has just found *a gene pivotal to the triggering of colon cancer*.

(1.6) a. **Some workers** are opposed to strike action.

b. *John Smith* fears that a strike could damage the industry's public perception.

(1.7) a. **Some workers** are opposed to strike action.

b. *David Jones*, however, is willing to put his job on the line for the cause. *(Not an instantiation.)*

The problem of identifying entity instantiations is untackled. The problem is related to Relation Extraction (RE), which is the discovery of semantic relations between pairs of entities. Much of the work in this field is connected to the Message Understanding Conferences (MUC, 1987-1998) and the NIST Automatic Content Extraction programs (ACE, 2000-2005), both of which provide annotated corpora of semantic relations. Entity

instantiations are not considered in the MUC and ACE annotation schemes, or any other semantic relation annotation scheme. MUC and ACE consider relationships between different types of entity, such as those between persons and locations or organisations and persons rather than the sets and instances of entities of the same type that we consider. The differences and similarities between entity instantiations and relation extraction, as well as the wider problem of Information Extraction, are discussed in Section 2.1.

The dependence on the surrounding context to establish the presence of an instantiation makes the problem distinct from that of harvesting hyponyms or other relations from large corpora or the World Wide Web (Hearst, 1992; Pantel and Pennacchiotti, 2006; Mintz et al., 2009). The relationships discovered in this way are largely *context-independent*. For example, relationships such as *'a dog is a type of animal'* or *'Microsoft is a company'* are truths that are unaffected by textual context. We further discuss this work and its links with entity instantiations in Section 2.2.

The problem is also related to, but distinct from, bridging. Bridging is the problem of establishing non-coreferent anaphoric entity relationships such as meronymy, where the relationship is necessary for the interpretation of the anaphor (Clark, 1975; Prince, 1981). For instance, in example 1.3 the subset *'Those relying on foreign imports'* requires knowledge of the set *'manufacturers'* to be understood. A large proportion of entities that participate in entity instantiations do not rely on an antecedent to be interpreted.

In this thesis, we explore the problem of entity instantiations, firstly from the perspective of human annotation and subsequently from the perspective of automatic identification. We present an annotated corpus of entity instantiations, containing 4521 instantiations annotated both intrasententially and between adjacent sentences, over 75 texts. We then use this corpus to train an automatic entity instantiation identifiers for inter- and intrasentential instantiations. We finally utilise our automatic instantiation identifier as part of a system for classifying *Expansion.Instantiation* discourse relations in the Penn Discourse Treebank (PDTB) (Prasad et al., 2008).

1.2 Importance of the Problem

Entity instantiations are important. Generally, knowledge of the relationship between two entities or sets of entities can help to supplement our knowledge about either participant. In Example 1.8, the Entity Instantiation between *'several EU countries'* and *'the UK'* gives us the knowledge that not only are interest rates dropping in the UK, but inflation is rising as well.

(1.8) a. Inflation has increased sharply in **several EU countries**.

b. In *the UK*, this has accompanied a drop in interest rates.

In addition to this property, entity instantiations have the potential to be important for a number of applications, which are detailed below.

Knowledge extraction. Knowledge extraction is the process of automatically extracting structured knowledge from both unstructured and structured data. The goal is to create some sort of schema to represent the data, such as an *ontology*. An ontology is a hierarchical organisation of concepts which captures subset and superset relationships between the concepts (Jurafsky and Martin, 2009).

In this thesis we develop techniques for identifying set membership and subset relations in text. Although our definition of entity instantiations includes complex noun phrases and context-dependent relationships rather than the more concrete facts represented in a typical ontology, the identification of entity instantiations could serve as an important pre-processing step in ontology construction.

Discourse relations. In text, clauses and sentences do not exist as isolated units but instead are connected by relations. These *discourse relations* represent relationships such as cause, contrast, restatement and condition.

Discourse relations are often signalled by a *connective*, a word or phrase which makes clear the relationship. For example, the connective *but* usually signals a contrast, *because* usually signals a cause and *if* signals a condition. Often, relations are not signalled by a connective, but are instead understood *implicitly*. Example 1.9 shows an example of an implicit causal discourse relation.

(1.9) a. John had no room for desert.

b. He'd eaten far too much already.

Whilst, due to the unambiguous nature of many connectives, explicit discourse relation classification can be performed with high accuracy (Pitler et al., 2008), the classification of implicit discourse relations remains a challenging task (Pitler et al., 2009; Lin et al., 2009; Zhou et al., 2010; Louis et al., 2010b; Park and Cardie, 2012; Wang et al., 2010). An understanding of the entity relationships within and between the arguments of an implicit discourse relation, including entity instantiations, could improve classification accuracy. In particular, the discourse relation *Expansion.Instantiation* from the PDTB scheme of discourse relations has a strong connection with entity instantiations,

which we explore in detail in Chapter 5. Example 1.10 shows an example of an Expansion.Instantiation discourse relation which co-occurs with an entity instantiation.

(1.10) a. Attempts to produce **"pan-European" TV programs** have generally resulted in disappointment.

b. *The Eurovision Song Contest, one such program*, has been described as the world's most boring TV show.

Sentiment analysis. Sentiment analysis is the sub-field of Natural Language Processing (NLP) concerned with automatically detecting and classifying sentiment and opinions in texts. Entity instantiations could aid the interpretation of sentiment. For instance, in Example 1.4 the second sentence considered in isolation does not carry any sentiment. However, the author's thoughts about the pay of Wayne Rooney can be inferred from the negative sentiment of the first sentence and the entity instantiation between the two sentences.

Summarisation. Entity instantiations could be used to improve techniques for the automatic summarisation of texts. We suggest that, in general, sentences which contain a number of sets may be more useful in a summary than those which contain mostly set members and subsets. Example 1.11 shows a set-dense first sentence which would be useful in a summary, and a member/subset dense second sentence which may be excluded from a summary.

(1.11) a. Several other Japanese companies and regional governments have sent aid to San Francisco.

b. Sumitomo Bank donated $500,000, Tokyo prefecture $15,000 and the city of Osaka $10,000.

1.3 Problem Definition

In this Section, we discuss the connection between entity instantiations and other related natural language processing problems, and make clear the limitations of our study.

Focus on set membership and subsets. There are a number of set relationships that could have been considered as part of this thesis, including set identity, set-complement relationships, co-set-membership or set disjointness. Examples 1.12, 1.13 , 1.14 and 1.15 show these relationships respectively, with participating phrases underlined.

(1.12) Program traders are fond of predicting that if they are blocked in the U.S., they will simply emigrate to foreign stock markets.

(1.13) The ordinance, in Moon Township, prohibits locating a group home for the handicapped within a mile of another such facility.[4]

(1.14) a. Bradford City's players have excelled this season.

b. The performances turned in by Gary Jones and James Hanson means they are attracting the attention of bigger clubs.

(1.15) a. English defenders have a tendency to hit the ball high and long towards the striker.

b. In contrast, Spanish defenders prefer to pass short to a deep-lying midfielder.

Set identity and set complement relationships have already been researched as part of the study of coreference (ACE, 2000-2005; Weischedel et al., 2011) and as part of work relating to *comparative anaphora* (Modjeska, 2000, 2004; Markert and Nissim, 2005), respectively. We therefore chose two relationships that had not been previously examined in depth — set membership and subsethood.

A further reason for selecting set membership and subsethood relationships was the challenge in identifying them. The variety in the internal structure of the participant NPs, the variety in their distribution and their overlap with other phenomena, such as metonymy and pronominal anaphora, means that no simple rules can adequately identify entity instantiations, and instead a machine learning approach is required. The complexity of these relationships means that the consideration of additional relationships is beyond the scope of this PhD.

We also consider entity instantiations important for a number of applications, including those listed in Section 1.2. Set membership and subset relationships are particularly important for the identification of the discourse relation Expansion.Instantiation, which we tackle in Chapter 5.

Anaphoric and non-anaphoric entity instantiations. An *anaphor* is a reference to an entity previously introduced in the discourse, known as the *antecedent* (Mitkov, 1999). Often, anaphora are pronouns. Example 1.16 shows a pair of sentences with two anaphoric references; *He* is an anaphoric reference to the antecedent *John*, and *it* refers back to *an apple*.

[4]This example, from the Penn Treebank, was identified in Markert and Nissim (2005).

(1.16) a. John ate an apple.

b. He thought it was very tasty.

Pronominal anaphora resolution — the problem of establishing which antecedent a pronoun refers to — is a well studied problem (Hobbs, 1978; Brennan et al., 1987; Lappin and Leass, 1994; Soon et al., 2001; Yang et al., 2003, *inter alia*). However, other sorts of anaphora, such as those that do not co-refer with their antecedent but can instead be inferred from it, are less well researched. These sort of anaphora are referred to as *bridging* anaphora, because often the reader is required to 'bridge' the gap between the anaphor and the antecedent based on some inference or prior knowledge. Example 1.17 shows a sentence, taken from Jurafsky and Martin (2009), which has two bridging anaphora; *a door* and *the engine*, which the reader infers are parts of the previously mentioned *1961 Ford Falcon*.

(1.17) I almost bought **a 1961 Ford Falcon** today, but *a door* had a dent and *the engine* seemed noisy.

Bridging anaphora can also be connected to their antecedent by set membership, as in this example from Clark (1975):

(1.18) I met **two people** yesterday. *The woman* told me a story.

Anaphoric set members and subsets have been covered partially in the study of bridging anaphora. At least three corpora have annotated them, but each corpora's treatment has shortcomings. Firstly, the GNOME corpus (Poesio, 2003) includes categories for set membership and subset bridging, but agreement on the annotation of bridging relations is poor (22%), and only 581 examples are annotated. Secondly, Nissim et al. (2004)'s Switchboard corpus includes a set membership category, which consists of set membership, subsets and co-set-members (two members of a common set). However, the annotation of set membership has a number of restrictions, including that it can only be used if anaphor and antecedent have either the same head, a synonymous head or are part of a hyponym relation encoded in WordNet. Additionally, the antecedent of each anaphor is not marked. Finally, Markert et al. (2012)'s corpus does not include a separate set membership or subset category, but rather has one category for all bridging anaphora. Their corpus is also relatively small, containing a total of 663 bridging references, of which only a small subset are bridged via set membership or subset relationships.

To our knowledge, no prior study has considered non-anaphoric entity instantiations. Relation extraction annotation schemes, such as those that formed part of ACE and MUC,

focus on relations that describe personal and social relationships, physical locations, employment and affiliation, rather than considering set membership or subsets. We fully discuss the distinctions between this work and our own in Section 2.1.3.

Our study encompasses both anaphoric set membership and subset relationships, *and* cases where the set member or subset does not require the set to be interpreted. Our motivation is the discovery of set members and subsets in general. The anaphoricity of the relationship may be important in some applications, such as generating prosodic markings (Baumann and Riester, 2011). However, we expect that for several other future applications, such as discourse relation classification, knowledge extraction and sentiment analysis, the realisation of the relationship is unlikely to be of consequence. We therefore do not consider the anaphoricity of the relationship as part of either our annotation study or machine learning experiments.

Our work does not directly contribute to the anaphoric interpretation literature, because we do not focus specifically on anaphoric relationships. Making the distinction between anaphoric and non-anaphoric entity instantiations would be a useful extension to the work of this thesis, and this could be achieved, at least in part, by applying a variety of heuristics. For example, all NPs with proper noun heads are non-anaphoric, as are intrasentential instantiations where the set member or subset is nested within the set. A further, fuller discussion of the connection between entity instantiations and bridging anaphora can be found in Section 2.3.

Our decision to conflate anaphoric and non-anaphoric relationships of the same type is methodologically similar to the study of *coreference*. In coreference, pronominal identity anaphora as well as non-anaphoric expressions which refer to the same entity are tackled together, as in our study. This approach is not without its drawbacks — the relationship being studied might not be considered a linguistic phenomenon in its own right in the way that anaphora are, but because both convey the same relationship — identity — and a knowledge of this relationship is useful for a range of applications, the conflation of the anaphoric and non-anaphoric relationships is sensible.

Distance restriction. In this thesis, we limit annotation of entity instantiations to those occurring within a sentence or between adjacent sentences. Annotating entity instantiations without distance restrictions is very difficult — one must compare a potential set member or subset to every potential set in the document, and designing an annotation process that allows for reliable and replicable annotation in this scenario would be chal-

lenging[5]. It is worth noting, however, that coreference annotation schemes also consider distant relationships and require the comparison of many mentions in a lengthy text.

Restricting the annotation to anaphoric cases may have made a distance restriction unnecessary, because bridging anaphora are a relatively local phenomenon — Hou et al. (2013) report that 71% of the antecedents in their corpus are within two prior sentences of the anaphor. However, we were motivated to include non-anaphoric entity instantiations because they too convey important relationships that are useful for a range of applications.

Our decision to restrict annotation to adjacent sentences, as opposed to within a single paragraph or within the two preceding sentences, is partially motivated by the fact that implicit discourse relations in the PDTB are also only annotated between adjacent sentences. Of course, implicit discourse relations can *exist* between non-adjacent sentences, regardless of the fact that *annotation* is restricted to between adjacent sentences in the PDTB. Currently, the PDTB, complete with this restriction, is the largest annotated corpus of discourse relations available, and so our annotation restriction allows us to consider the implicit relations in this important resource. Other discourse relation frameworks, such as Rhetorical Structure Theory (RST) and the framework used by the Discourse Graphbank (Wolf and Gibson, 2005), do not limit their annotation in this way.[6]

We demonstrate in Chapter 5 that entity instantiations are useful in the discovery of one implicit discourse relation, and the decision to annotate between adjacent sentence should allow further exploration of the connection between discourse relations in the PDTB and entity instantiations in future.

Limitations on participating noun phrases. There are four possible NPs involved in entity instantiations: the set in a set member entity instantiation, the member in a set member entity instantiation, the superset in a subset entity instantiation and the subset in a subset entity instantiation. Other than ensuring that the member in a set member entity instantiation is singular, and that the other three possible NPs are plural, we place very few restrictions on the NPs involved. Unlike relation extraction schemes, we do not limit the participants to entities of fixed types, and we do not place any limitation on the form of the NPs — pronouns, proper names, demonstratives, quantified NPs, definite descriptions and so on are all annotated as long as they are participating in a set member or subset

[5]See Section 3.7 for a the results of a short annotation annotation study in which the distance restrictions are removed.

[6]Whilst RST and the Discourse Graphbank do not limit their annotations to between adjacent sentences, the building blocks of their annotation are small discourse segments (usually clauses) which cannot overlap and cover the whole text. This definition of discourse segments means that towards the bottom of the discourse tree/graph there are many sentence-internal relations, which entity instantiations as annotated in this thesis could help disambiguate.

relation.

We limit sets to plural NPs and members to singular NPs using the algorithm outlined in Section 3.5.3. We enforce this restriction to avoid marking relations other than entity instantiations, such as part-of, employment or location, and also to avoid situations where the set is in some way more than a simple grouping of entities, such as *'the EU'* or *'Manchester United'*.

In order to streamline the annotation process, we automatically remove NPs which, by their nature, can never be in an instantiation, such as adverbial NPs, NPs that are children of adverbial phrases and existential 'there' phrases. We also remove NPs that are not the largest NP that describes a certain concept, such as appositions, conjunctions of appositions, and NPs that are the head of a larger NP modified by prepositional phrases. This process is described in further detail in Section 3.5.2.

As part of the annotation process, annotators are asked to mark generic uses of *'we'* and *'you'*, references to the reader, non-referential uses of *'it'* and idiomatic NPs with no literal meaning as non-mentions. They are also required to refrain from marking negated mentions, indefinite pronouns such as *'either, each'* and *'any'*, and singular generic NPs as instantiations. The motivation for these exclusions to avoid situations, such as those in Examples 1.19 and 1.20 below, where the set membership or subset relation is unclear.

(1.19) a. **John Smith and John Doe** are competing for the contract.

b. *Either* could clinch it.

(1.20) a. *John*, Mary and James just sat and watched.

b. **Not one of them** dared intervene.

The limitations on participating noun phrases are discussed further in Section 3.3.1.

1.4 Research Questions and Hypotheses

1.4.1 Research questions

In this thesis, we aim to explore a variety of research questions, grouped into three main categories: human identification of entity instantiations, machine identification of entity instantiations and applications of entity instantiations.

Human identification and corpus study. Firstly, we wish to establish more clearly the nature of the phenomenon. How often do they occur in texts? Can they be identified

reliably by humans? Are there common lexical or syntactic patterns that indicate them? Are there patterns in the internal structure of a noun phrase which make it more likely to be part of an instantiation?

Machine identification. Secondly, assuming that humans can reliably identify instantiations, we wish to explore the possibility of using computational methods for automatic identification of instantiations. Can a computer identify instantiations with a reasonable degree of accuracy? Can a supervised machine learning approach be used to identify entity instantiations?

If we employ a machine learning approach, what sort of features will best represent potential instantiations to the learner? Is there a way of incorporating world knowledge? Might we learn directly from structured data such as syntactic parse trees and dependency structures? Do different types of instantiation (i.e. set membership, subset, intrasentential, intersentential) behave similarly enough to be considered a single task, or is it best to treat each individually?

Applications. Finally, we wish to consider the connections between entity instantiations and other natural-language phenomena. How do entity instantiations interact with discourse-level phenomena, such as discourse relations? Might knowledge of entity instantiations aid the classification of discourse relations?

1.4.2 Hypotheses

We make a number of specific hypotheses related to the research questions detailed in Section 1.4.1. These hypotheses are described below.

Our most significant hypothesis is that, in this thesis, we introduce a novel, untackled research problem — entity instantiations. We supplement this with the hypothesis that our formulation of entity instantiations makes it a well defined problem, and that one may develop an annotation schema and reliably annotate the phenomenon in text.

Secondly, we hypothesise that a machine learning approach can be used to automatically identify entity instantiations from texts. Specifically, we hypothesise that a supervised machine learning method can classify potential instantiations as positive or negative examples of the phenomenon. Based upon this hypothesis, we have several sub-hypotheses related to the machine interpretation of instantiations:

1. The surface form of the words involved is important for instantiation identification.

2. The salience of the two potential participants in an instantiation in the text is an indicator of the presence of an instantiation.

3. Features which use *world knowledge* to discover established links between noun phrases are useful for entity instantiation classification.

4. Knowledge of the syntactic relationship between the two participants in an entity instantiation aids classification. We hypothesise that this knowledge is especially relevant for intrasentential instantiations, where the two participants are part of the same syntactic parse tree.

Finally, we hypothesise that there is a strong link between entity instantiations and discourse relations, particularly the *Expansion.Instantiation* discourse relation. We hypothesise that knowledge of entity instantiations can improve the classification of *Expansion.Instantiation* discourse relations.

1.5 Contributions

The main contributions presented in this thesis are summarised below.

The creation of the first, reliably annotated, corpus of entity instantiations. We annotate examples of instantiations between adjacent sentences and within single sentences, over 75 Penn Treebank Wall Street Journal texts. This leads to the identification of 2118 intersentential instantiations, composed of 1477 set membership instantiations and 641 subset instantiations, and 2403 intrasentential instantiation composed of 1538 set membership instantiations and 865 subset instantiations.

We measure the agreement of two annotators over 5 randomly selected intersentential texts, and further 5 randomly selected intrasentential texts. We achieve agreement of $\kappa = 0.65$ and $\kappa = 0.75$ for inter- and intrasentential instantiations respectively. We also perform a corpus study, analysing the annotated entity instantiations in terms of their internal structure, the distance between arguments and their syntactic relationship.

The first automatic instantiation identifier. We create the first instantiation identification algorithm, using a supervised machine learning approach. Our feature set draws on surface, syntactic, contextual, salience and knowledge features to aid classification. We separately apply our classifier to intersentential and intrasentential entity instantiations and experiment with both balanced data, with a 50/50 positive/negative split, and

the original unbalanced corpus. We record highly significant performance increases over both unigram-based and majority baselines on the balanced data, and also on the original distribution of intrasentential instantiations.

The application of tree kernels to the problem of intrasentential instantiation identification. We use tree kernels to learn directly from the syntactic parse trees which contain the two potential participants in an intrasentential instantiation. We employ two tree kernels — the Shortest Path Tree (SPT) and the Shortest Path Enclosed Tree (SPET). These tree kernel features perform similarly to the flat feature set, with a much shorter development time. Combining tree kernels with flat features gives further improvements over both a unigram-based and a majority baseline and either method in isolation.

The first correlation study demonstrating a clear link between the presence of an entity instantiation and the *Expansion.Instantiation* discourse relation. We annotate 491 *Expansion.Instantiation* discourse relations and 509 randomly selected other discourse relations for the presence of entity instantiations. The 491 Expansion.Instantiation relations contained 642 entity instantiations, the other 500 relations contained 233 — a highly significant difference in proportion.

The first discourse relation classifier to incorporate entity instantiation data. We create a binary Expansion.Instantiation discourse relation classifier, based on the feature set detailed in Sporleder and Lascarides (2008), but augment it by adding features which indicate the presence of entity instantiations between the two arguments of the discourse relation, based on gold standard annotations. The classifier which includes entity instantiation data performs significantly better than the same classifier without entity instantiation data. We also experiment with the incorporation of *machine-identified* entity instantiations. However, our entity instantiation classifier is not sufficiently accurate to impact on discourse relation classification.

1.6 Thesis Overview

This rest of this thesis is structured as follows:

Chapter 2 reviews literature from related research domains, namely Information Extraction, Context-independent Relation Extraction and Bridging Anaphora. We establish the important differences between these fields and the work of this thesis, as well as identifying approaches that may be applicable to our problem.

Chapter 3 details the creation of our corpus of entity instantiations. We detail our motivation, and further define the problem. We describe our annotation principles and methodology. The results of our agreement study are presented, along with a statistical analysis of our final, gold standard corpus.

Chapter 4 presents our supervised machine learning approaches to the automatic identification of entity instantiations. We develop a feature set which is applied to both inter- and intrasentential instantiations. This feature set is augmented by tree kernels for intrasentential entity instantiations, which allow learning directly from syntactic parse trees representing potential instantiations. We evaluate our learners on both the original corpus, and *balanced* data sets containing identical numbers of positive and negative examples. We attain highly significant improvements over our baseline on both original and balanced data sets for intrasentential instantiations, and on the balanced set for intersentential instantiations.

Chapter 5 explores the connection between entity instantiations and discourse relations. We review relevant literature regarding the automatic classification of discourse relations. We then examine the connection between entity instantiations and a specific discourse relation, *Expansion.Instantiation*, in detail. Encouraged by the close correlation between entity instantiations and the Expansion.Instantiation discourse relation, we leverage gold-standard annotations of entity instantiations to aid in the classification of Expansion.Instantiation relations, gaining significant improvements over a strong baseline. We experiment with including machine-learned instantiations as a feature for discourse relation classification.

Chapter 6 provides a conclusion for the thesis. We summarise and reflect on the results of the thesis, and discuss potential future research in this area.

Chapter 2

Literature Review

Our work is closely related to the fields of information extraction (IE), and context-independent relation extraction. It also partially overlaps with the resolution of bridging anaphora. We summarise the historical development and detail the current state-of-the-art of these fields, and also describe how entity instantiations relate to them. We focus on key papers and methodologically similar work to our own, as well as highlighting the particular approaches that inspired our work in the remainder of this thesis.

2.1 Information Extraction

Information extraction is the process of extracting semantic information from natural language in context. Broadly, this semantic information traditionally falls into one of 4 categories, namely:

1. The recognition of named entities, and their classification into semantic types, such as *Person, Organisation* or *Location*. (Grishman and Sundheim, 1996; Miller et al., 1999; Ratinov and Roth, 2009).

2. The detection and classification of semantic relationships between entities, such as *Part-Of, Employed-By, Located-In*. This is referred to as relation extraction (RE).

3. The interpretation of temporal expressions, and the extraction of data related to the temporal ordering of events within a text (Lascarides and Asher, 1993; Pustejovsky

et al., 2003; Lapata and Lascarides, 2004; Mani et al., 2006).

4. Template filling tasks which involve extracting pre-specified types of information about a given event. For example a *Management Succession* event may have fields which represent the company involved, the management position in question, and the incoming and outgoing individual. These more complex tasks involve elements of Tasks 1 and 2.

Task 4 forms an important part of the historical context of IE, and is discussed in Section 2.1.2. Tasks 1 and 3 are not discussed in this thesis, as they are not directly relevant to our work.

Our work is most closely related to Task 2, relation extraction, which we discuss at length in Section 2.1.3.

RE and the detection of entity instantiations are similar tasks; they are both problems involving the discovery of binary semantic relations in *context*. However, there is a fundamental principled difference — we do *not* restrict the participants to mentions of entities representing concrete, real-world objects, but instead consider heterogeneous noun phrases. Despite this important distinction, the similarities mean than many of the methods used are relevant to our work and therefore worthy of discussion.

An additional difference between entity instantiations and RE is the scope of the context considered. Whilst the evidence for an entity instantiation can be drawn from anywhere in the document or from existing world knowledge, RE schemes generally restrict the scope of their relations to within a sentence. The excerpt below from the ACE 2005 annotation manual[1] illustrates this point.

> "We will only tag Relations between entity mentions when the relationship is explicitly referenced in the sentence that contains the two mentions. Even if there is a relationship between two entities in the real world (or elsewhere in the document), there must be evidence for that relationship in the local context where it is tagged."
>
> (ACE English Annotation Guidelines for Relations, v. 5.8.3. LDC, b, p. 5)

Additionally, we note that set membership and subset relations have not been annotated as part of the RE corpora which formed part of the important MUC and ACE programs, nor have any machine efforts been made to identify the phenomena.

SemEval-2 had a shared task, *Multi-Way Classification of Semantic Relations Between Pairs of Nominals* (Hendrickx et al., 2010), which does include a *Member-Collection* relation. However, their task also differs from ours in at least three ways. Firstly, and in

[1] The ACE RE task is discussed fully in Section 2.1.3

contrast to the general IE RE paradigm, they only consider relations which exist only between base NPs with common noun heads — named entities and pronouns are excluded. Secondly, and similarly to ACE/MUC, they do not mark relations which rely on discourse knowledge and restrict annotations to sentence internal relations. Finally, rather than annotating full texts they focus on single sentences extracted from web searches.

Our work is distinct from prior information extraction work, and as far as we are aware, entirely novel.

2.1.1 Early information extraction

One of the earliest IE systems was FRUMP — the Fast Reading Understanding and Memory Program (DeJong, 1979), which processed newswire texts. FRUMP matches a text with a relevant hand-coded 'sketchy script', which is then used to parse the text into a description of the events which occurred. The author attributed the success of the system to not trying to represent each text in some intermediate form — such as a discourse parse, simplified English or a collection of conceptual primitives — but instead developing an architecture based on making and substantiating predictions about what might happen next in the text.

Two other early rule based systems used key word matching to extract information. Zarri (1983) use a hand-coded system to automatically extract details about a person's life story from historical French texts, using verbal keywords. Cowie (1983) use hand-coded rules, written in Prolog, to extract plant attributes from plant descriptions. The system uses a dictionary to match possible descriptive terms in conjunction with some specially coded rules.

2.1.2 Template filling

After the somewhat disparate early IE research, the next phase of IE research centered around the Message Understanding Conferences (MUC) (MUC, 1987-1998). These conferences provided a shared task and data, and uniform evaluation metrics for IE. This reduced the complexity in comparing algorithms, and provided a focus for new research.

Each conference had one or more topics (e.g. Military Fleet Operations, Latin American Terrorist Activities, Corporate Joint Ventures), and the task was *template filling*.

Template filling tasks involve filling set slots in a template with information extracted from the text. For example, Figure 2.1 might result in a filled template for the event of a Fare-Raise Attempt such as the one in Figure 2.2. Each slot in the template has a constraint as to its semantic type — the Lead Airline and Follower slots must be filled by

> Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit of AMR Corp., immediately matched the move, spokesman Time Wagner said. United, a unit of UAL Corp., said the increase took effect Thursday and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Denver to San Francisco.

Figure 2.1: An example paragraph which might be subject to slot filling. Taken from Jurafsky and Martin (2009), pp 759.

Fare-Raise Attempt:	Lead Airline: United Airlines Amount: $6 Effective Date: 2006-10-26 Follower: American Airlines

Figure 2.2: An example of a filled template. Taken from Jurafsky and Martin (2009), pp 759.

an Airline company, the Amount slot must be a monetary value and the Effective Date must be a date.

Accomplishing this task involves identifying the relevant named entities (Task 1 from Section 2.1), identifying relationships between them (Task 2 from Section 2.1 and discussed at length in Section 2.1.3), and consolidating this information into a correctly filled template.

We detail the task, corpus size and example templates for each MUC conference in Table 2.1[2]. The first conference was essentially exploratory, and thus no data is available. MUC-5 offered tasks in languages other than English and MUC-7 offered other tasks in addition to template filling, both of which are excluded from our summary table.

Rule-based approaches. Early approaches to the template filling task varied. The systems developed involved complex architectures, made up of many small processing and parsing steps. Hobbs (1993) describes the generic architecture that had evolved by the end of MUC-4, which comprised a pipeline approach composed of modules representing many now standard NLP tasks, such as tokenisation, parsing and coreference resolution.

Each of the modules in these systems often depended heavily on hand-constructed

[2]The data in this Table was collated from the following sources: Hirschman (1991); Gaizauskas and Wilks (1998); Carlson et al. (1993).

			# Texts		
Conference	**Text source**	**Task**	**Training**	**Testing**	**# Template Slots**
MUCK	Navy Operational Reports	Ship sightings and engagements	N/A	N/A	N/A
MUCK-II	Navy Operational Reports	Ship sightings and engagements	105	5	10
MUC-3	Newswire	Latin American terrorist attacks	1,300	100	18
MUC-4	Newswire	Latin American terrorist attacks	1,300	400	24
MUC-5	Newswire	Joint ventures	1,000	582	47, over 11 templates.
MUC-5	Newswire	Microelectronics	1,000	400	47, over 11 templates.
MUC-6	Newswire	Management Succession	100	100	20
MUC-7	Newswire	Rocket Launches	100	100	7 slots, including 2 relational objects (VEHICLE_INFO, PAYLOAD_INFO)

Table 2.1: Details of the tasks of the Message Understanding Conferences

Example heuristic	**Example phrase that matches heuristic**
<Subject >Passive Verb	<Victim >was murdered
Gerund <Direct Object >	Killing <victim >
Verb Infinitive <Direct Object >	Threatened to <victim >

Table 2.2: Some example heuristics from Riloff (1993).

rules, and many person-hours of work were required to adapt the system to a particular domain. The internal mechanisms of the modules dealing with parsing and semantics differed; some relied on finite state automata (Appelt et al., 1993), some worked with full syntactic parses of the text (Grishman et al., 1991; Montgomery et al., 1992), and others concentrated on partial parses (Ayuso et al., 1992).

Frame-based approaches. Another common approach to the MUC task was to use *frames* to fill templates. The CIRCUS system (Lehnert, 1991), used a hand-created dictionary of frames. For instance, given the word *bombed*, and the fact that the verb occurs in an active rather than passive voice, the system will use syntactic rules to assign the subject of the verb to be the *bomber* and the direct object of the verb to be the *bombed*. GE's NLTOOLSET (Krupka et al., 1991) employed a similar system which combined frames with top-down and bottom-up searches. Both systems scored competitively, achieving a precision and recall of 38%/51% and 46%/42% respectively in MUC-3.

Clearly, creating these frames can be very time consuming — it is estimated in Riloff (1993) that the dictionary of frames from Lehnert et al. (1993) took approximately 1,500 person-hours to construct. The author of Riloff (1993) proposes a method of automatically generating the dictionary of frames, using heuristics such as the ones listed in Table 2.2. These automatically generated rules achieve 98% of the performance of the hand-crafted rules over 200 texts.

The CIRCUS system with automatically generated frames competed in MUC-4 (Lehnert et al., 1992) and MUC-5 (Lehnert et al., 1993). It scored well in MUC-4 (precision and recall of 47% and 57% in TST-3, MUC-4's first test run TST-3, but less so in MUC-5 with an F-score of just 35.18.

Similarly, Cardie (1993) uses a relatively small (120 texts) training set to automatically generate frames for unseen words in a financial corpus. A k-nearest neighbour algorithm is used to predict the part-of-speech, semantic class (e.g. entity, facility, location, etc.) and concept (e.g. tie-up, total-capitalization, ownership-%). The algorithm performs significantly better $(p = 0.01)$ than two baselines; one which selects the majority class and one which selects randomly. Other work adopting a machine-learning approach to

Conference	Highest performance
MUC-3	R <50%, P <70%
MUC-4	F <56%
MUC-5	Joint Ventures F <53%, Microelectronics F <50%
MUC-6	F <57%
MUC-7	F <51%

Table 2.3: A summary of the highest scoring algorithms in MUCs 3–7, from Chinchor (1998)

automatically construct patterns/frames/templates for the template filling task includes Soderland et al. (1995), Kim and Moldovan (1995) and Huffman (1996).

Summary of MUC competition systems. At this point, we note that template filling is a challenging task, and none of the techniques used at the time achieved high levels of precision and recall. In Chinchor (1998), the author provides an overview of scores of the best performing systems in MUC-3 to MUC-7, which we reproduce as Table 2.3.

Template filling with machine learning, using MUC and other corpora. Template filling research did not end with the conclusion of the MUC series, and subsequently, authors began to experiment with machine learning systems for the task.

For example, Collins and Miller (1998) implement a Probabilistic Context Free Grammar (PCFG) to identify the incoming person (IN), the indicating verb (IND) and the post (POST) in MUC-6 management succession events. The tree structures for the training data are automatically constructed based on annotations of IN, IND and POST in the texts. The algorithm scores a Precision of 80.6% and a Recall of 74.6% after being trained on 563 sentences and tested on 356.

Both Freitag (1998) and Freitag and McCallum (2000) train separate learners for each slot in a template. In Freitag (1998) the author applies *SRV*, a supervised learner which uses a top-down greedy rule search, combined with features representing grammatical links, WordNet paths, words and part-of-speech. In Freitag and McCallum (2000) a Hidden Markov Model (HMM) is applied, scoring an average F-Score of 57.2 over 8 slot types, such as the speaker of a seminar announcement event and the deadline in a Call-for-Papers conference announcement.

Following the same methodology of training a learner per slot, Chieu et al. (2003) use a variety of syntactic parse tree derived features, including related verbs, whether the NP is an agent or a patient, the head word of the NP and the NE class of the NP, to learn

template filling for the MUC-4 task. They experiment with 4 supervised machine learning classifiers; Maximum Entropy, Support Vector Machine, Naïve Bayes and Decision Tree. The Maximum Entropy classifier performed best, with an F-Score of 48 on MUC-4's first test data set, a performance which would have placed them 4th in the original evaluation.

More recently, Patwardhan and Riloff (2009) employed a different strategy, which uses two different learners in conjunction for MUC-4 template filling. The first learner decides whether the sentence is describing a relevant event, and the second learner decides which NPs are plausible slot fillers if the sentence is relevant. This approach beats the baseline in 3 out of 5 slots; *Individual Perpetrator*, *Victim* and *Weapon*, with F-Scores for these slots of 55, 56 and 55 respectively.

ACE event detection. The NIST Automatic Content Extraction (ACE) program, a joint IE evaluation that ran from 1999–2008, also provided a similar task to template filling — Event Detection and Characterisation (EDC). This task was introduced in the ACE 2005 evaluation. Rather than the MUC scenario templates, which were highly topic specific, the ACE events were intended to be events which are common in a wide variety of news stories. The list of possible events, from Grishman (2012), is displayed in Table 2.4. Each event has between 2 and 7 slots to fill, compared to the large numbers of slots required in some of the MUC scenario templates.

We also note that ACE EDC events are restricted in scope to a single sentence:

> "The first step in annotating an Event mention is identifying its extent. The extent of an Event mention is always the entire sentence within which the Event is described."
>
> (ACE Annotation Guidelines for Events, v5.4.3, LDC, a, p. 7)

In general, the supervised machine learning methodology used for ACE event extraction is to train one classifier to identify *trigger words* for the event, and classify them according to type, and then train a second classifier to fill the slots of the identified event (Grishman, 2012).

Ahn (2006) breaks this down further, using 4 separate classifiers:

1. Anchor identification: finding event anchors (the basis for event mentions) in text and assigning them an event type;

2. Argument identification: determining which entity mentions, date-time mentions, and values are arguments of each event mention;

Event type	**Subtypes**
Life	Be-born, Marry, Divorce, Injure, Die
Movement	Transport
Transaction	Transfer-ownership, Transfer-money
Business	Start-org, Merge-org, Declare-bankruptcy, End-org
Conflict	Attack, Demonstrate
Personnel	Start-position, End-position, Nominate, Elect
Justice	Arrest-jail, Release-parole, Trial-hearing Charge-indict, Sue, Convict, Sentence, Fine, Execute, Extradite, Acquit, Appeal, Pardon

Table 2.4: The list of ACE Event Types, reproduced from Grishman (2012)

3. Attribute assignment: determining the values of the modality, polarity, genericity, and tense attributes for each event mention;

4. Event coreference: determining which event mentions refer to the same event

Each classifier has a number of features, including the word, the part of speech, the WordNet synset it belongs to, the surrounding context and the dependency relation it has. They attain F-Scores of 0.601, 0.573 and 0.658 for tasks 1,2 and 4, and score highly with individual modality, polarity, genericity and tense classifiers for task 3. The entire system attains an ACE value[3] of 22.3%, which is within the range of the systems that competed in the 2005 evaluation, 19.7% – 32.7%.

Other work has focused on improving performance by creating more advanced machine learning models which incorporate information from beyond the scope of the single sentence containing the possible relation. Ji and Grishman (2008) improve their performance by inferring additional information from topically-related documents, Liao and Grishman (2010) use document-level models to capture event trigger co-occurrence and Liao and Grishman (2011) use an unsupervised document topic model to include information about the relative likelihood of a word being a trigger for an event, given a topic.

Template filling and entity instantiations The development of the template filling task was the catalyst for a great deal of research in the field of IE. Whilst the task is significantly different from that of identifying entity instantiations, we have discussed it both because it plays an important part in the historical development of the field, and because there are some general conclusions that one can draw from this research.

[3]ACE value is the value of system output relative to maximum achievable value. The formula is $ACEValue = 100\% - FA\% - Miss\% - Err\%$) where FA = False Alarms, Miss = Missing and Err = Errors in recognising attributes.

Firstly, we note that, in general, the approaches to the task progressed from complex, modular rule-based systems with long development times to recent approaches based on supervised machine learning, which offer the advantages of shorter development time and a much higher degree of portability to new domains. On this basis, we develop a machine learning based classifier, rather than attempt to construct a complex rule-based system.

We also see that the task itself is challenging — recent supervised machine learning approaches attain F-Scores of < 0.7 for identifying words which trigger events, and < 0.6 for subsequently identifying the correct text to fill the slots of a template. The difficulty of the task has led to a larger focus on smaller and more general IE tasks, such as named entity recognition and relation extraction, and the development of smaller and more general templates. This notion of breaking down complex problems into simpler steps, coupled with the fact that a problem as multi-faceted and difficult as template filling is beyond the scope of a single PhD thesis means that we concentrate on a single, general problem.

2.1.3 Relation extraction

Relation extraction is the information extraction task which is the closest to our task of identifying entity instantiations. Both RE relations and entity instantiations are binary semantic relations between noun-phrases, and methods for RE influence our approach to tackling our problem.

The work considered in this Section poses the problem of RE in the following manner:

> We are given a set E of known entities, and a set R of possible predefined relations. For each pair of entities, E_1, E_2, the task is to assign a label from the set $\mathcal{Y}$, where $\mathcal{Y}$ is a set composed of the relations in R and a special label to indicate that no relation is present. (Sarawagi (2007))

An important point to note is the context dependence of this type of relation extraction. For each pair of entities, E_1, E_2, we are only interested in relations which are explicitly or implicitly stated in the given context, rather than relations based on some form of external knowledge. Example 2.1 shows an example of a semantic relation which is context dependent — a Physical Location relationship between *Bob Dylan* and *London*. Bob Dylan's location is not a fixed property, it will vary over time and the existence of a Physical Location relationship between the two entities in a text depends upon the context.

Alternative forms of relation extraction which are not context dependent and do not fit the definition of relation extraction above are discussed in Section 2.2.

(2.1) Bob Dylan travelled to London this week for a concert.

An early focus for RE was the introduction of the Template Relation task as part of MUC-7. In contrast to the complex template filling task, this task involved the extraction of three binary relationships between named entities in the text:

PRODUCT_OF. The products made by each company.

EMPLOYEE_OF. The employees of each organisation.

LOCATION_OF. The location of the headquarters of each organisation.

The Template Relation corpus for MUC-7 consisted of 1,612 annotated relations. Five template relation systems were submitted to MUC-7, with F-scores in the range of 23.66–75.63.

The highest scoring system, Aone et al. (1998) was based on hand-crafted rules. In contrast, the second highest scoring system, Miller et al. (1998), instead used machine learned rules, and attained an F-Score of 71.23. Their approach was to augment a parse tree with semantic information, such as details of the named entities, and train a statistical parser based on these modified trees. An example augmented tree is shown in Figure 2.3.

After the end of the MUC programs, the NIST Automatic Content Extraction (ACE, 2000-2005) programs begun. In addition to the Event Detection and Characterisation task described in Section 2.1.2, ACE included several other tasks including Entity Detection and Tracking (EDT) which is a combination of Named Entity Recognition and Coreference Resolution, and Relation Detection and Recognition (RDR), which is the detection of semantic relations between entities — RE, in other words.

The exact schema of relations changed throughout the various iterations of ACE. The ACE-2004 scheme, for example, includes 7 broad relation types, divided into a total of 23 subtypes. The full list of ACE-2004 types and subtypes is presented in Table 2.5. Table 2.6 shows an overview of the ACE relation extraction tasks, including the size of each corpus and the number of relations in the schema[4]. We reiterate at this point that no ACE schema contains a relation type or subtype that corresponds to either a set membership or subset entity instantiation.

A variety of automatic RE algorithms have been developed for the ACE RDR task, which generally use a machine learning approach. They fall largely into two groups; those that learn directly from structured data such as trees by using kernels, and those that use traditional, flat features. These two approaches are discussed in Sections 2.1.3.1 and 2.1.3.2 respectively.

[4]We exclude details of both the ACE Pilot corpus and ACE-1, neither of which had an RE task.

Relation type	**Subtypes**
physical	located, near, part-whole
personal-social	business, family, other
employment/membership/ subsidiary	employ-executive, employ-staff, employ-undetermined, member-of-group, partner, subsidiary, other
agent-artifact	user-or-owner, inventor-or-manufacturer, other
person-org affiliation	ethnic, ideology, other
GPE affiliation	citizen-or-resident, based-in, other
discourse	-

Table 2.5: The full ACE-2004 relation schema, as described in Grishman (2012).

Corpus	**# Words**	**# Documents**	**# Relation Instances**	**# Types**	**# Subtypes**
ACE 2002	180K Training, 45K Development/Test, 45K Evaluation	519	7,646	5	24
ACE 2003	100K Training, 50K Evaluation	771	11,069	5	24
ACE 2004	150K Training, 50K Evaluation	451	5,702	7	23
ACE 2005	260K Training, 50K Evaluation	754	10,650	6	18

Table 2.6: An overview of the RDR portions of each ACE evaluation.

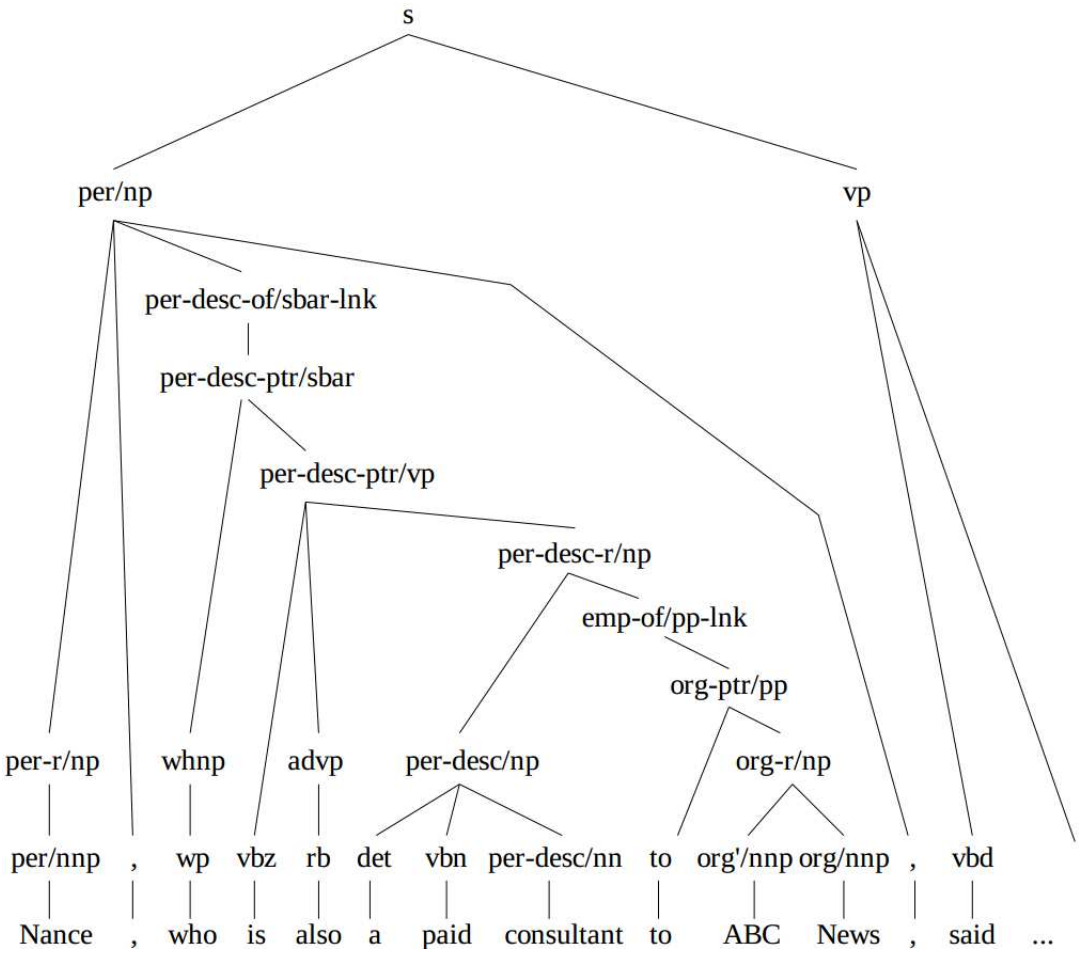


Figure 2.3: An augmented parse tree from Miller et al. (1998).

There is also a research field which deals with the identification of relations within biomedical texts, such those between proteins. We do not discuss this work in this thesis, and instead focus on RE research which deals with texts similar in genre to our own.

2.1.3.1 Approaches using unstructured features

A variety of unstructured featured approaches have been used for RE. For example, Roth and Yih (2002) use a joint-learning approach to classify named entities and relations. They use SNoW (Carlson et al., 1999), a multi-class learner tailored for large scale learning together with a Bayesian Network algorithm to combine the learners. They perform 5-fold cross validation on 2 corpora — one with person entities and a murder-victim relation, and one with person and location entities connected by a born_in relation. For relations, they find that their joint approach gives an improvement in F-Score over the basic (i.e. non-joint) classifier. For the born_in relation task they gain an F1 score of 78.0,

improving over the basic classifier's 70.9, and on the murder-victim relation, the joint classifier scores 62.2 compared to the basic classifier's 58.6.

In Kambhatla (2004), the author uses Maximum Entropy models to perform relation classification on the ACE data set, using the full ACE set of relation subtypes. They use gold standard named entity and mention data. The features that are used fall into 6 categories, the descriptions of which are reproduced below:

Words. The words of both the mentions and all the words in between.

Entity Type. The entity type of both the mentions. Possible values are PERSON, ORGANIZATION, LOCATION, FACILITY, or GPE (Geo-Political Entity).

Mention Level. The mention level (one of NAME, NOMINAL, PRONOUN) of both the mentions.

Overlap. The number of words (if any) separating the two mentions, the number of other mentions in between, flags indicating whether the two mentions are in the same noun phrase, verb phrase or prepositional phrase.

Dependency. The words on which the mentions are dependent in the dependency tree derived from the syntactic parse tree, along with their part-of-speech and chunk labels.

Parse Tree. The path of non-terminals (removing duplicates) connecting the two mentions in the parse tree, and the path annotated with head words.

We note that the dependency and parse tree categories contain flat features derived from trees, *not* tree-kernel features. The author finds that words provide a high precision but low recall classifier (81.9 and 17.4 respectively), and that adding any of the other categories reduces precision but increases recall significantly. They find that adding their parse tree features is most useful, leading to a precision, recall and F-score of 63.5, 45.2 and 52.8 respectively.

The authors in Zhou et al. (2005) take a similar approach, but use a richer feature set and an SVM classifier. They also include features based on base phrase chunks, and some semantic data from Wordnet to aid identification of country names and familial relationships. They find that their chunk based features are useful, but the dependency and parse tree derived features do not make much of a difference because most of the relationships in the ACE corpus occur between mentions that are separated by very few words. Their best precision, recall and F-score on the full set of subtypes is 63.1, 49.5 and 55.5. On the 5 coarse-grained types, they score 77.2, 60.7 and 68.0.

Two recent flat-featured approaches successfully exploit background knowledge to improve RE performance. Chan and Roth (2010) implement features which use queries to search for taxonomic *parent-child* relationships — such as *'George W. Bush is a child of Presidents of the United States'* (Do and Roth, 2010) — between Wikipedia entities. They attain an F-score of 68.2% at the coarse-grained level and 54.4% at the fine-grained level on a set of directed, sentence-internal relations from the ACE-2004 data set. Sun et al. (2011) generate large-scale word clusters from an 83 million word corpus and incorporate information regarding which cluster the mention head word belongs to. This method results in an F-score of 71.5% on coarse-grained ACE-2004 relations.

2.1.3.2 Kernel approaches

The first paper to apply a tree kernel approach to RE was Zelenko et al. (2003), who attempted to extract *person-affiliation* and *organization-location* relations. They used SVM and Voted-Perceptron learners which learn directly from shallow parsed sentences. They found that their tree kernel methods fare as well, if not better than their flat-featured counterparts, as well as running faster and taking less development time.

Rather than shallow parses, other work has considered different trees from which to learn. Culotta and Sorensen (2004) use dependency parse tree kernels on the ACE-2003 corpus, and achieve an F-Score of 45.8 on the 5 coarse-grained types. This is extended by Bunescu and Mooney (2005), who included only the shortest path between two entities in a dependency tree, gaining an F-Score of 50.5 on coarse-grained ACE-2002 relations. Figures 2.4 and 2.5 show examples of the trees presented to the learners in these two approaches, respectively. Zhao and Grishman (2005) combine both constituency and dependency kernels, achieving a best F-score of 70.4 on ACE-2004 coarse-grained relations.

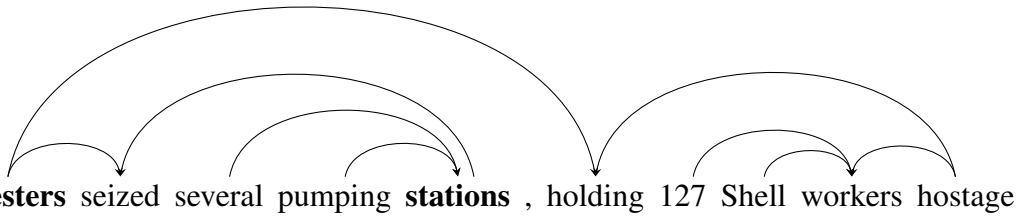


Figure 2.4: The full dependency tree for the sentence containing the relation Protesters AT stations, as presented to the learner by Culotta and Sorensen (2004)

An alternative approach is that of Bunescu and Mooney (2006), who implement a subsequence kernel, which computes the number of common segments between sentences.

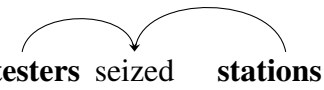


Figure 2.5: The shortest path dependency tree for the sentence considered in 2.4, as presented to the learner by Bunescu and Mooney (2005)

They use the coarse-grained relations in the ACE-2003 corpus, gaining an F-score of 47.7.

Zhang et al. (2006) *combine* tree kernels and flat features for relation extraction. They experiment with 5 different constituency parse tree kernels and find that their best performing tree kernel is the subtree representing shortest path between the two entities including intervening leaves. This achieves a precision, recall and F-score of 72.8, 53.8 and 61.9 on the ACE-2003 data set. An example of this subtree, referred to as the path-enclosed tree is shown in Figure 2.6. The authors suggest that their other types of tree perform worse because they include too much context to the left and right of the entities, introducing noisy features and encouraging over-fitting.

Their flat features consist of the headword of each entity, the entity type and subtype and the mention type, and achieves scores of 75.1, 42.7 and 54.4. A polynomial combination of flat features and the tree kernel scores 76.1, 68.4 and 72.1, a significant performance improvement over either kernel in isolation.

Zhou et al. (2007) also use a combination of tree kernels and flat features. Their innovation is to suggest that in cases like *'John and Mary get married'* the tree between the two entities (i.e. the tree which spans just *'John and Mary'*) does not contain enough information to classify the relation, and instead in these cases the predicate should also be included. They implement an algorithm which dynamically decides how much context to include as part of the tree, and in conjunction with their flat features it achieves an F-score of 75.8% on the 7 coarse-grained relation types in the ACE-2004 data set, and 66.0 on the 23 fine-grained types.

Another approach is to augment a tree structure with information such as POS data, entity type data and indications as to whether each node is part of one of the entities. Then one can learn from this more complex tree (Jiang and Zhai, 2007). Coupled with some heuristics, such as removing nodes from the parse tree that are not part of the Path-enclosed Tree, and nodes that correspond to articles, adjectives and adverbs, they attain a best F-Score of 72.9 on the 7 major types of the ACE-2004 data set.

Where as the approaches discussed so far apply tree kernels to intra-sentential relations, Swampillai and Stevenson (2011) apply tree kernels to *inter-sentential* relations. They experiment with a version of the MUC6 corpus with relations annotated between

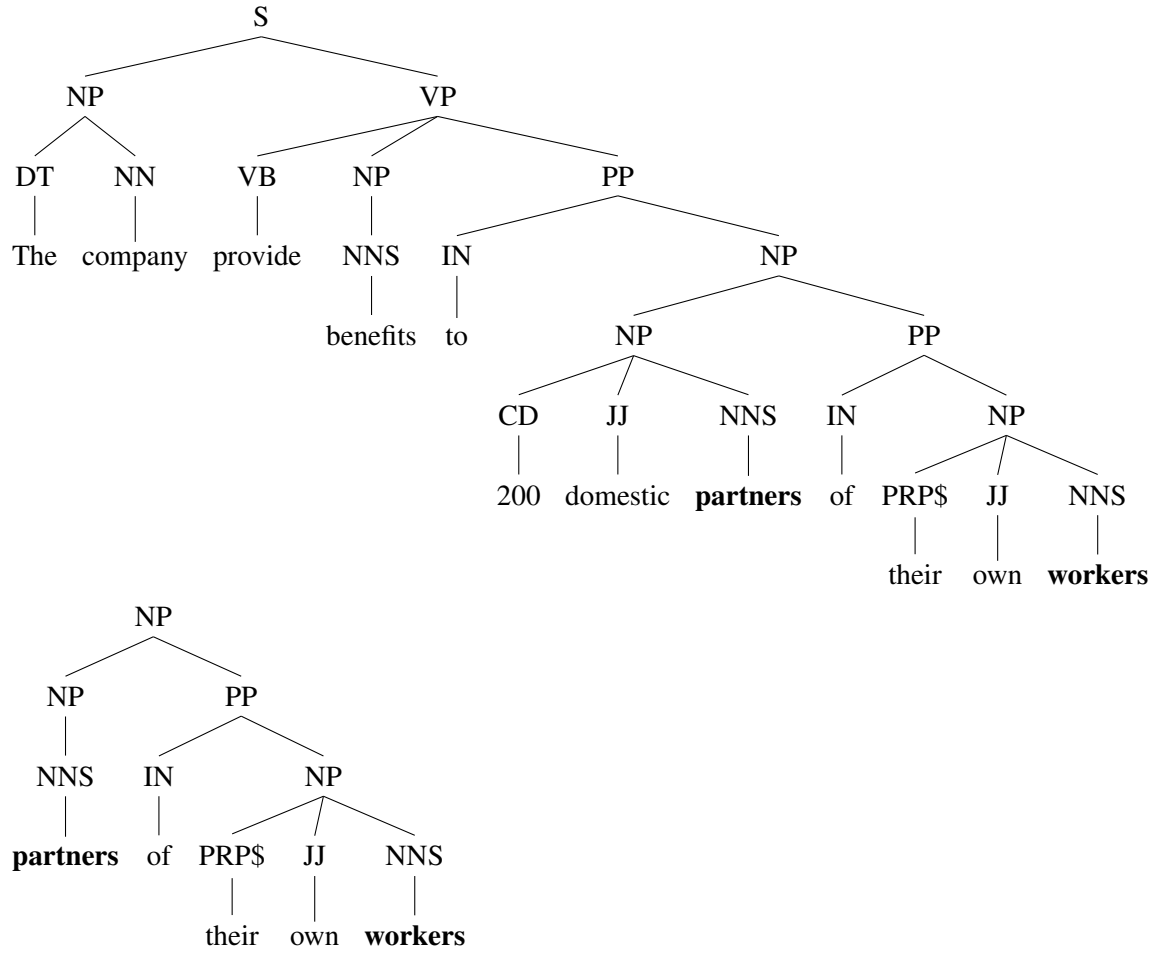


Figure 2.6: A sentence parse tree with a SOCIAL.Other-Personal relation between *partners* and *workers*, and the path-enclosed tree for the relation.

sentences, which is introduced in Swampillai and Stevenson (2010).

Instead of a complex template, in this version of the corpus the templates are converted into binary relations between either a person and a post (PerPost), a person and an organisation (PerOrg) and an organisation and a post (OrgPost). These intrasentential relations are then supplemented by intersentential annotation of the same three relations, over a total of 200 documents.

Similarly to Zhang et al. (2006) and Zhou et al. (2007), they experiment with flat features, syntactic parse tree kernels and a combination of the two. They use two types of tree; the Shortest Path Enclosed Tree (SPET), which is identical to the Path-enclosed tree described in Zhang et al. (2006) and the Shortest Path Tree (SPT) which excludes nodes in between the two entities which are not part of the shortest path. To overcome the fact that intersentential relations are not part of the same syntactic parse tree, they artificially join the tree for each sentence together with a node labelled `ROOT`.

		Best F-score	
Algorithm	**Approach**	**5 types**	**24 subtypes**
Kambhatla (2004)	Unstructured	—	52.8
Culotta and Sorensen (2004)	Dependency Tree Kernel	45.8	—
Zhou et al. (2005)	Unstructured	68.0	55.5
Bunescu and Mooney (2006)	Subsequence Kernel	47.7	—
Zhang et al. (2006)	Combination	70.9	57.2
Zhou et al. (2007)	Combination with Dynamic Context	**74.1**	**59.6**

Table 2.7: Comparison of relation extraction results on the ACE-2003 corpus.

Their flat features comprise a window of 12 tokens and POS tags surrounding each entity, the two nearest dominating verbs for each entity and an intersentential specific distance feature, corresponding to the number of intervening sentences between the entities.

Intersentential and intrasentential relations are trained and tested on separately, and 10-fold cross validation is employed. Their best performing flat features used a window size of 12, and all features. Their best performing tree kernel was the SPT kernel. A combination of SPT and the best flat features exceeded the performance of either in isolation, though not significantly in the case of intersentential PerOrg and PerPost relations. This combination method achieved PerOrg/PerPost/PostOrg F-scores of 0.651/0.200/0.765 for intersentential relations and 0.699/0.652/0.750 for intrasentential relations.

2.1.3.3 Comparison of research from Sections 2.1.3.1 and 2.1.3.2

The work described in Sections 2.1.3.1 and 2.1.3.2 varies in terms of the data sets used (ACE-2003, ACE-2004, MUC6), and the granularity of the relations learned, making comparisons difficult. However, we collate the best reported results from each paper in terms of F-score, organised by corpus. Table 2.7 shows the best results on the ACE-2003 data set, Table 2.8 shows the best results on the ACE-2004 and Table 2.9 shows the best results from work based on other corpora.

2.1.3.4 Connection to entity instantiations

The RE task described in this Section is closely related to our own task, and the important questions raised by the research also require answers for our problem. The research indicates that both unstructured and kernel approaches have merit, and the best approaches involve the combination of the two. We will therefore experiment with both types of features.

		Best F-score	
Algorithm	**Approach**	**7 types**	**23 subtypes**
Chan and Roth (2010)	Unstructured with Background Knowledge	68.2	54.4
Sun et al. (2011)	Unstructured with Background Knowledge	71.5	—
Zhao and Grishman (2005)	Dependency and Constituency Tree Kernels	70.4	—
Zhang et al. (2006)	Combination	72.1	63.6
Zhou et al. (2007)	Combination with Dynamic Context	**75.8**	**66.0**
Jiang and Zhai (2007)	Augmented Tree Kernel	72.9	—

Table 2.8: Comparison of relation extraction results on the ACE-2004 corpus.

Algorithm	**Corpus**	**Approach**	**Best F-Score**
Roth and Yih (2002)	TREC documents annotated with *murder_victim* and *born_in* relations.	Unstructured	78.0 (*born_in*) 62.2 (*murder_victim*)
Zelenko et al. (2003)	Corpus of sentences annotated with *person-affiliation* and *org-location* relations.	Kernel	86.8 (*person-affiliation*) 83.3 (*org-location*)
Bunescu and Mooney (2005)	ACE-2002	Kernel	52.5
Swampillai and Stevenson (2011)	MUC6 Per-Org/PerPost/PostOrg relations.	Combination	65.1/20.0/76.5 (intersentential) 69.9/65.2/75.0 (intrasentential)

Table 2.9: Comparison of relation extraction results on corpora other than ACE-2003 and ACE-2004.

In developing our flat feature set, we take inspiration from repeatedly used features, including those that represent words, part-of-speech and overlap. We also note the success of Chan and Roth (2010) and Sun et al. (2011) who improve their results by attempting to incorporate world knowledge. Incorporating such information seems likely to be helpful for our task.

However, there are at least two important differences between RE and entity instantiations; the type of noun-phrases employed in the relations and the scope of the context considered. Additionally we note that almost all RE research tackles intrasentential relations, and both unstructured and kernel approaches are developed with this constraint in mind. We tackle both intra- and intersentential relations, and expect these three differences will mean that not all RE features will be suitable for our task, and that it will be necessary to develop new features that are particular to our problem.

Swampillai and Stevenson (2011) are an exception in terms of the scope of their relations and context considered, and they detail a method for tree-kernel learning of intersentential relations that may be applicable to our problem. However, their method of simply joining unrelated trees under a new root node lacks theoretical grounding and as such, we do not use their method.

2.1.4 Unsupervised relation extraction

A subtopic of relation extraction considers an unsupervised approach. The problem tackled generally follows the definition below:

> We are given a large corpus, and possibly a list of known entities E and must automatically induce a set of relations R and between entity pairs. The set of relations is not predefined. (Sarawagi (2007))

The algorithms in this Section share the characteristic of receiving no supervision, either by means of training data or seed instances of relations, and the classes of relations are not determined in advance but are instead discovered from the data.

We also note that this problem is still *context dependent*. In fact, the context of the relations is often used as a mechanism for inducing the relations.

These methods could theoretically be used to discover entity instantiations, and so we discuss them. However, none of the work detailed mentions set member or subset relations, instead focusing on similar relations to those considered by supervised relation extraction methods.

Hasegawa et al. (2004) extract named entities which occur within the same sentences and are at most N words apart, along with the intervening context. Each entity pair is

represented as a feature vector, and feature vectors which have the same NE types are grouped — i.e. a PERSON-GPE relation is only compared with another PERSON-GPE relation.

For a given entity pair, the vector consists of a bag of words of all intervening context words from all occurrences of the NE pair in the corpus. Each word is weighted according to their importance in the corpus, and vectors are compared using cosine similarity.

For evaluation purposes they manually identify a set of 177 distinct PERSON-GPE entity pairs in newswire texts and classify these into 38 relations they perceive to exist based on their manual inspection. The same procedure is applied to COMPANY-COMPANY relations, finding 65 distinct entity pairs and 10 relations. Each cluster generated by the algorithm was manually inspected to determine the relation which represented the majority of each cluster. F-scores of 80 and 75 are recorded on the PERSON-GPE and COMPANY-COMPANY relations. PERSON-GPE relations discovered included *President, Governor* and *Senator*. COMPANY-COMPANY relations included *Merger & Acquisition, Rival* and *Parent*.

Zhang et al. (2005) note two problems with Hasegawa et al. (2004):

- Hasegawa et al. (2004) make the assumption that the same entity pairs in different sentences always represent the same relation. Zhang et al. (2005) discover that 9.88% of low frequency, 24.4% of intermediate frequency and 15.4% of high frequency distinct entity pairs have more than one relation in the corpus in the PERSON-GPE domain. A similar trend is observed in the COMPANY-COMPANY domain.

- The cosine similarity of flat features only considers the words between entity pairs, and does not take into account any syntactic structure.

They address these issues by removing the assumption about entity pairs and implementing a constituency parse tree similarity measure. They use the same evaluation corpus as Hasegawa et al. (2004), and on entity pairs with a co-occurrence frequency of over 30 exceed their F-measure by 5 percentage points on the PER-GPE pairs (87 vs 82) and 3 percentage points on the COM-COM pairs (80 vs 77).

Other related work includes Chen et al. (2005), who use feature selection, a matrix-based comparison metric and a method for automatically estimating the number of clusters to improve results, and Rosenfeld and Feldman (2007), who compare 5 different clustering techniques over 3 corpora, with a variety of feature sets.

Another interesting approach to unsupervised relation extraction is that of Banko et al. (2007). They employ a technique that they describe as *self-supervision*, in which they use

some heuristics to automatically label a small amount of data, which is then used to train a classifier which is applied to a much larger corpus. To automatically generate the training they extract the syntactic parse tree path between base noun phrases[5], and paths that are over a certain length, involve a pronoun or cross clauses are marked as negative. All other examples are marked as positive, and a generalised version of the connecting phrase between each entity pair is stored to represent the relation.

This self annotated data then trains a Naïve Bayes classifier, which runs one pass over a large corpus extracting relations. On a 9 million web page corpus, with 133 million sentences, the algorithm scores an error rate of 12% over 11,476 extracted instances from 10 frequent classes.

A problem with this approach is the number of synonymous relations that are considered distinct. For example, the system considers the following tuples as distinct:

('Bletchley Park', 'being called', 'Station X')
('Bletchley Park', ', known as', 'Station X')
('Bletchley Park', ', codenamed', 'Station X')

Yates and Etzioni (2007) attempt to solve this problem with a two step clustering process; clustering of synonymous entity names and clustering of synonymous relations. The two steps are repeated iteratively and feed into each other — with better entity name clusters comes better relation clusters and vice-versa. They experiment with a data set of 2.1 million assertions, measuring precision by manually inspecting each cluster and estimating recall on a small subset of the data. Their best method achieved a precision, recall and F-score of 0.78/0.68/0.73 on entities and 0.90/0.35/0.50 on relations.

Other work which uses a joint learning approach between entities and relations includes Kok and Domingos (2008), who use Markov logic networks to produce a probabilistic graphical model, and Yao et al. (2011) who use a technique called Latent Dirichlet Allocation, which is more commonly used to create document topic models.

Whilst in the future these methods could be used for entity instantiation learning, in this thesis we employ solely *supervised* machine learning for the problem. We had a need to establish that the problem was well defined, so we annotated a corpus of entity instantiations. This then generated examples for supervised machine learning, which also tends to be more accurate than unsupervised methods.

A second reason for using solely supervised methods is that it avoids having to deal with the other sorts of relations an unsupervised method could discover, meaning we can focus entirely on the relations that interest us.

[5]Base noun phrases are those which do not contain nested noun phrases or other modifying phrases.

2.2 Context-independent Relation Extraction

Section 2.1 has described exclusively context dependent information extraction research. In this Section, we consider the following two relation extraction problems (Sarawagi, 2007):

1. We are given a relation r, or alternatively a small set of *seed* entity pairs which represent the relation r, $S = \{(a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)\}$. Over a large corpus, harvest as many examples of the relation r as possible.

2. We are given several seed entities $E = \{E_1, E_2, \ldots, E_n\}$ which belong to the same class — in other words all have an *is-a* relationship with some other entity e. Over a large corpus, harvest as many entities which also belong to the same class as possible.

As the objective is retrieving as many good examples of each relation or class as possible, the focus is on relationships that hold true regardless of context. Both problems are often referred to as *minimally supervised*, as the only supervision provided is the selection of the seed set.

We also note that these approaches have often been used for the extraction of *lexical* relationships, which can exist between any words or concepts rather than just between entities. Often, the goal in these cases is automatic construction of thesauri or ontologies. However, the methods used in both research communities inform and build upon each other, and so we group them together in our discussion.

Experimenting with very large unlabelled corpora can make evaluation problematic, especially compared to the relatively straightforward evaluation associated with supervised relation extraction (Grishman, 2012). Clearly, it is implausible to manually inspect thousands of documents or millions of web pages to check which relations have been missed, and so the focus is often on high precision instead of comprehensive recall.

The most common approach to evaluation involves manually inspecting a small subset of relations to verify their precision, but other evaluation techniques which estimate precision and recall are also used.

In Section 2.2.1 we summarise some notable methods for tackling Problem 1. Section 2.2.2 describes methods for tackling Problem 2.

2.2.1 Minimally supervised relation extraction

Hearst (1992)'s patterns. One early important insight into the problem of context-independent relation extraction was that of Hearst (1992), who suggested that *patterns*

Pattern	**Example Match**
such NP as {NP,}* {(or | and)} NP	*works by such authors as Herrick, Goldsmith, and Shakespeare.*
NP {,NP}* {,} or other NP	*Bruises, wounds, broken bones or other injuries*
NP {,} including {NP,}* {(or | and)}	*All common-law countries, including Canada and England*

Table 2.10: Example patterns from Hearst (1992)

could be employed to automatically harvest examples of relations. The author identified a number of lexical patterns which were strong indicators of hyponymy relations (i.e. *X is-a Y*). Table 2.10 shows some of the patterns described, along with example matches.

The first pattern from Table 2.10 was applied to an 8.6 million word academic encyclopedia, with the matches restricted to instances where both the hyponym and hypernym are either unmodified NPs or NPs consisting of two nouns or a present/past participle and a noun, and 330 examples were found.

Much of the work in this field has followed this initial insight, expanding the work by exploring which patterns to use, how to automatically generate new patterns, and how to evaluate the utility of patterns. A number of methods have also applied pattern-based techniques to relations other than hyponymy, such as meronymy (X is a part of Y) and a variety of application specific entity relations.

Finding new patterns and evaluating their utility. Hearst (1998) expands on her prior work by suggesting an algorithm for finding further patterns. For a given relation, such as hyponymy, word pairs are extracted from WordNet (Fellbaum, 1998), a lexical database containing semantic relations. A large corpus is then searched for sentences containing both words of the pair, and manually inspected to discover new patterns.

Rather than relying on manual inspection to discover patterns, much work has explored automatic methods for discovering new patterns. With the automatic discovery of new patterns comes the need to automatically judge how effective the generated patterns are, and potentially the need to filter the results of the patterns to exclude erroneous examples.

Riloff and Shepherd (1997) use a *bootstrapping* approach to discover hyponyms. They exploit the fact that members of the same semantic category (i.e. words that are all hyponyms of some other term) often appear in lists, appositives, conjunctions and noun compounds, and use a number of seed terms to collect relevant examples of these grammatical

constructions, rather than hand-specifying the patterns in advance. Co-occurrence statistics are used to rank new examples of the category, and the bootstrapping occurs when the best new examples become part of the seed set of the next iteration of the algorithm. Roark and Charniak (1998) follow the same paradigm but introduce a different measure of co-occurrence, along with better ranking and seed selection techniques to improve results.

Another important early approach employing automatic pattern discovery was that of Brin (1999), who tackled the problem of discovering (author, title) pairs from the Web rather than hyponymy. They use a technique called Dual Iterative Pattern Relation Extraction (DIPRE), which consists of 5 steps:

1. Start with a small sample of the target relation. In this case, the authors use 5 (author, title) pairs.

2. Find all occurrences of the seed pairs in the corpus — the Web in this case — by searching for proximate mentions of a corresponding author and title in a text.

3. Generate patterns based on the occurrences found in Step 2, based on their context

4. Search the corpus for tuples which match the generated patterns. Update our original sample set with the matching tuples.

5. If the set of tuples extracted is sufficiently large, end. Otherwise go to Step 2.

The most important and complex stage of this process is the pattern generation. Patterns which return bogus tuples lead to bogus patterns in the next iteration. Over a number of iterations the relation tuples harvested may no longer represent the original relation that was intended to be discovered — a process known as *semantic drift*. For instance, the accidental introduction of a (character, title) pair might lead to semantic drift of the relations away from (author, title) pairs.

Their pattern generation algorithm involved finding examples where the context between the author and title was identical, and included preceding and subsequent words if they were also identical. Applying DIPRE to the Web resulted in over 15,000 (author, title) pairs being discovered. The output was evaluated by checking a random subset of 20 extracted pairs for correctness, by searching online book databases such as Amazon. 19 of the 20 were correct, with the 1 error being an article rather than a book. As such a small subset of the output data was evaluated in this way, it is hard to firmly establish the success of the algorithm.

Agichtein and Gravano (2000) build upon Brin (1999)'s DIPRE algorithm. Rather than (author, title) pairs, they concentrate on the discovery of (company, location of headquarters) pairs. They improve by using a Named Entity tagger, and making their patterns

more flexible, allowing for partial matches and weighting of preceding, intervening and subsequent context. They also estimate the confidence of each pattern by comparing its matches to previously harvested tuples that have high confidence. They manually inspecting a random sample of 100 relation pairs, finding that their algorithm only makes 7 errors, compared to 26 for DIPRE and 75 for a co-occurrence based baseline.

Automatically filtering the results retrieved by a harvesting algorithm to remove false positives can considerably improve performance. Filtering approaches applied to hyponymy extraction include using a vector-based model of word similarity called Latent Semantic Association (LSA) (Cederberg and Widdows, 2003), and applying a graph representation to model the utility of harvested instances (Kozareva et al., 2008).

Filtering results is especially important for meronymy, as the patterns which identify meronymy often can express other relations. For example, Berland and Charniak (1999) use the pattern <whole noun>'s <part noun> which would match a meronymy example such as *'the building's basement'*, but could also match a possessive relation such as *'the man's dog'*. The filtering procedure Berland and Charniak (1999) apply estimates the likelihood that the two nouns are actually part of meronym relationship, based on calculating the difference between $p(whole|part)$ and $p(whole)$. For a given seed word representing a whole, they report accuracies of 55% for the top 50 part words extracted, and about 70% for the top 20. These relatively low results also reflect the fact that meronymy is a problematic relation to discover, because it is a complex relation, with Iris (1989) suggesting that it "should be treated as a collection of relations, not as a single relation", and both linguistic (Winston et al., 1987) and applied (Fellbaum, 1998) authors choosing to break the relation down into sub-relations such as *member-of-collection* and *stuff-of*.

One important and comprehensive method for minimally supervised RE, including a bootstrapping approach and a complex filtering mechanism is that of Pantel and Pennacchiotti (2006). Their algorithm, *Espresso*, uses Pointwise Mutual Information (PMI) based measures to better model the reliability of both patterns and instances. PMI is a metric for measuring the strength of association between two events:

$$pmi(x,y) = \log \frac{P(x,y)}{P(x)P(y)}$$

They define their pattern and instance reliability measures in terms of each other — a reliable pattern is evidenced by reliable instances, and a reliable instance is the production of a reliable pattern. The reliability of a pattern p, $r_\pi(p)$ is defined as an average of the PMI over a set of input instances I, weighted by the reliability of each instance, $r_\iota(i)$:

$$r_\pi(p) = \frac{\sum_{i \in I} \left(\frac{pmi(i,p)}{max_{pmi}} * r_\iota(i) \right)}{|I|}$$

Similarly, $r_\iota(i)$ is an weighted average of PMI over a set of patterns, P:

$$r_\iota(i) = \frac{\sum_{p \in P'} \frac{pmi(i,p)}{max_{pmi}} * r_\pi(p)}{|P|}$$

Prior work had targeted *specific* patterns, with high precision but low recall. Pantel and Pennacchiotti (2006) also exploit *generic* patterns — ones with low precision and high recall — by evaluating the quality of each instance a pattern generates and retaining good instances. They do this by measuring the PMI of an instance with a series of reliable patterns collected in previous iterations, based on Google queries.

They compare their algorithm to two other semantic relation harvesters, and consistently score higher precision over 5 relations, including hyponymy, meronymy and application specific relations such as corporate succession. They compare results with and without the inclusion of generic patterns, and find that including generic patterns tends to engender a small (up to 10%) drop in precision, but increases recall by between one and two orders of magnitude.

Alternatives to pattern-based approaches. One notable method not based on patterns is that of Mintz et al. (2009), who experiment with a different method of supervising their algorithm, *distant supervision*. They use Freebase[6] (Bollacker et al., 2008), a large online database of structured data with 116 million instances of 7,300 relations between 9 million entities at the time of their experiments, to train a classifier for relation extraction. Their intuition is that a sentence which contains a pair of entities that are connected by a Freebase relation is likely to be a mention of that relation.

To train their algorithm they trawl an unlabelled corpus searching for sentences that contain a pair of related entities and extract a number of features, including words, part-of-speech tags, contextual words, dependency parse paths and named entity types. Negative training data is generated by searching for sentences containing entities that are in Freebase but not related.

For human evaluation, the algorithm is trained on 800,000 unlabelled Wikipedia articles and then extracts relations from a further 400,000 unlabelled Wikipedia articles. The

[6] `http://www.freebase.com/`

entirety of Freebase's 1.8 million relations are employed in training, and the extraction process only harvests relations not already in Freebase.

The human evaluation is then performed using Amazon's Mechanical Turk service. Samples of 100 instances from the first 100 and 1000 instances extracted of the 10 most commonly found relations were sent to Mechanical Turk, where they were labelled by one to three annotators. The test is performed with purely lexical features, purely syntactic features and the full feature set. The average precision over the 10 relations was relatively similar, regardless of the feature set, with scores in the range of 67%–69%.

For a machine evaluation, they hold out half of the Freebase relation set and run the algorithm as before but with the other 900,000 relations for training. The resulting extracted instances are then compared to the held out relations. They measure precision over recall levels of 10 to 100,000 relation instances, with results ranging from 0.9 for 10 instances to 0.2 for 100,000.

2.2.2 Minimally supervised set extraction

In this Section we discuss the problem of extracting sets of entities which belong to a given class. This is an implicit hyponym link between all the members of a class and a some other concept.

This problem is related to entity instantiations in that both involve set membership. However, our work differs in two ways; we are not restricted to set membership between entities and our set membership is context dependent.

Etzioni et al. (2005) implement a system for collecting sets of named entities from the World Wide Web (WWW), using search engine queries and a method inspired by Hearst (1992). The algorithm concentrates on simple noun phrases, and retrieves examples matching a pattern from the WWW. Extracted examples are then assessed by PMI queries, which measure the strength of association between the extracted example and the class. A Naïve Bayes classifier learns what level of association is needed for an example to be judged as correct, based on training data acquired with minimal supervision. The standard algorithm achieves a precision/recall of 0.98/0.76 for cities, 1.0/0.98 for US States, 0.97/0.58 for countries. Performance is calculated by comparing results to a gazetteer.

They improve their recall in three ways. Firstly, they automatically learn new extraction patterns from the web. Secondly, they implement a method for subclass extraction, which involves finding examples of a subclass of the original class. For instance, rather than searching for *Scientist*, they automatically discover that *Physicist, Chemist* and *Bi-*

ologist are subclasses, and extract examples for them. Finally, they use a method which exploits the structured nature of some web pages to extract lists of items more easily. The list extraction method was particularly helpful, leading to an extraction rate *forty* times greater than the other methods.

Paşca et al. (2006) make two contributions. Firstly, they introduce a method for generating generalised extraction patterns based on word similarity. This vastly reduces the number of iterations over a corpus required to add new patterns, and does not require parsers or named entity recognisers. Secondly, their method for calculating the quality of extracted examples and generated patterns also uses word similarity. For example, a pattern searching for Language-SpokenIn-Country facts is likely to be better if it contains words similar to *language* or *spoken*.

Other set expansion work includes Sarmento et al. (2007), who use a vector space model to calculate the similarity between set elements and take advantage of explicitly stated lists of entities in Wikipedia, and Wang and Cohen (2007) who exploit semi-structured web pages for their language independent system.

2.2.3 Differentiating entity instantiations from context-independent relation extraction

There are two major differences between our work and that described in this Section. Firstly, these methods return relations that do not depend on context, where as entity instantiations are often context dependent. Secondly, whilst we consider heterogeneous noun phrases, these methods are more focused on either entities or, in the case of more lexically motivated work, simple nouns or noun phrases.

On the other hand, the output of these methods is useful in identifying instantiations based upon world knowledge, and so we use a feature based upon one of Hearst (1992)'s patterns to include knowledge of context-independent relations in our learning process. Clearly, there was an opportunity to use some of the more sophisticated techniques described in this Section — such as including more patterns, bootstrapping or filtering the results — and this is certainly an avenue for future work. However, as this is an initial study of the problem of entity instantiations our focus was on more basic features, representing syntax, salience and context.

2.3 Bridging Anaphora

In this Section we discuss bridging anaphora and its connection to entity instantiations. Firstly, we describe bridging in the wider context of anaphora resolution and information status, before discussing prior work in theoretical and corpus linguistics. We then review some relevant bridging annotation efforts, and algorithms for automatically resolving bridging references.

2.3.1 Anaphora and bridging

An *anaphor* is a reference to an entity previously introduced in the discourse, known as the *antecedent* (Jurafsky and Martin, 2009). Many researchers tackling this problem apply a stricter definition, where an expression is considered anaphoric *if and only if* it cannot be interpreted without its antecedent (van Deemter, 1992; van Deemter and Kibble, 2000; Modjeska, 2004).

Example 2.2 shows a pair of sentences with two anaphoric references[7]; *He* is an anaphoric reference to the antecedent *John*, and *it* refers back to *an apple*. Both these anaphoric references represent an *identity* relation, and both are examples of pronominal anaphora.

(2.2) *John* ate *an apple*. **He** thought **it** was very tasty.

Pronominal, identity-based anaphora is a well studied problem (Hobbs, 1978; Brennan et al., 1987; Lappin and Leass, 1994; Soon et al., 2001; Yang et al., 2003, *inter alia*). It also overlaps[8] with the broader problem of *coreference resolution*. Coreference resolution is the process of determining which mentions in a text refer to the same discourse entity. The mentions are not restricted to pronominal or anaphoric mentions. In Example 2.3 the following four mentions make a coreference *chain* referring to Barack Obama: {*Barack Obama, his, President Obama, his*}. The two mentions of *his* are anaphoric, the two other mentions are not.

(2.3) Barack Obama made his State of the Union address today. President Obama discussed several topics in his speech, including defence policy and tax increases.

[7] Antecedents are displayed in *italics* and anaphors are displayed in **bold** throughout the examples in this Section.

[8] See van Deemter and Kibble (2000) for a full discussion of the relationship between anaphora resolution and coreference.

Coreference resolution is also an area that has received a great deal of attention (McCarthy and Lehnert, 1995; Kehler, 1997; Cardie and Wagstaff, 1999; Raghunathan et al., 2010; Ng, 2010, *inter alia*). We do not consider identity relationships as part of our entity instantiation problem, and therefore we do not discuss them further in this thesis.

There are, however, a number of other anaphoric relationships that are not based upon identity, such as *comparative anaphora* and *bridging anaphora*. Comparative anaphora are anaphoric expressions linked to their antecedent by means of a comparison (Modjeska, 2000). Examples 2.4 and 2.5, identified in Markert and Nissim (2005) from the Penn Treebank, demonstrate this phenomena.

(2.4) In addition to *increasing costs* as a result of greater financial exposure for members, these measures could have **other, far-reaching repercussions.**

(2.5) The ordinance, in Moon Township, prohibits locating *a group home for the handicapped* within a mile of **another such facility**.

These examples contrast with entity instantiations in that they form a set-complement relationship (Markert and Nissim, 2005). In Example 2.4, there is an implicit set of *repercussions*, R, where *increasing costs* $\subset R$, *other, far-reaching repercussions* $\subset R$ and *increasing costs* $\not\subset$ *other, far-reaching repercussions*.

We are interested in explicitly stated sets and their members, and so these anaphora are also not considered as part of our problem. Another anaphoric relationship that *does* have overlap with our problem is *bridging anaphora*. Bridging anaphora require some inference to 'bridge' the gap between the anaphor and the antecedent (Clark, 1975). The classical example is in the form of meronymy, as in Example 2.6, from Jurafsky and Martin (2009).

(2.6) I almost bought *a 1961 Ford Falcon* today, but **a door** had a dent and **the engine** seemed noisy.

However, bridging anaphora can also be connected to their antecedent by set membership, as in this example from Clark (1975):

(2.7) I met *two people* yesterday. **The woman** told me a story.

However, not all entity instantiations are bridged. In Example 2.8, the set member, *Wayne Rooney*, is not anaphoric.

(2.8) **Footballers** are vastly overpaid. Manchester United pay *Wayne Rooney* £200,000 per week.

In the remainder of this Section we focus mostly on bridging work which explicitly considers set membership.

2.3.2 Linguistic bridging research

One early linguistic work which identified bridging anaphora was Clark (1975). The author describes a series of *implicatures* which the reader may use to bridge from previous knowledge to the intended antecedent. The implicatures include necessary, probable and inducible parts of the antecedent (Examples 2.9, 2.10 and 2.11 respectively), necessary and optional roles the anaphor may play in an antecedent event (Example 2.12) and causal implicatures (Example 2.13). Importantly, the author also refers to set membership directly, as a means of bridging (Example 2.14). Examples 2.9–2.14 are taken directly from Clark (1975).

(2.9) I looked into *the room*. **The ceiling** was very high.

(2.10) I walked into *the room*. **The windows** looked out to the bay.

(2.11) I walked into *the room*. **The chandeliers** sparkled brightly.

(2.12) *John was murdered* yesterday. **The murderer** got away.

(2.13) *John fell.* **What he did** was trip on a rock.

(2.14) I met *two doctors yesterday* yesterday. **The tall one** told me a story.

Prince (1981) discusses *inferrable* discourse entities, which can be inferred via logical or plausible reasoning from previously evoked entities. These inferrables include both set members and subsets. A special type of inferrable is mentioned, a *containing inferrable*, in which the inference is contained within the NP. For example, in the NP *one of these eggs*, there is an inferred set membership relationship between *one of these eggs* and *these eggs*. This inferrable is likely to overlap regularly with intrasentential entity instantiations.

Other theoretical linguistic literature to discuss bridging includes Asher and Lascarides (1998), who suggest that discourse relations[9] can be helpful in resolving bridging references.

A number of corpus studies have explored bridging anaphora. Fraurud (1990) studies *definite* NPs — those NPs that begin with the definite article *'the'* and commonly refer to previously introduced entities. They organised definite NPs into two simple categories; first mentions and subsequent mentions, and found that a surprisingly large number (61%) of definite NPs were first mentions, rather than anaphoric references, in their Swedish corpus.

[9] See Chapter 5 for a further, fuller discussion of discourse relations.

Prince (1992) also examines a text for information status. Rather than the two categories proposed by Fraurud (1990), she devises three; *discourse/hearer new, discourse/ hearer old* and *inferrable*.

We also note Recasens et al. (2010), in which the authors develop a typology of *near-identity* coreference relationships. These are similar to bridging references, in that they require inference, but are distinct in that the linked entities cannot be said to be non-identical. Their scheme includes part-of relationships and largely overlapping sets. Set membership relations, however, are not tackled.

2.3.3 Bridging corpora

A number of efforts have been made to construct corpora labelled with information status, which include bridging anaphora.

Poesio and Vieira (1998) investigated the practicality of annotating definite NPs. Their annotations were restricted to *definite descriptions* (DDs), which excludes pronouns and possessive descriptions. They experimented with two schemes, based on Hawkins (1978) and Prince (1992) respectively. However, the annotators agreement was marginal for both schemes.

The GNOME corpus (Poesio, 2003) attempts to address these agreement difficulties by limiting the categories of bridging references annotated to set membership, subsets, and generalised possession, which includes part-of relations. However, agreement is still poor; only 22% of bridging references were annotated identically by both annotators. The resulting corpus is relatively small, containing 581 bridging references.

Nissim et al. (2004) annotate a corpus of dialogues for *information status*. The information status of an entity represents whether it is *new* to the reader, *old* because it is coreferent to a prior mention, or can be *mediated* from old information, often by bridging. Their mediated entities comprise 9 subcategories including part-of and set membership mediation. However, their set membership category consists of set membership, subsets and co-set-members (two members of a common set), and has a number of restrictions, including that it can only be used if anaphor and antecedent have either the same head, a synonymous head or are part of a hyponym relation encoded in WordNet. Their agreement study, over 3 dialogues, resulted in a Kappa of 0.845 for the 3 main types, falling to 0.788 over the subtypes, a large improvement over prior corpora. We note, however, that agreement on part-of and set membership subtypes was considerably lower, at 0.594 and 0.696 respectively.

Following the agreement study, a total of 147 dialogues were annotated, for a corpus

of 43,358 sentences with 69,004 NPs, 23,816 of which were mediated. The resulting corpus was the first substantially sized bridging corpus.

Markert et al. (2012) annotate a portion of the OntoNotes corpus[10] for fine-grained information status. They modify Nissim et al. (2004)'s scheme, with mediated entities falling into one of six subtypes, including the four below which differ from Nissim et al. (2004)'s scheme:

Bridging. Bridging references. No distinction is currently made between the relationships used to bridge the anaphor and antecedent, such as part-of, set membership, etc, but this is planned in the near future. No restriction is made on the type of relationship used to bridge the anaphor and antecedent.

Comparative. Comparative anaphora, as described in Section 2.3.1.

Knowledge. Entities generally known to the hearer, such as place names.

Syntactic. Entities syntactically linked to their antecedents by a possessive relation.

Agreement is tested over 26 texts, and is generally good, with Kappa in the range 0.773 – 0.801. Agreement for the bridging category is lower, however, with Kappa in the range 0.606 – 0.708. Subsequently, a total of 50 texts were annotated in this manner, resulting in 3,708 mediated entities, 662 of which are bridged. The lack of further distinctions in the bridging category means it is not possible to establish how many of the bridging anaphors discovered are set members or subsets.

Additionally, we note that a variety of research has involved the creation of bridging/IS corpora in languages other than English including, German (Riester et al., 2010), French (Gardent et al., 2003), Dutch (Hendrickx et al., 2008) and Czech (Nedoluzhko et al., 2009).

2.3.4 Bridging anaphora resolution algorithms

A number of approaches have been detailed for the resolution of bridging anaphora.

Early, rule-based approaches include Markert et al. (1996), who use Centering-based rules (Grosz et al., 1995) and a terminological knowledge base to classify bridging anaphora, and Poesio et al. (1997) and Vieira and Poesio (2000) who both use a large number of hand-crafted linguistic rules to classify DDs using the corpus from Poesio and Vieira (1998).

[10]The English OntoNotes corpus is itself a portion of the Penn Treebank WSJ corpus.

Initial machine learning approaches focused solely on meronymy-based bridging, and attempted to find the correct antecedent for a known anaphor. Markert et al. (2003) used two Hearst-like patterns coupled with web queries to choose the most likely antecedent of a bridging reference in an unsupervised manner. However, their test set is very small, comprising of only 12 examples. Poesio et al. (2004) combine lexical distance features, calculated using WordNet and Google, and Centering-based salience features in a supervised machine learning model. They use the previously described GNOME corpus, which is also relatively small with only 153 positive instances. Their best F-Score on realistic data (as opposed to artificially balanced test sets) is 0.5.

Other work has focused on learning the information status of an entity, rather than identifying its antecedent. Nissim (2006) experiment with learning the coarse-grained IS of an entity (i.e new, mediated or old). They use the Switchboard corpus, and their feature set comprises 7 features, including the number of previous mentions of the entity, the presence of a partial previous mention, the determiner of the NP, and the grammatical role of the NP. Their method scores an accuracy of 79.5%, significantly better than a hand-crafted rule-based baseline., Their classifier performs very well on old entities (F-Score = 0.928), but quite poorly on new entities (F-Score = 0.320).

Rahman and Ng (2011) expand on Nissim (2006) by introducing a lexical feature which attempts to estimate whether an entity is 'generally known', and a tree kernel. They reduce the error rate over Nissim (2006) by 2.7%, and increase performance on new entities to F = 0.465.

More relevant to our work is the learning of fine-grained IS, which involves learning subtypes of the mediated category, including set membership. Rahman and Ng (2012) again use the Switchboard corpus, which does include a restricted version of set membership, as discussed in Section 2.3.3. Using a feature set based on unigrams, markables and a number of binary features based on hand-coded rules their learner scores an impressive accuracy of 86.4% using gold-standard coreference data. On set mediation they achieve an F-Score of 89.1 over 1771 instances. The restrictions on set mediation, such as requiring heads to be identical, synonymous or related in WordNet (see Section 2.3.3, pg 47) make the examples reasonably straightforward to classify using WordNet look ups.

Markert et al. (2012) learn fine-grained IS on a portion of OntoNotes corpus (see Section 2.3.3). They couple Nissim (2006)'s and Rahman and Ng (2011)'s features with some additional local features, and implement a collective learning model, with links between instances based upon syntactic parent-child and precedence relations. Their algorithm achieves an overall accuracy of 76.8 using all features and links, but on the fine-grained bridging class, which includes set membership, they only achieve an F-Score of 18.9.

2.4 Conclusion

Our problem is novel, and therefore there is no prior literature that deals specifically with entity instantiations. In this Chapter we have considered three research problems which closely relate to our problem; Information Extraction, Context-Independent Relation Extraction and Bridging Anaphora.

We discussed in detail the IE sub-problem of relation extraction, which also considers binary relationships between noun phrases, but does not consider set membership and subset relationships. It also differs by limiting participating noun phrases to entities. However, there are several parallels, and in our machine learning experiments we take inspiration from a number of RE methods, including the use of tree kernels (Zelenko et al., 2003; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; Zhao and Grishman, 2005; Zhang et al., 2006; Bunescu and Mooney, 2006; Zhou et al., 2007; Jiang and Zhai, 2007; Swampillai and Stevenson, 2011) and background knowledge (Chan and Roth, 2010; Sun et al., 2011).

Context-Independent Relation Extraction is less closely related to our problem, but also considers binary relationships between noun phrases. The output of these algorithms are a potential feature in our machine learning experiments.

The problem of resolving Bridging Anaphora overlaps with entity instantiations; not all bridging anaphora are entity instantiations, and conversely neither are all entity instantiations bridged. However, we reviewed bridging literature that considered set membership, and summarised some notable attempts at automatically resolving bridging references.

In the next chapter we discuss in detail our annotation of entity instantiations, including our annotation scheme, agreement study and gold standard corpus.

Chapter 3

Creating a Corpus of Entity Instantiations

3.1 Motivation

Having, in the previous Chapter, identified the phenomenon of entity instantiations, and established that the phenomenon has not been tackled prior to our examination of it, we then used human annotators to manually identify examples. This required us to tightly specify and elucidate our definition of what constitutes an entity instantiation. The full detail of this can be seen in Sections 3.2 and 3.3.

In terms of our approach, we took inspiration from the Recognising Textual Entailment (RTE) task (Dagan et al., 2006). In RTE, the challenge is to automatically ascertain whether a text (T) *entails* a hypothesis (H). Table 3.1 shows several examples of positive and negative cases of textual entailment, reproduced from Dagan et al. (2006).

Rather than framing the problem as an issue of logical implicature, they regard RTE as an applied, empirical task:

> "We say that T entails H if, typically, a human reading T would infer that H is most likely true." (Dagan et al., 2006, pp. 178)

We also consider our problem in this manner. We are interested in the phenomena from the perspective of a human reading the text, and do not apply strict logical rules

T	*H*	*T entails H*
Norways most famous painting, "The Scream" by Edvard Munch, was recovered Saturday, almost three months after it was stolen from an Oslo museum.	Edvard Munch painted "The Scream".	True
Most Americans are familiar with the Food Guide Pyramid — but a lot of people dont understand how to use it and the government claims that the proof is that two out of three Americans are fat.	Two out of three Americans are fat.	True
The SPD got just 21.5% of the vote in the European Parliament elections, while the conservative opposition parties polled 44.5%.	The SPD is defeated by the opposition parties.	True
Reagan attended a ceremony in Washington to commemorate the landings in Normandy	Washington is located in Normandy	False
Time Warner is the world's largest media and Internet company.	Time Warner is the world's largest company	False
Bush returned to the White House late Saturday while his running mate was off campaigning in the West.	Bush left the White House.	False

Table 3.1: Positive and negative examples of textual entailment from Dagan et al. (2006).

for identifying entity instantiations, instead taking an applied approach. We discuss the implications of this approach in Section 3.4.

3.2 Intuitive Definition and Examples

In this Section, we provide a general definition of an entity instantiation, and demonstrate the extent and variety of the phenomenon, by means of a series of examples.

3.2.1 Instantiation definition

An entity instantiation is a relationship between a *set* of entities and either a *member* of that set, or a *subset* of that set.

Our primary principle for identifying instantiations is that we require all statements that apply to the set to also hold true for the member/subset. We exclude cases where the statements could not apply to an individual member of the set, but instead describe the nature of the set, such as *'is large'* or *'contains five members'*. This principle also applies to any intrinsic properties that the set might have that are applicable to individual members. A simple method of checking that this rule holds is to rephrase the potential instantiation in the following format:

> {Set Member/Subset} is a/is one of/are {Set} that {statements made about the set}

For instance, we might rephrase the instantiation from Example 3.1[1] as *'Mr Packwood is one of the two lawmakers that sparred in a highly personal fashion'*.

(3.1) a. **The two lawmakers** sparred in a highly personal fashion, violating usual Senate decorum.

b. Their tone was good-natured, with *Mr. Packwood* saying he intended to offer the proposal again and again on future legislation and *Sen. Mitchell* saying he intended to use procedural means to block it again and again.

The presence of an entity instantiation is highly context dependent and requires careful consideration of prior mentions of both the set and member/subset. In Example 3.2, one has to look back 2 sentences to establish that *'they'* is coreferent with *'the Montreal Protocol's legions of supporters'*, and a set from which *'Peter Teagan, a specialist in*

[1] All examples in this Chapter are either taken from the Penn Treebank Wall Street Journal corpus, or created for the purpose of illustrating a particular point.

heat transfer' may be drawn. In Example 3.3, we need the knowledge that Mr. Mason is Jewish, from the first sentence of the extract, to establish the instantiation in the final sentence.

(3.2) But even though by some estimates it might cost the world as much as $100 billion between now and the year 2000 to convert to other coolants, foaming agents and solvents and to redesign equipment for these less efficient substitutes, the Montreal Protocol's legions of supporters say it is worth it. They insist that CFCs are damaging the earth's stratospheric ozone layer, which screens out some of the sun's ultraviolet rays. Hence, as **they** see it, if something isn't done earthlings will become ever more subject to sunburn and skin cancer.

Peter Teagan, a specialist in heat transfer, is running a project at Arthur D. Little Inc., of Cambridge, Mass., to find alternative technologies that will allow industry to eliminate CFCs.

(3.3) . . . Or so it must seem to Jackie Mason, the veteran Jewish comedian appearing in a new ABC sitcom airing on Tuesday nights (9:30-10 p.m. EDT). Not only is Mr. Mason the star of "Chicken Soup," he's also the inheritor of a comedic tradition dating back to "Duck Soup," and he's currently a man in hot water.

Here, in neutral language, is the gist of Mr. Mason's remarks, quoted first in the Village Voice while he was a paid spokesman for the Rudolph Giuliani mayoral campaign, and then in Newsweek after he and the campaign parted company. Mr. Mason said that many Jewish voters feel guilty toward blacks, so they support black candidates uncritically. He said that many black voters feel bitter about racial discrimination, so they, too, support black candidates uncritically. *He* said that **Jews** have contributed more to black causes over the years than vice versa.

In an example such as Example 3.4, where no statements are made about the set in the text, we can rely on our knowledge of the intrinsic properties of the two NPs — that *'Canadian Indians'* describes all Indians living in Canada, and that *'Inuit and Cree peoples living in [. . .] northeastern Canada'* are subtypes of Indians living in a region of Canada — to deduce that *'Representatives of the Inuit and Cree peoples [. . .] are Canadian Indians'* holds.

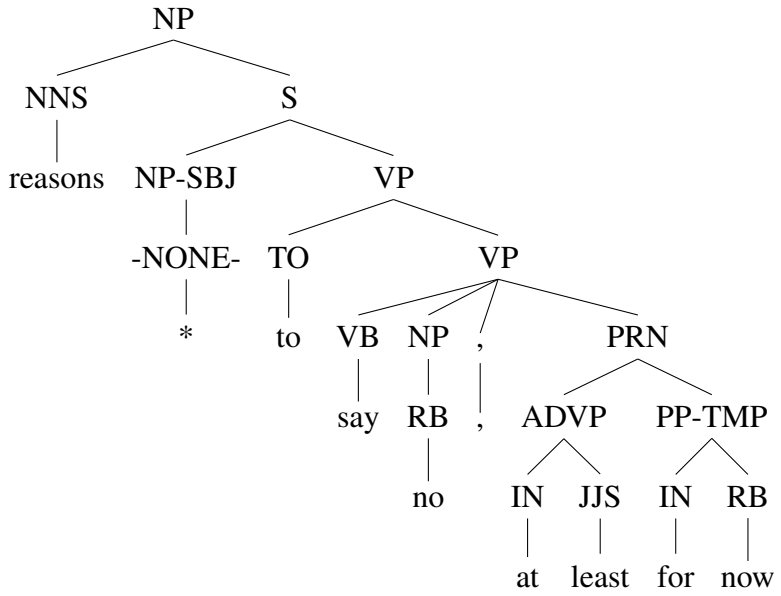


Figure 3.1: A constituency parse tree representation of the set noun phrase from Example 3.6.

(3.4) a. **Canadian Indians** are taking five countries to court in a bid to stop low military flights over their homes, the Dutch Defense Ministry said.

b. *Representatives of the Inuit and Cree peoples living in Quebec and Labrador in northeastern Canada* told the ministry of the planned action at a meeting, a ministry spokesman said.

3.2.2 Variety in entity instantiations

Variety in internal structure. There is great variety in the internal structure of NPs involved in instantiations, as well as the grammatical constructions in which they play a part. Example 3.1 shows a set member entity instantiation between an NP headed by a plural noun, and a named entity. Example 3.5 is similar, but in this case the set member is modified by an apposition — *'an analyst with Drexel Burnham Lambert'*. In Example 3.6, neither set nor set member are named entities, and the set is a complex plural noun phrase which is made up of several constituents, a parse of which is shown in Figure 3.1.

(3.5) a. But **other analysts** said that having Mr. Phillips succeed Mr. Roman would make for a smooth transition.

b. "Graham Phillips has been there a long time, knows the culture well, is aggressive, and apparently gets along well with Mr. Sorrell", said *Andrew Wallach, an analyst with Drexel Burnham Lambert.*

(3.6) a. And Democrats, who are under increasing pressure from their leaders to reject the gains-tax cut, are finding **reasons to say no, at least for now.**

b. *A major reason* is that they believe the Packwood-Roth plan would lose buckets of revenue over the long run.

Variety in ordering. Sets may also occur after the member or subset in a text. Example 3.7 shows a situation where the set NP is a conjunction of smaller singular NPs — *Italy, Spain, Turkey, Greece and the Soviet Union* — and the member is introduced first. The set NP in Example 3.8 is constructed from the conjunction of a singular NP and a plural NP.

(3.7) a. Japan has been testing imported food from Europe since the April 1986 Chernobyl accident in *the Soviet Union*, the spokesman said.

b. Since then, the ministry has announced 50 bans on food imports from European countries, including **Italy, Spain, Turkey, Greece and the Soviet Union**.

(3.8) a. According to West German government sources, **Mr. Honecker and several senior Politburo members** fought over the last week to delay any decisions about a leadership change.

b. The removal of *Mr. Honecker* was apparently the result of bitter infighting within the top ranks of the Communist party.

Overlap with other phenomena. Other linguistic phenomena can make the identification of instantiations more complex. In Example 3.9, the set member is the pronoun *'I'*. This pronoun is coreferent with *'President Bush'*, and this coreference link is required to comprehend the instantiation. The set in Example 3.10 mentions *'Beijing'*, a location. However, in this context it is a metonymic reference to the Chinese government, and therefore *'China'* can be drawn from it..

Sets can often be very general — in Example 3.11, we see a very general set (*'entrepreneurs'*), with a vaguely quantified subset (*'some entrepreneurs'*).

(3.9) a. But **U.S. officials** have strong doubts that he is a reformer.

b. President Bush told reporters: "Whether that the leadership change reflects a change in East-West relations, *I* don't think so.

(3.10) a. In a sign of easing tension between **Beijing and Hong Kong**, China said it will again take back illegal immigrants caught crossing into the British colony.

b. *China* had refused to repatriate citizens who sneaked into Hong Kong illegally since early this month, when the colony allowed a dissident Chinese swimmer to flee to the U.S.

(3.11) a. Whatever the monetary crime losses, they may not be nearly as important to **entrepreneurs** as the risk of personal injury.

b. After repeated gun robberies, *some entrepreneurs* may give up a business out of fear for their lives.

Intrasentential nesting. When we consider the possibility of intrasentential instantiations, the variety of configurations of instantiations further increases. Example 3.12 shows an example where a set member is nested within the conjunction that forms the set. Example 3.13 shows another example where the set member is nested in the set, but this time as a subtree of the prepositional phrase that complements the set NP. Example 3.14 shows a different sort of nesting — the set is nested within the set member. There are also many intrasentential instantiations where the participant NPs do not overlap, such as Examples 3.15 and 3.16.

(3.12) So if anything happened to me, I'd want to leave behind enough so that my 33-year-old husband would be able to pay off ***the mortgage*** **and some other debts** (though not, I admit, enough to put any potential second wife in the lap of luxury).

(3.13) Over the past nine months, **several firms, including discount broker Charles Schwab & Co. and *Sears, Roebuck & Co. 's Dean Witter Reynolds Inc. unit***, have attacked program trading as a major market evil.

(3.14) When he is presented with a poster celebrating the organization's 20th anniversary, he recognizes a photograph of *one of* ***the founders*** and recalls time spent together in Camden.

(3.15) Before **the two parties** resumed talks last week, *De Beers* offered 17% and the union wanted 37.6%.

(3.16) **Banking stocks** were the major gainers Monday amid hope that interest rates have peaked, as *Deutsche Bank and Dresdner Bank* added 4 marks each to 664 marks ($357) and 326 marks, respectively.

Variety in distribution. All examples in this Section so far have highlighted a single instantiation between a pair of sentences or within a single sentence. Often, many instantiations occur between a pair of sentences. Example 3.17 shows a pair of sentences with a large number of instantiations present. A full list of instantiations present is shown in Table 3.2

(3.17) a. The survey found that nearly half of Hong Kong consumers espouse what it identified as materialistic values, compared with about one-third in Japan and the U.S.

b. The study by the Backer Spielvogel Bates ad agency also found that the colony's consumers feel more pressured than those in any of the other surveyed markets, which include the U.S. and Japan.

3.3 Exact Definition and Annotation Guidelines

3.3.1 Annotation restrictions

We impose two restrictions on our annotations.

Firstly, both participants in an entity instantiation are restricted to noun phrases, and both must be mentions, as defined in Section 3.3.2. Set and subset NPs must be plural, and set member NPs must be singular. We enforce the restriction regarding plural sets and subsets to avoid marking meronymy or other relationships such as employment or location. This restriction also helps to exclude the possibility of the set being a mention that is in some way more than a simple grouping of entities. For example, although *'the EU'* might refer to a group of countries, it has properties (e.g. a council, a parliament, a court) that make it more than simply a sum of its parts. Similarly, *'Manchester United'* is more than 11 footballers playing on the field.

Secondly, we restrict the scope of our annotation, by allowing intersentential annotation to occur only between adjacent sentences. We restrict this on the basis that marking entity instantiations between all available NPs leads to complex annotation, making it very difficult for an annotator to track entity instantiations and leading to errors. Clearly, this restriction leads to the omission of some entity instantiations, and the full impact of this restriction is discussed in Section 3.7.

Both restrictions are enforced automatically, as we filter the NPs our annotators are shown. Our algorithm for classifying NPs as singular or plural is described in Section 3.5.3.

Instantiation type	**Set NP**	**Set Member/Subset NP**
Intersentential, subset	any of the other surveyed markets, which include the U.S. and Japan	Japan and the U.S.
	the other surveyed markets, which include the U.S. and Japan	Japan and the U.S.
Intersentential, set member	any of the other surveyed markets, which include the U.S. and Japan	Japan
	any of the other surveyed markets, which include the U.S. and Japan	the U.S.
	the other surveyed markets, which include the U.S. and Japan	Japan
	the other surveyed markets, which include the U.S. and Japan	the U.S.
	the U.S. and Japan	Japan
	the U.S. and Japan	the U.S.
	Japan and the U.S.	the U.S.
	Japan and the U.S.	Japan
Intrasentential, subset	any of the other surveyed markets , which include the U.S. and Japan	the U.S. and Japan
	the other surveyed markets , which include the U.S. and Japan	the U.S. and Japan
Intrasentential, set member	any of the other surveyed markets , which include the U.S. and Japan	the U.S.
	any of the other surveyed markets , which include the U.S. and Japan	Japan
	the other surveyed markets , which include the U.S. and Japan	the U.S.
	the other surveyed markets , which include the U.S. and Japan	Japan
	the U.S. and Japan	the U.S.
	the U.S. and Japan	Japan
	Japan and the U.S.	Japan
	Japan and the U.S.	the U.S.

Table 3.2: Instantiations present in Example 3.17.

3.3.2 Definition of a mention

We consider all NPs in the text to be mentions, unless they meet one of the 3 definitions described below.

3.3.2.1 Generic pronouns

We consider generic uses of *'we'* and *'you'* as non-mentions, along with references to the reader or audience of a text. Examples 3.18, 3.19 and 3.20 are all considered non-mentions.

Our justification is that these references convey very little information about the entities considered in the document and instead refer to an abstract notion of a reader, or some entirely undefinable set. An undefinable set is impossible to draw an entity instantiation from. Consider a case of a generic *'we'*, such as that in Example 3.19. Were we to consider this use to be a mention, and therefore a set from which we can draw an instantiation, a variety of complicated, and unanswerable questions arise:

- Does this use of *'we'* refer to all human beings alive today?
- Or an educated subset that are aware of the issues surrounding non-violent civil disobedience?
- Or those reading the article?
- Or is it simply a throwaway reference that could have easily been written as 'as it is known today'

Not considering these as mentions removes these complexities, and reduces the number of instantiations that convey little information.

(3.18) *You* know, it's really tricky to figure out where to begin with this mess.

(3.19) Maybe he didn't start it, but Mohandas Gandhi certainly provided a recognizable beginning to non-violent civil disobedience as *we* know it today.

(3.20) So *dear reader*, we advise that you don't rush into your investments.

3.3.2.2 Idiomatic mentions

Idiomatic NPs that have no literal meaning are considered not a mention. Examples 3.21, 3.22 and 3.23 show examples of idiomatic NPs which are not mentions. Example 3.24 is a mention — the MP's eyes exist.

(3.21) How many senators does it take to change *a light bulb*?

(3.22) The chairman has *an axe* to grind with the regulators.

(3.23) On *the ropes*.

(3.24) The MP, known for his *eagle eyes*, spotted the error immediately.

However, metaphoric mentions can occur as part of a recurring theme, with the potential for instantiations to occur, such as in Example 3.25. These are considered mentions.

(3.25) a. Bob Dylan asked '**How many roads** must a man walk down?'

b. Well, *one road* is particularly well walked.

3.3.2.3 Non-referential 'it'

Non-referential uses of *'it'*, such as those in Examples 3.26 and 3.27 are not considered a mention.

(3.26) *It* seems that this weather is here to stay for the week.

(3.27) *It* is said that only fools rush in.

Their non-referentiality means that they can never participate in an instantiation, and are therefore excluded.

3.3.3 Specific annotation rules and special cases

In this Section we detail cases that, whilst covered by our general definition of the problem (Section 3.2), are worthy of further clarification.

3.3.3.1 Generic mentions

In Example 3.28, *'the planner'* mentioned in the second sentence refers to a notional planner, rather than any actual member of the set of **Planners**, and therefore should not be marked as an instantiation.

(3.28) a. **Planners** often have to make difficult decisions.

b. The issue: does *the planner* have the required qualifications to make them.

3.3.3.2 Indefinite pronouns

Indefinite pronouns such as *either*, *any* or *each* which could refer to any member of a set, but do not refer to a specific member or subset, are not marked as instantiations. Examples 3.29 and 3.30 show examples where the instantiation should not be marked.

(3.29) a. **John Smith and John Doe** are competing for the contract.

b. *Either* could clinch it.

(3.30) a. **All three companies** are struggling.

b. *Any* might go bust before the year is out.

However, indefinite pronouns which do refer to a subset or a single member of a set, such as those in Examples 3.31 and 3.32, are instantiations.

(3.31) a. **Seven companies** are bidding.

b. *Most* are US-based.

(3.32) a. **All three companies** are struggling.

b. *One, which I visited earlier this month,* might go bust before the year is out.

3.3.3.3 Negated mentions

If a set, set member or subset is negated, such as those in Examples 3.33 and 3.34, it cannot participate in an instantiation.

(3.33) a. **Neither the US nor the UK** have managed to keep their debt under control.

b. *The UK*'s debt has risen by 10% this year alone.

(3.34) a. *John*, Mary and James just sat and watched.

b. **Not one of them** dared intervene.

3.3.3.4 Members implicitly excluded from sets

Occasionally the context of the candidate instantiation can implicitly exclude a member or subset from participating. In Example 3.35, the set of **Democrats** excludes *Senator Smith*, as he is certainly going to vote for the measure.

(3.35) a. **Democrats** are reluctant to break ranks and vote against the measure.

b. *Senator Smith, their leader in the senate,* has staked his reputation on the bill, and those voting against would be betraying his confidence.

3.3.3.5 Metonymic mentions

If a candidate set member or subset is a metonymic reference, then an instantiation should only be marked if the set is of the concept the metonymy represents rather than the literal reading of the word. This often occurs in the WSJ corpus with regards to shares in a company being referred to by the name of the company itself.

Example 3.36 shows a situation in which an instantiation should not be marked — *'Hollywood'* is referring to the industry rather than the district of L.A. Conversely, an instantiation should be marked in Example 3.37, as *'Westminster'* refers to the UK Parliament, rather than the area in this context.

(3.36) a. *Hollywood* has made countless films about L.A.

b. **All of the districts in the city** have starred at some point.

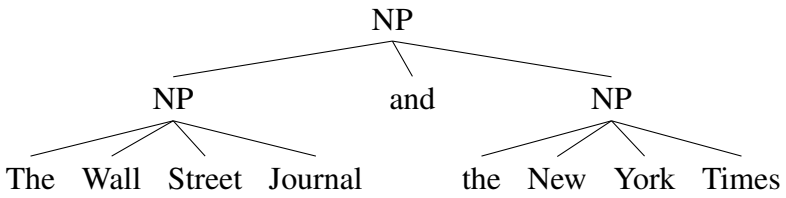
(3.37) a. **Parliaments around the EU** were ratifying the treaty this week.

b. *Westminster* passed it on Tuesday.

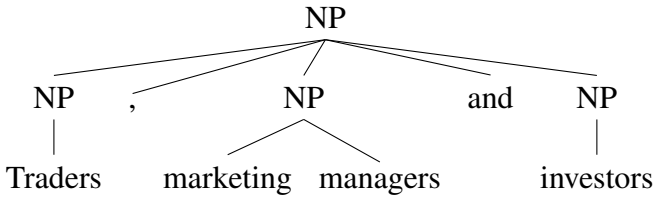
3.3.3.6 Co-ordinations

A co-ordination is considered a set. Examples 3.38, 3.39 and 3.40 show three sets constructed from co-ordinations; one from two singular NPs, one from three plurals and one from a mixture of plural and singular NPs.

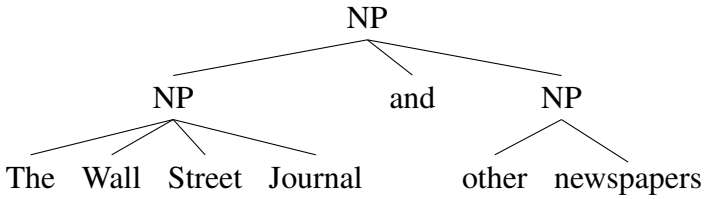
(3.38)



(3.39)



(3.40)



The only instantiations that can be drawn from Example 3.38 are NPs coreferent with *The Wall Street Journal* and *The New York Times*.

Any instantiation of the child plural NPs in Example 3.39 must also be marked as an instantiation of the whole phrase. In other words, *John Smith, an NYSE trader* would be a set member of both (NP Traders) and (NP (NP Traders), (NP marketing managers) and (NP investors)).

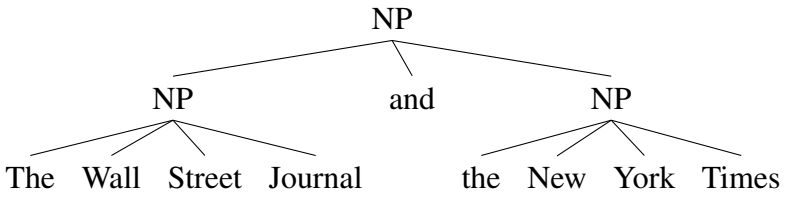
Any instantiation of the child plural NP (NP other newspapers) in Example 3.40 must also be marked as instantiation of the whole NP.

3.3.3.7 Nested mentions

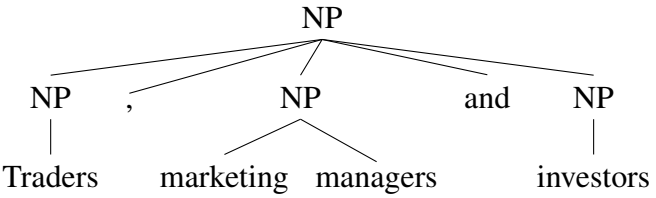
One feature of intrasentential annotation is the possibility of nested mentions. Several types of nested mentions are common.

Example 3.41 shows a simple co-ordination that contains two singular NPs. Both singular NPs should be marked as set members of the larger co-ordination NP. Similarly, in Example 3.42, the child plural and singular NPs should be marked as a subset and a set member of the larger NP.

(3.41)

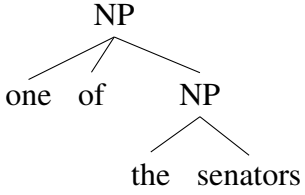


(3.42)

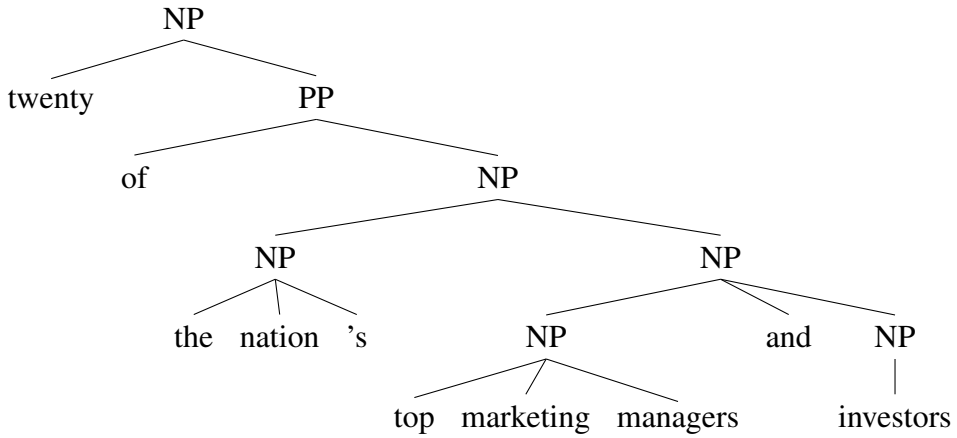


In Examples 3.43 and 3.44 the set is nested within the NP describing the set member or subset. These instantiations should be marked as normal.

(3.43)



(3.44)



3.4 Potential Difficulties and Borderline Cases

Whilst our decision to take an applied approach allows for intuitive annotation, there are some borderline cases where the lack of strict logical rules can be a drawback.

The plural NPs which act as sets in our corpus fall into 4 rough categories:

Extensionally defined. The NP is made up of an explicit list of elements, from which set members and subsets can be drawn. Examples 3.45 and 3.46 show extensionally defined sets.

Clearly intensionally defined. The NP describes a finite set, whose members are easily identifiable. Examples 3.47 and 3.48 show clearly defined intensional examples.

Vaguely intensionally defined. The NP describes a set whose members are non-finite, or difficult to exactly establish.

Generic. The NP describes a class of objects, rather than a specific set.

For those NPs which are either extensionally defined or are clearly intensional, set members are easy to identify. The other two categories cause more difficulties. Not knowing the members in a vaguely intensionally defined set makes it difficult judging whether the relationship between NPs is a subset, coreference or set overlap. In Example 3.49, for instance, it is difficult to know for certain whether *'175'* and *'136'* are subsets of *'The 189 Democrats who supported the override yesterday'*, though it may be assumed to be the case. A similar situation exists with the drift-net vessels in Example 3.50.

(3.45) a. However, the disclosure of the guidelines, first reported last night by NBC News, is already being interpreted on *Capitol Hill* as an unfair effort to pressure Congress.

b. It has reopened the bitter wrangling between **the White House and Congress** over who is responsible for the failure to oust Mr. Noriega and, more broadly, for difficulties in carrying out covert activities abroad.

(3.46) a. Although the proposal, authored by **Mr. Packwood and Sen. William Roth (R., Del.)**, appears to have general backing by Republicans, their votes aren't sufficient to pass it.

b. Their tone was good-natured, with *Mr. Packwood* saying he intended to offer the proposal again and again on future legislation and Sen. Mitchell saying he intended to use procedural means to block it again and again.

(3.47) a. To the extent that the primary duty of personal staff involves local benefit-seeking, this indicates that political philosophy leads **congressional Republicans** to pay less attention to narrow constituent concerns.

b. First, economists James Bennett and Thomas DiLorenzo find that *GOP senators* turn back roughly 10% more of their allocated personal staff budgets than Democrats do.

(3.48) **Banking stocks** were the major gainers Monday amid hope that interest rates have peaked, as *Deutsche Bank and Dresdner Bank* added 4 marks each to 664 marks and 326 marks, respectively.

(3.49) **The 189 Democrats who supported the override yesterday** compare with *175 who initially backed the rape-and-incest exemption two weeks ago* and *136 last year*.

(3.50) Earlier this year, Japan said it would cut the number of **its drift-net vessels in the South Pacific** by two-thirds, or down to *20* on a similar vote.

In our annotation scheme, we make no distinction between those plural NPs which represent sets and those which represent generics, and allow instantiations to be drawn from both. This leads to annotation that is more akin to hyponymy than set membership or subset relationships, such as in Example 3.51.

(3.51) a. A customs official said the arrests followed a "Snake Day" at Utrecht University in the Netherlands, an event used by some collectors as an opportunity to obtain **rare snakes**.

b. British customs officers said they'd arrested eight men sneaking *111 rare snakes* into Britain — including one man who strapped a pair of boa constrictors under his armpits.

Despite these problems, we still achieved substantial agreement (see Section 3.6.1). This is likely due to the genre of the texts involved; the financial-based newswire texts annotated tend to include many sets, subsets and members which are concrete, such as companies, countries and people. Applying this scheme to a genre of texts that contains more generics and less straightforwardly defined NPs, for example a philosophy text, could lead to a more problematic annotation. One possible way to improve agreement would be to introduce a layer of annotation that identified generic NPs, such as that employed by Reiter and Frank (2010), and prevent these generic NPs from participating in instantiations.

3.5 The Annotation Design and Process

This Section details the actual implementation of the annotation tool, and the pre-processing steps required to present the tool with relevant NPs classified into singular or plural.

All code was written in Python. Tree structures were dealt with using NLTK's nltk.tree package (Bird et al., 2009), and graphical user interface programming was done using wxPython[2].

3.5.1 Choice of texts to annotate

The texts chosen to be annotated for entity instantiations were newspaper texts drawn from the Wall Street Journal (WSJ) corpus of the Penn Treebank (PTB) (Marcus et al., 1993). We chose this source due to the many layers of annotation that already exist for these texts. These include the syntactic parses that form the PTB, discourse relations (the Penn Discourse Treebank, Prasad et al. (2008)), verbal propositions and their arguments (PropBank, Palmer et al. (2005)), the arguments of noun-phrases (NomBank, Meyers et al. (2004)), and coreference (OntoNotes, Weischedel et al. (2011)). Using these texts allowed us the possibility of leveraging these resources in future attempts at automatic

[2]wxPython is available from `http://wxpython.org/`.

entity instantiation identification, as well as further exploring the relationship between these phenomena and our annotations.

The choice of these texts is not without its drawbacks, however. At the time of writing, the texts of the WSJ corpus are 23 years old and cover topics including the politics of the German Democratic Republic and General Noriega's reign as dictator of Panama. The dated nature of the texts may provide an obstacle to efficient and accurate annotation. Also, the financial focus of the WSJ means that several of the texts contain large amounts of fiscal jargon, which may be hard to understand for an unfamiliar annotator.

We annotated full texts for intrasentential and sentence-adjacent intersentential instantiations, on the basis that establishing relationships between entities throughout a whole text would be easier for an annotator than considering sentences or sentence pairs in isolation.

3.5.2 Pre-processing of noun phrases

The first pre-processing task undertaken was the extraction of NPs and the removal of NPs that cannot be mentions, and therefore should not be presented to the annotator.

The noun-phrases were extracted from the gold standard parses of the PTB. We excluded the following types of noun-phrases at this pre-processing step, based on syntactic parse data only:

NP-ADV Adverbial NPs, such as '50 dollars *(NP-ADV a share)*'. These describe rates such as 'a share', 'per hour' or 'each year', rather than any sort of concrete notion that might participate in an instantiation.

NP-PRD Predicate NPs, such as 'Mr Vinken is *(NP-PRD chairman of Elsevier)*'. We are interested in the head of these phrases — Mr Vinken in this case — rather than the predicates which are non-mentions.

Child NPs of ADVPs For example '(ADVP Many *(NP years)* (IN ago)'. These are exclude for the same reasons as NP-ADVs.

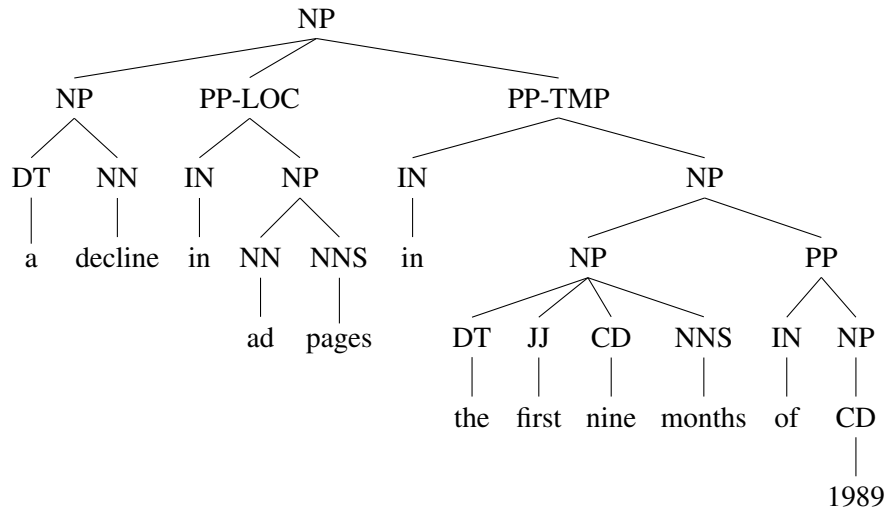
Existential 'There' phrases NPs with a single existential 'there' as a child are excluded. For example '*(NP (EX There))* 's no question he's the best'.

Null NPs In the PTB annotation, null elements are included to mark phenomena such as ellipsed material. These syntactic constructs are removed.

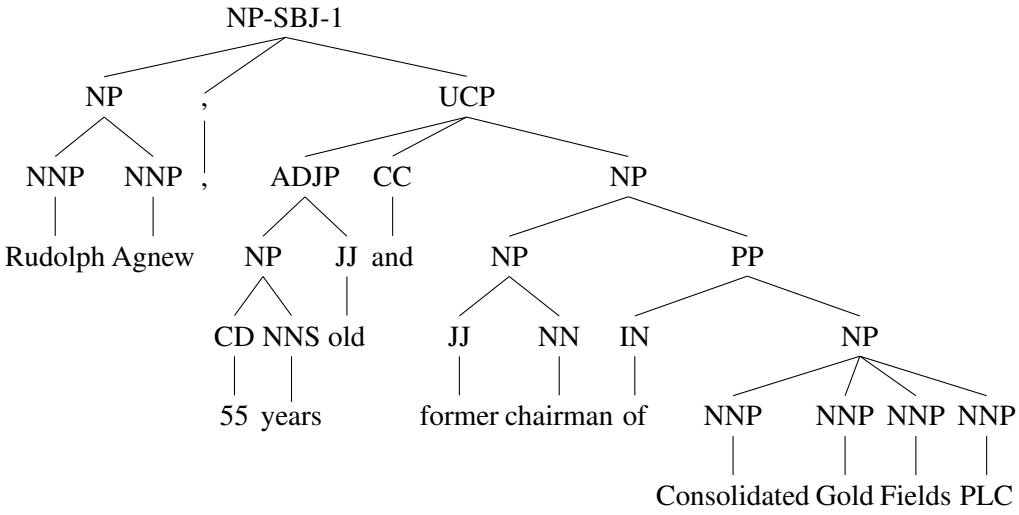
At the next step we used gold standard dependency parse data, generated using the LTH Constituent-to-Dependency Conversion Tool for Penn-style Treebanks (Johansson

and Nugues, 2007), to exclude further categories of NPs. In general, we only include the largest NP that describes a certain concept, and exclude appositions, conjunctions of appositions, and NPs that are the head of a larger NP modified by prepositional phrases. Example 3.52 shows a complex NP with nested child NPs, and Table 3.3 shows which NPs are excluded by this process. For Example 3.53, the excluded and included NPs are shown in Table 3.4

(3.52)



(3.53)



3.5.3 Classification of noun phrases into singular or plural

We classified NPs as either singular or plural. This was necessary, as described in Section 3.3.1, in order to avoid marking meronymy or other relationships such as employment or location as entity instantiations. We applied the following general rules in this process:

1. If the NP is a named entity, or the headword of the NP is a named entity → singular.

Noun Phrase	**Included?**	**Explanation**
'a decline'	No	The head of the bigger phrase 'a decline in ad pages in the first nine months of 1989'.
'a decline in ad pages in the first nine months of 1989'	Yes	Top-level NP
'ad pages'	Yes	NP not head of any other NP.
'the first nine months'	No	The head of the bigger phrase 'the first nine months of 1989'.
'the first nine months of 1989'	Yes	NP not head of any other NP.

Table 3.3: The NPs present in Example 3.52, and whether they are included after preprocessing.

Noun Phrase	**Included?**	**Explanation**
'Rudolph Agnew'	No	Head of larger NP.
'55 years'	No	An apposition.
'former chairman of Consolidated Gold Fields PLC'	No	Conjunction to apposition.
'former chairman'	No	Head of larger NP.
'Rudolph Agnew, 55 years old and former chairman of Consolidated Gold Fields PLC.'	Yes	Top-level NP
'Consolidated Gold Fields PLC.'	Yes	NP not head of any other NP.

Table 3.4: The NPs present in Example 3.53, and whether they are included after preprocessing.

2. If the NP is a conjunction → plural.

3. If the head POS tag of the NP is that for a singular noun or singular proper noun → singular.

4. If the head POS tag of the NP is that for a dollar or pound value → singular.

5. If the head POS tag of the NP is that for a cardinal number, check whether it is a decimal number, the number 1 or likely to be a year. If so → singular, otherwise → plural.

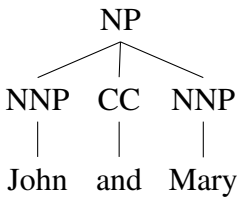
6. If the head POS tag of the NP is that for a personal pronoun, look it up in a list of singular and plural pronouns and return the according value. If it is ambiguous, return both.

7. If the head POS tag of the NP is that for a determiner, look it up in a list of singular determiners (e.g. 'this','that','another'). If present → singular, otherwise → plural.

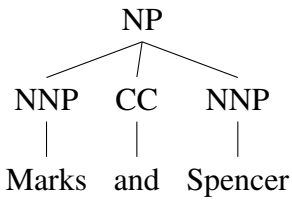
For ambiguous personal pronouns, such as 'you', which are very difficult to write a simple rule to classify, we simply allowed classification as both singular and plural, allowing the annotator to choose at annotation time. The incorrectly categorised instance of the pronouns were marked as non-mentions in the annotation tool.

The syntactic annotation of the Penn Treebank does not extend to the internal structure of base noun phrases (Vadas and Curran, 2007), meaning it is impossible to determine whether flat noun phrases containing conjunctions are plural or singular. Examples 3.54 and 3.55 show two NPs with identical structure, one of which is plural and one of which is singular. Both examples would be classified as singular with our algorithm, but the NP that is clearly plural should not be annotated as set member, and should be marked as *not a mention*.

(3.54)



(3.55)



A variety of other rules were needed to deal with rare cases, and the peculiarities of the dependency parses and errors in the PTB trees. A full pseudo-code representation of the algorithm is present as Appendix C.

Clearly, there are some singular nouns which would be valid sets, such as *family, set* or *group*, which are excluded by this algorithm. We feel the positive aspect of this process — reduction in the complexity of the annotation — outweighs the non-annotation of some possible instantiations. In the future, however, we intend to include such nouns, either by using a manually constructed list or employing lexicosyntactic patterns to automatically identify them.

3.5.4 Annotation tool

3.5.4.1 Annotation tool requirements

We had a range of requirements for our intersentential annotation tool:

- The ability to annotate a text a sentence pair at a time.
- The ability to display only the preselected NPs returned by the pre-processing steps detailed in Sections 3.5.2 and 3.5.3.
- To provide an interface which ensures that the annotator properly considers each potential instantiation, and has to provide some form of input before continuing to the next sentence pair.
- To provide separate views for annotating set members and subsets.
- To provide consistency between annotations where possible by incorporating coreference data.
- To provide the ability to view the sentence pair in the context of the whole text.
- To have the ability to mark NPs as non-mentions. This is required because some of the NPs that we consider non-mentions[3], such as idiomatic NPs, are very difficult to automatically identify.

For our intrasentential tool, we additionally had the requirement of preventing identical NPs being selectable as a subset.

These requirements were motivated by a desire to reduce annotation error and increase annotation efficiency.

[3]The NPs we consider non-mentions are described in Section 3.3.2.

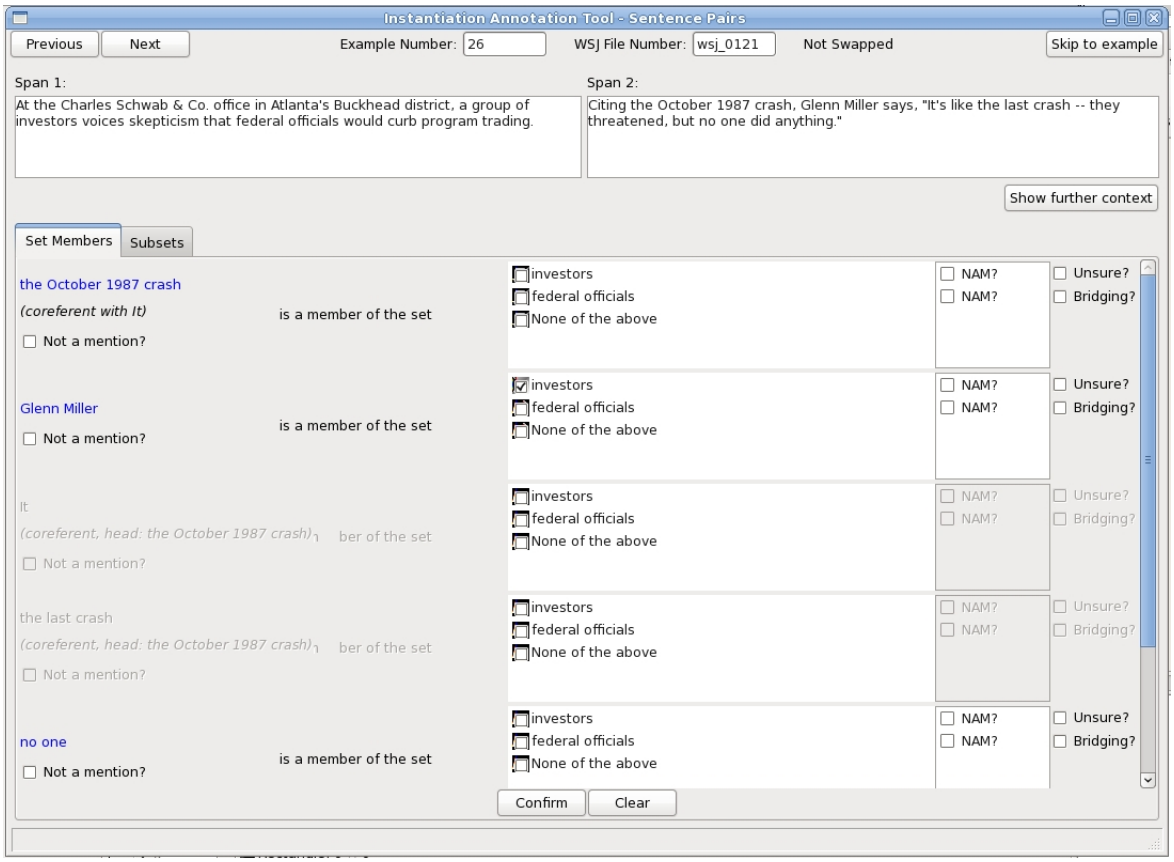


Figure 3.2: The completed annotation tool.

3.5.4.2 Annotation tool implementation

An image of the completed annotation tool is shown in Figure 3.2.

Figure 3.3 shows the tool, with its functions numbered. The numbers represent the following:

1. Navigation buttons, for navigating between sentence pairs.

2. Information panel, showing the example number, file number and whether the sentence pairs are being annotated backwards or forwards.

3. Further navigation option, allowing the annotator to skip to a particular numbered example.

4. The display boxes for the two sentences being annotated.

5. Tabs for selecting set membership or subset annotation. One may only select subset annotation when set membership annotation is complete.

6. A potential set member to be annotated. Note the presence of the *'Not a mention'* check-box below.

7. The possible annotations for this set member.

8. *'Not a mention'* check boxes for the potential sets.

9. Greyed out instances are coreferent to another NP within the sentence. Their annotation is filled when the head of the coreference chain in this particular sentence pair is annotated.

10. Confirm and clear buttons. One may only confirm once all annotation for set members is complete. Confirm then shows the subset annotation panel, as seen in Figure 3.4.

11. Show further context button. In certain circumstances it is useful to re-read other parts of the document. This button shows a dialogue containing the whole text, with the current sentence pair highlighted. This is shown in Figure 3.5.

The intrasentential annotation tool looks much the same, with the exception of having only one display box. An image is shown in Figure 3.6. One may also note that in the subset view, we prevent a plural NP from being accidentally marked as an instantiation of itself, or another coreferent NP.

3.6 Annotation Results

3.6.1 Agreement study

3.6.1.1 Intersentential agreement

To ascertain the reliability of our intersentential annotations, we undertook a short agreement study. Five texts containing a total of 6,177 NP pairs were independently annotated by the author of this thesis and the author's academic supervisor, Dr Katja Markert. Agreement was measured in the following three variations:

1. Does this pair of candidate noun phrases participate in a set membership/subset relationship or not?

2. Does this candidate set member/subset participate in a set membership/subset relationship with any potential set or not?

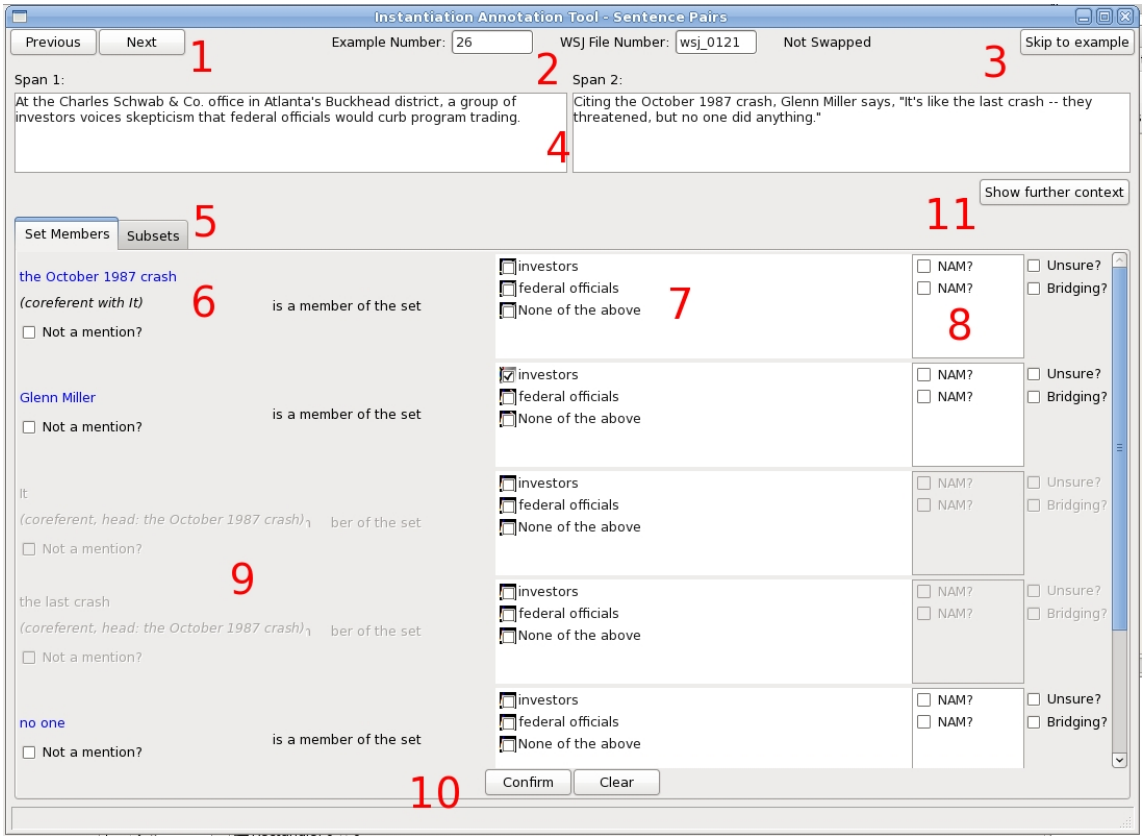


Figure 3.3: The annotation tool with numbered functions.

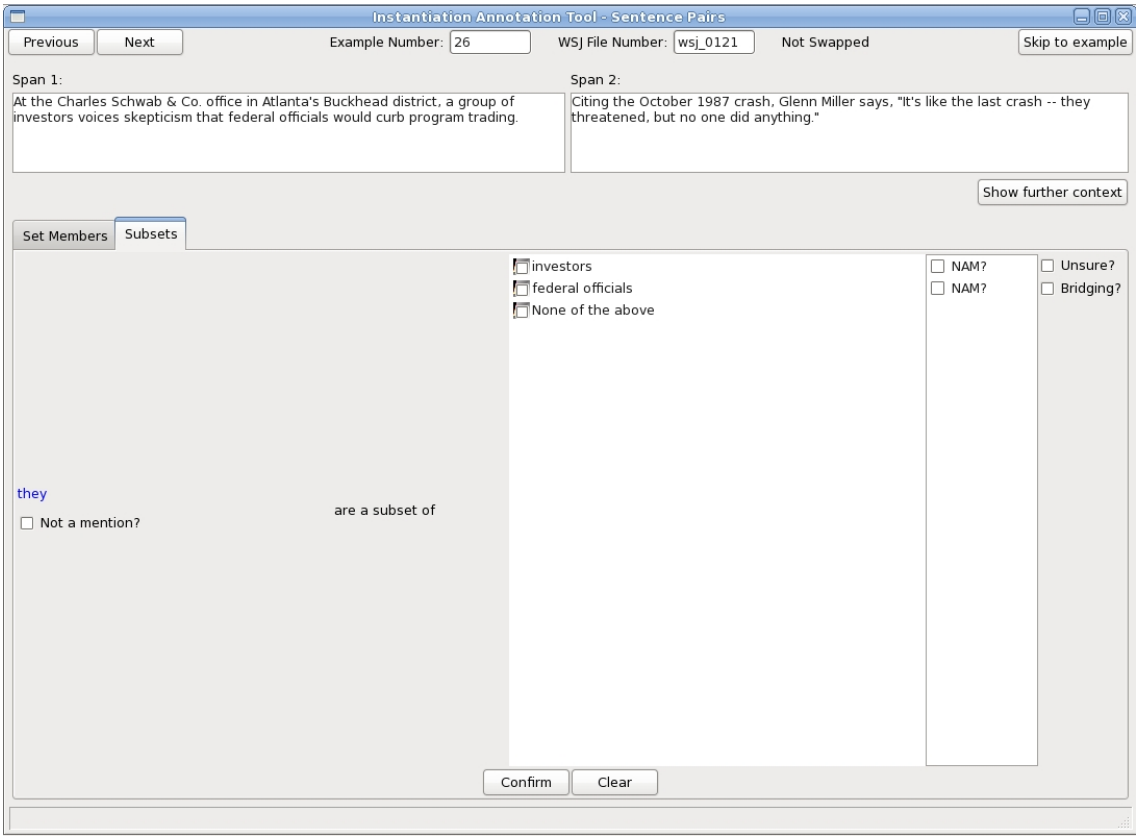


Figure 3.4: The annotation tool showing the Subset panel.

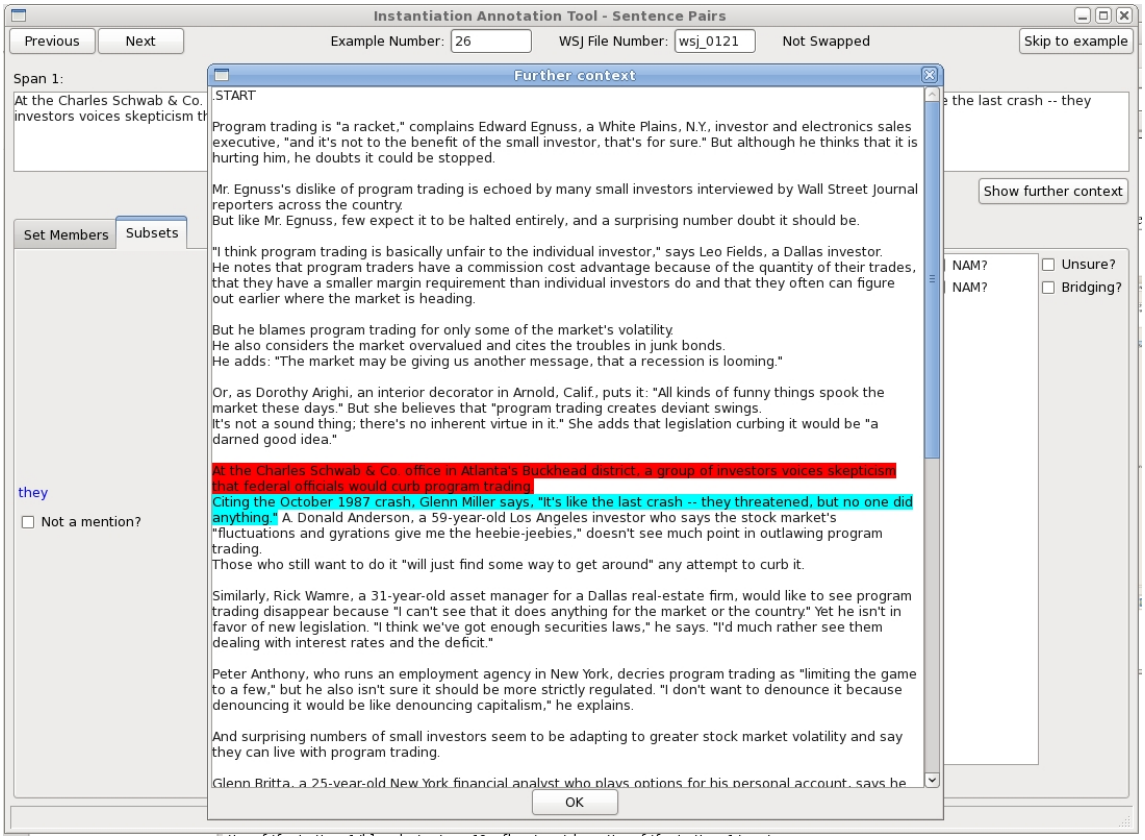


Figure 3.5: The Show Further Context function of the annotation tool.

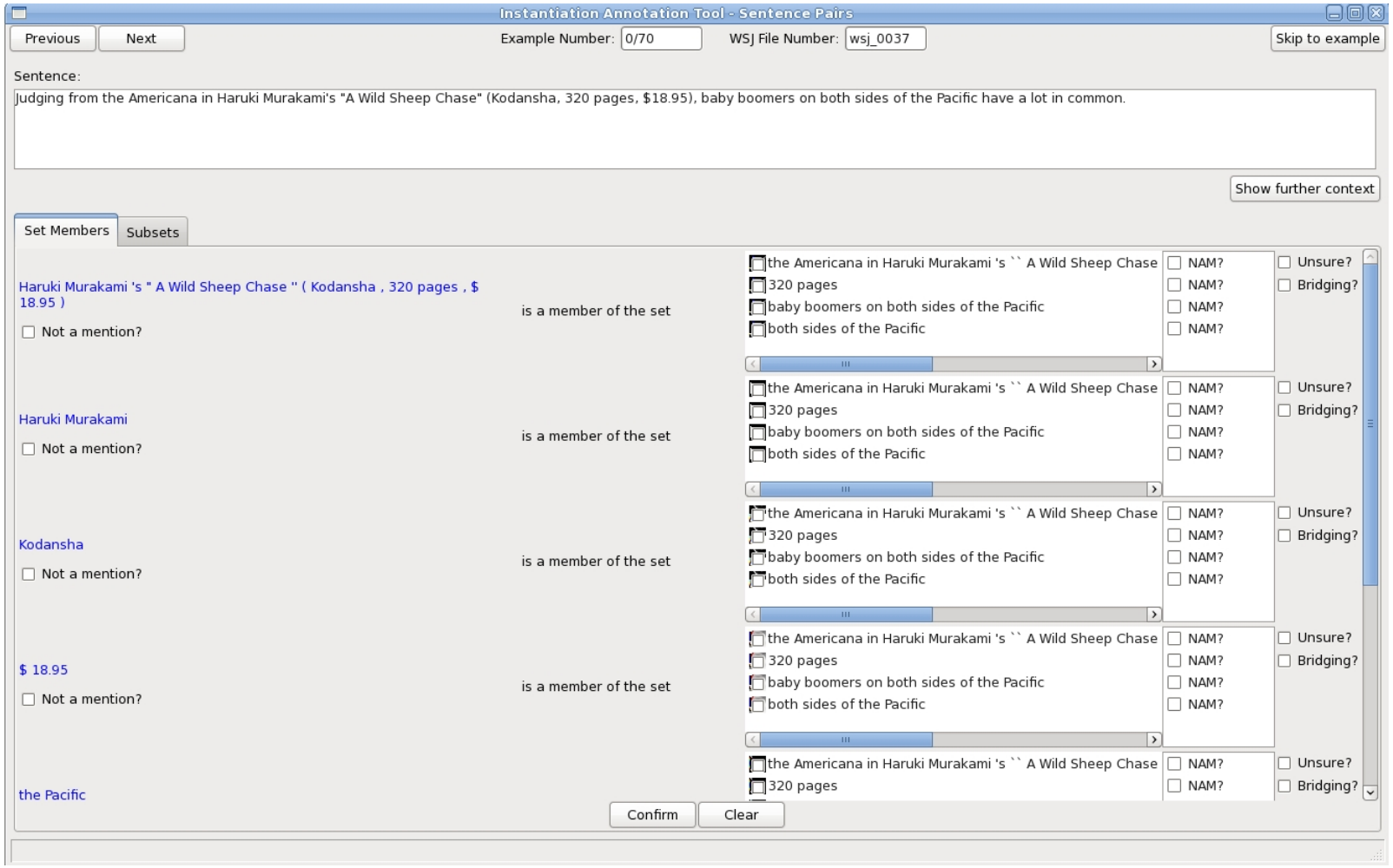


Figure 3.6: The intrasentential annotation tool.

Method	# of items tested	Kappa	Agreement
1	6177 pairs of NPs	0.6504	97.31%
2	2994 candidate set member/subset NPs	0.6403	95.23%
3	607 sentence pairs	0.7317	91.09%

Table 3.5: Results of the intersentential agreement study

3. Is there an Entity Instantiation between these two sentences?

The results of the agreement study, including percentage agreement and chance corrected agreement (Kappa, (Cohen, 1960)), are presented in Table 3.5. Our agreement about which candidates were *"Not a mention"* was $\kappa = 0.7146$. These agreement statistics show reasonable agreement on the task, and that our annotation scheme is reliable.

3.6.1.2 Intersentential disagreement analysis

There were at least 3 re-occurring types of disagreements:

Omissions and misinterpretations. We found that a large number of the disagreements were down to either simple omissions, or misinterpretations. For instance, in Example 3.56, one annotator missed the instantiation between *'firms with more than 10 employees'* and *large shopping centers'*. In Example 3.57, one annotator mistakenly assumed that *'Some Wall Street firms'* included the 6 firms mentioned in the prior sentence, when in fact it refers to financial services or banking companies.

(3.56) a. The New York study found that the cost of security measures in firms with fewer than five employees was almost $1,000 per worker, compared with one-third that amount for **firms with more than 10 employees**.

b. The shift of retailing to *large shopping centers* has created even greater economies of scale for providing low-crime business environments.

(3.57) a. Among companies saying they monitor employees are *United Airlines, American Airlines, United Parcel Service, Nynex Corp., Spiegel Inc., and the circulation department of this newspaper*.

b. **Some Wall Street firms** monitor for recordkeeping purposes.

It is hard to see how omissions and misinterpretations of this type could be removed entirely — the possibility of human error is hard to eliminate. Doubly-annotating the

texts, or redesigning the annotation tool to ensure that each possible instantiation is properly considered are potential solutions, but both would slow down the annotation process significantly.

Determining whether two sets are subsets, coreferent or overlapping. As we discussed in Section 3.4, It can be difficult for annotators to establish whether a pair of sets are subsets, coreferent or overlapping. In Example 3.58, it is difficult to establish the relationship between *'surveillance gear'* and *'their products'*. Do the vendors sell other products than surveillance gear? If so, it is a subset of *'their products'*. If instead they sell nothing but surveillance gear, the two NPs could be coreferent. It is hard to tell from the two sentences, and the surrounding context does not illuminate the matter either.

Similarly, in Example 3.59, it is hard to tell whether the *'1,124 businesses'* are the entirety of the *'small businesses there'* or a subset of them.

(3.58) a. Some marketers of *surveillance gear* – including Communication Control System Ltd., which owns the Counter Spy Shop and others like it – already put warning labels in their catalogs informing customers of the one-party law.

b. But vendors contend that they can't control how **their products** are used.

(3.59) a. A survey of **small businesses there** was conducted this spring by Interface, a policy research organization.

b. It gave *1,124 businesses* a questionnaire and analyzed 353 responses.

Systematic polysemy. Another problematic issue was systematic polysemy. In Example 3.60, *'Most cosmetic purchases'* might comprise a set of transactions or a set of products. The result of this interpretation then affects whether one considers *'lipstick'* to be a set member.

In Example 3.61, one has to decide whether an acquisition is the act of acquiring the brand, and therefore there is no instantiation, or the brand that will be acquired, in which case there is an instantiation present.

(3.60) a. **Most cosmetic purchases** are unplanned.

b. *Lipstick* is often bought on a whim.

(3.61) a. Members of the audience gasped or laughed nervously; their industry has been unsettled recently by **acquisitions**.

b. First Unilever, the Anglo-Dutch packaged-goods giant, spent $2 billion to acquire brands such as *Faberge and Elizabeth Arden*.

Method	# Items Tested	Kappa	Agreement
1	3098 NP pairs	0.7493	97.81%
2	1414 NPs	0.7742	96.39%
3	237 sentences	0.7277	89.87%

Table 3.6: Results of the intersentential agreement study

It is difficult to see how issues relating to systematic polysemy, and determining coreference/overlap/subsets between sets, can be easily resolved by a simple rule change. The decision is down to the individual interpretation of the text.

We also note that disagreements often propagated. A single decision about the relationship between two entities early on in a text can result in a large number of follow-on disagreements, due to subsequent coreferent mentions within the text.

3.6.1.3 Intrasentential agreement study

Despite the differences between inter- and intrasentential annotation process being minor, and the intersentential annotation scheme being shown to be reliable (Section 3.6.1.1), we undertook a further, intrasentential, agreement study. Again, five texts were selected randomly and then annotated by the author of this thesis and Dr Katja Markert independently. We measured agreement in the same three ways as in Section 3.6.1.1.

The results, showing both Kappa (Cohen, 1960) and percentage agreement, are shown in Table 3.6. We achieve good agreement with all three metrics, exceeding the agreement figures shown for the intersentential annotation. This suggests that intrasentential annotation is more straightforward than its intersentential counterpart.

3.6.1.4 Intrasentential disagreement analysis

Our disagreements for intrasentential annotation were similar to the intersentential disagreements, with fewer omissions due to the reduced scope each annotator had to consider.

We again found disagreements related deciding whether two sets were in a subset relationship or overlapping, such as *'the key districts'* and *'the state's major cities'* in Example 3.62, and whether *'some'* is a subset of *'most of "the volunteers"'* in Example 3.63.

(3.62) With ballots from *most of the state's major cities* in by yesterday morning, the Republicans came away with 10% of the vote in several of **the key districts**.

(3.63) Mind you, **most of "the volunteers"** would be unskilled 17- to 18-year-olds, *some* not even high school graduates, and many saving money by living at home.

We found one rare intrasentential specific type of disagreement, relating to nested NPs which were part of a larger expression. In Example 3.64, can one say that there are a set of *'ways'*, of which the *'dozens'* are a subset? Example 3.65 is slightly more concrete — there are certainly *'jobs'*, and one could say that *'the same kinds of jobs'* is a subset. These examples raise the question of whether 'ways' or 'jobs' are mentions, or whether only the larger phrase is a mention, and rely on the annotator's interpretation to some degree.

(3.64) *dozens of* ***ways***

(3.65) *the same kinds of* ***jobs***

3.7 Restriction of Annotations

We restrict the scope of our annotation, by allowing intersentential annotation to occur only between adjacent sentences. We restrict this on the basis that marking entity instantiations between all available NPs leads to complex annotation, making it very difficult for an annotator to track entity instantiations and leading to errors.

Clearly, this restriction will lead to the omission of some entity instantiations. To investigate the proportion of entity instantiations that may be omitted, non-adjacent intersentential instantiations were annotated over a sample of 3 texts, in two ways. Firstly, we used coreference data from OntoNotes (Weischedel et al., 2011) to automatically extrapolate intersentential instantiations between non-adjacent sentences from inter- and intrasentential instantiations. Secondly, we manually annotated the texts for further non-sentence-adjacent intersentential instantiations. The annotator was the author of this thesis. Details of the numbers of instantiations found are shown in Table 3.7.

Annotating the texts without restriction was difficult; on an initial pen-and-paper attempt it proved very challenging to consider all associations, and even with the aid of a GUI-based tool highlighting the potential set members and subsets for each set and automatically marking coreferent mentions the task was substantially harder than simply considering intrasentential and adjacent intersentential annotations. Achieving acceptable inter-annotator agreement on such a problem would seem unlikely without extensive training.

Adjacent intersentential and intrasentential entity instantiations formed 18.0% of all annotated instantiations (15.3% of set members, 20.9% of subsets), with automatically identified non-adjacent intersentential entity instantiations, which are repeats of already identified instantiations, forming 47.0% of annotated instantiations (57.9% of set members, 24.3% of subsets). Manually identified non-adjacent intersentential entity instantia-

Text	# Sentences	# Intrasentential E.Is		# Adjacent Intersentential E.Is		# Coreference-based Non-adjacent Intersentential E.Is		# Manually identified Non-adjacent Intersentential E.Is	
		Set members	Subsets	Set members	Subsets	Set members	Subsets	Set members	Subsets
wsj_0598	43	12	2	4	1	50	2	10	1
wsj_1570	57	29	71	25	30	66	15	151	314
wsj_2454	35	23	19	20	15	311	143	137	46
All	135	64	92	49	46	427	160	198	361

Table 3.7: Frequency of intrasentential, adjacent intersentential and non-adjacent intersentential entity instantiations in 3 sample texts

tions formed 40.0% of annotated instantiations, breaking down to 26.8% of set members and 54.8% of subsets.

The number of non-adjacent manually identified instantiations varied considerably between the three texts. In the case of `wsj_1570`, the number is inflated by the very general mention *'taxpayers'*, with which almost every person or group of people in the text is in an instantiation. Similarly, `wsj_2454`, a text about apartheid South Africa, contains the mention *'blacks'*, which participates in an instantiation with the vast majority of persons in the text. This large variation between texts suggests that the texts chosen might not be representative of a typical text in the corpus, and a bigger annotation study would be required to fully understand the proportion of entity instantiations that exist outside our restrictions.

3.8 Gold Standard Corpus

A further 70 texts were annotated both inter- and intrasententially by the author of this thesis. In this Section, we analyse the dimensions of the completed corpus and the text types contained within it (Section 3.8.1), as well as the distribution of entity instantiations over the annotated texts (Section 3.8.2).

Additionally, we computed a number of pertinent statistics about the annotated instantiations, which are detailed in Sections 3.8.3 and 3.8.4 below.

3.8.1 Corpus dimensions

The number of words and sentences contained within the 75 texts of the corpus is shown in Table 3.8. The corpus contains over 100,000 words and over 4,000 sentences in total.

Table 3.8 also shows the *genres* that these texts belong to. We follow the genre distinctions formulated by Webber (2009), who classify the texts into four broad genres that occur in the PDTB, defined below[4]:

NEWS Texts containing news reports.

ESSAYS Op-Ed pieces,[5] reviews ending with a byline, sourced articles from another

[4]The author also identifies the additional genres of CORRECTIONS, WIT AND SHORT VERSE and QUARTERLY PROFIT REPORTS, but does not include them in her analysis on the basis that *"they are so obviously different from the other texts"*. None of our 75 texts fall in to these genres, so we too ignore them.

[5]Op-Ed pieces are defined as "The page of a newspaper facing the editorial page, typically devoted to personal comment and feature articles." (OED Online, 2013).

	Corpus Size		
Genre	Words	Sentences	Texts
All Genres	104 711	4 254	75
News	76 009	3 115	54
Essays	26 482	1 042	19
Summaries	2 220	97	2

Table 3.8: Total corpus size in words, sentences and texts

newspaper or magazine, editorials and other reviews but without a source or a by-line or essays on topics commemorating the WSJ's centennial.

SUMMARIES Daily summaries of offerings and pricings in capital markets, daily summaries of financially significant events, daily summaries of interest rates, summaries of recent SEC filings and weekly market summaries.

LETTERS Letters to the editor.

As the texts are a subset of those examined in Webber (2009), we used the list of files corresponding to each genre made available by the author rather than manually inspecting and judging them.[6]

The vast majority of files in the PDTB — 1,902 out of 2,110 — belong to the NEWS genre, and this is reflected in the fact that 54 of our 75 randomly selected texts are NEWS texts. Our corpus has a considerably higher proportion of the ESSAY genre than the full PDTB (25.3% vs 4.9%), and does not contain any texts belonging to the LETTERS genre.

Tables 3.9 and 3.10 show the distribution of the lengths of the texts in each genre, measured in words and sentences, respectively. We see that the NEWS texts have a slightly higher mean length and much higher variance than the ESSAY texts. The two SUMMARIES texts are considerably shorter than the other two genres.

We conducted a two-sampled t-test to compare the means of the NEWS and ESSAYS genres, and found no significant difference between them. We also applied the F-test for equality of variances, and found significant differences for both the measurements in sentences and in words ($p < 0.01, (53, 18)$ d.f.), suggesting a difference in the distribution of the length of the texts. Due to the fact that only two SUMMARIES are present in the corpus, we did not include them in any statistical test which compares genre, nor do we in Section 3.8.2.

[6]The list of files corresponding to each genre from Webber (2009) is available from `http://www.let.rug.nl/~bplank/metadata/genre_files_updated.html`.

Genre	Maximum	Minimum	Median	Mean	Standard Deviation
All Genres	3946	788	1271.0	1396.15	500.65
News	3946	788	1259.5	1407.57	557.71
Essays	2497	974	1352.0	1393.79	324.15
Summaries	1127	1093	1110.0	1110.00	24.04

Table 3.9: Distribution of text lengths in corpus by genre, measured in words.

Genre	Maximum	Minimum	Median	Mean	Standard Deviation
All Genres	167	28	52.0	56.72	21.78
News	167	28	51.0	57.69	24.47
Essays	93	32	56.0	54.84	13.09
Summaries	53	44	48.5	48.5	6.36

Table 3.10: Distribution of text lengths in corpus by genre, measured in sentences.

3.8.2 Entity instantiation distribution

Having established the dimensions of the corpus, and the fact that the genre of the texts affects their length, we next explored the distribution of entity instantiations over the 75 texts. Table 3.11 shows the frequency of the entity instantiations in the corpus, including the numbers of inter- and intrasentential instantiations, and set members and subsets.

Next, we considered the frequency distribution of instantiations over the texts. Table 3.12 shows a number of relevant metrics regarding the distribution of instantiations, as well as a breakdown by genre. As in the previous Section, we tested the significance of difference between the means of the ESSAYS and NEWS genres using a two-sampled t-test, and the difference between the variances using an F-test. We found no significant differences in either case. We also found no significant differences when we considered solely intrasentential and solely intersentential instantiations.

However, when we considered solely set member entity instantiations or subset instantiations, we did find significant differences between the means associated with each genre, suggesting that set membership occurs more frequently in NEWS texts than ESSAYS, and that the converse is true for subsets.

Next, we considered the frequency of entity instantiations per sentence, or per sen-

	Intersentential	Intrasentential	Total
Set Member	1 477	1 538	3 015
Subset	641	865	1 506
Total	2 118	2 403	4 521

Table 3.11: Frequency of entity instantiations in 75 texts

Genre	Instantiation Type	Maximum	Minimum	Median	Mean	Standard Deviation
All Genres	All Instantiations	159	19	54	60.28	32.32
	— All Intrasentential	100	10	28	32.04	16.91
	— All Intersentential	85	3	23	28.24	18.79
	— All Set Members	125	10	35	40.20	21.74
	— All Subsets	101	2	15	20.08	16.65
News	All Instantiations	159	19	55	61.94	32.60
	— All Intrasentential	74	10	28	31.19	15.79
	— All Intersentential	85	5	27	30.76	19.18
	— All Set Members	125	10	39	44.28	22.93
	— All Subsets	74	2	13	17.67	13.97
Essays	All Instantiations	155	26	46	57.26	33.31
	— All Intrasentential	100	13	29	35.32	20.12
	— All Intersentential	78	3	18	21.95	17.24
	— All Set Members	73	13	28	30.11	14.32
	— All Subsets	101	7	23	27.16	22.22
Summaries	All Instantiations	52	36	44	44.00	11.31
	— All Intrasentential	36	12	24	24.00	16.97
	— All Intersentential	24	16	20	20.00	5.66
	— All Set Members	35	17	26	26.00	12.73
	— All Subsets	19	17	18	18.00	1.41

Table 3.12: Distribution of entity instantiations in corpus per text, by genre.

tence pair for intersentential entity instantiations, rather than the per text frequency. Tables 3.13 and 3.14 show relevant metrics for intrasentential and intersentential entity instantiations, respectively. Due to the fact that the data certainly does not follow a normal distribution — the majority of sentences contain zero instantiations — we applied the non-parametric Mann-Whitney U test, rather than the t-test and F-test.

In the case of intrasentential instantiations, we find no significant differences between the genres when considering both set members and subsets, or only set members. However, there is a significant difference between the distribution of subsets across genres ($p < 0.05$) — intrasentential subset instantiations occur more often in ESSAYS.

For intersentential instantiations, we find significant differences between the NEWS and ESSAYS genres, when we consider set members and subsets together, and set members in isolation ($p < 0.05$). Intersentential set members occur significantly more often in NEWS texts.

The presence of these significant differences suggests that it could be useful to take the genre of the texts into account in the classification process. Although we do not implement features that indicate text genre, we would like to explore this further in future. It also suggests that creating a more balanced corpus, in terms of genre, would be a sensible extension of our annotation.

In the subsequent Sections of this Chapter, and in the remainder of the Thesis, we consider the problem of identifying entity instantiations as one of distinguishing between *instantiation* NP pairs and *non-instantiation* NP pairs. Therefore, we calculate the number of NPs with no instantiation, whether the NP pairs exist within single sentences for the intrasentential case, or between adjacent sentences for the intrasentential case, along with the numbers of each positive entity instantiation instance.

The frequency distribution of instantiations in the intersentential corpus is shown in Table 3.15. The distribution for the intrasentential corpus is shown in Table 3.16.

3.8.3 Intrasentential analysis: syntactic arrangement of noun phrases

We organised the syntactic arrangement of NP pairs into four categories: the set NP is a parent of the member/subset NP, the member/subset NP is a parent of the set NP, they occur in the same clause but not in a parent/child relationship and they occur in different clauses. The distribution of instantiations amongst these classes, along with the distribution of non-instantiation NP pairs for comparison, is shown in Tables 3.17 and 3.18 for set members and subsets respectively.

The majority of intrasentential instantiations consist of instances where the set is the

Genre	Instantiation Type	Maximum	Minimum	Median	Mean	Standard Deviation
All Genres	All Instantiations	20	0	0.0	0.56	1.41
	— Set Members	18	0	0.0	0.36	1.04
	— Subsets	13	0	0.0	0.20	0.77
News	All Instantiations	20	0	0	0.54	1.33
	— Set Members	18	0	0	0.37	1.06
	— Subsets	8	0	0	0.17	0.63
Essays	All Instantiations	18	0	0.0	0.64	1.62
	— Set Members	12	0	0.0	0.33	0.97
	— Subsets	13	0	0.0	0.31	1.08
Summaries	All Instantiations	7	0	0	0.49	1.39
	— Set Members	6	0	0	0.33	0.99
	— Subsets	7	0	0	0.16	0.81

Table 3.13: Distribution of intrasentential entity instantiations in corpus per sentence, by genre.

Genre	Instantiation Type	Maximum	Minimum	Median	Mean	Standard Deviation
All Genres	All Instantiations	40	0	0	0.51	1.36
	— Set Members	21	0	0	0.35	0.98
	— Subsets	19	0	0	0.15	0.67
News	All Instantiations	40	0	0	0.54	1.45
	— Set Members	21	0	0	0.40	1.07
	— Subsets	19	0	0	0.14	0.65
Essays	All Instantiations	11	0	0	0.41	1.09
	— Set Members	7	0	0	0.22	0.70
	— Subsets	7	0	0	0.18	0.71
Summaries	All Instantiations	8	0	0	0.42	1.13
	— Set Members	2	0	0	0.21	0.52
	— Subsets	6	0	0	0.21	0.86

Table 3.14: Distribution of intersentential entity instantiations in corpus per sentence pair, by genre.

Entity Instantiation	**# NP pairs**	**%**
Set Member	1477	1.89
Subset	641	0.82
No inst. plural-singular	46 128	59.11
No inst. plural-plural	29 793	38.18
Total	78 039	100

Table 3.15: Frequency of intersentential entity instantiations and non-instantiation NP pairs in 75 texts

Entity Instantiation	**# NP pairs**	**%**
Set Member	1 538	3.51
Subset	865	1.98
No inst. plural-singular	24 363	55.63
No inst. plural-plural	17 028	38.88
Total	43 794	100

Table 3.16: Frequency of intrasentential entity instantiations and non-instantiation NP pairs in 75 texts

parent of the member/subset. Instances with the member/subset as parent and instances where the NPs occur in the same clause are relatively infrequent, but instances in separate clauses comprise a significant percentage of instantiations, especially for set member instantiations.

To test the significance of the difference between the distribution of instantiations and non-instantiations we used a χ^2 test for consistency in a 4×2 table. This gave $\chi^2 = 4605$ for set members and $\chi^2 = 3123$ for subsets, both corresponding to $p = 0$, making the increased proportion of instances where the set is parent highly significant.

These statistics suggest that syntax dependent features, such as tree kernels, are likely to be appropriate for the intrasentential problem. They also suggest that a proportion of intrasentential instantiations have participants in different clauses and are likely to behave similarly to intersentential instantiations, and that a single feature set may therefore be used to tackle both problems.

3.8.4 Intersentential analysis

3.8.4.1 Ordering of and distance between noun phrases

We investigated the ordering of intersentential entity instantiations, by calculating the proportion of each entity instantiation subtype with the set preceding and the set succeeding the member/subset. For comparison, we also calculated the distribution of plural-singular

Relationship	**Set Member**	**Other Sing-Plur pair**	**Total**
Set NP Parent	1 065	2 294	3 359
Member NP Parent	55	1 843	1 898
Same Clause	84	7 068	7 152
Different Clause	334	13 158	13492
Total	1 538	24 363	25 901

Table 3.17: Frequency of syntactic relationships between NPs in intrasentential set member entity instantiations.

Relationship	**Subset**	**Other Plur-Plur pair**	**Total**
Set NP Parent	615	1 489	2 104
Subset NP Parent	85	1 991	2 076
Same Clause	90	4 945	5 035
Different Clause	75	8 603	8 678
Total	865	17 028	17 893

Table 3.18: Frequency of syntactic relationships between NPs in intrasentential subset entity instantiations.

and plural-plural NP pairs in the corpus with no annotation in the corpus in the same way. The results are shown in Tables 3.19 and 3.20 for set members and subsets respectively. We find that for both set member and subset instantiations, the set precedes the member/subset more frequently than it succeeds it in the text.

We carried out similar significance tests to those described in Section 3.8.3, but this time for consistency in a 2×2 table. For set members, $\chi^2 = 22.922$ and $p = 0.00000169$, and for subsets $\chi^2 = 15.847$ and $p = 0.00006868$, showing highly significant differences between the distribution of instantiations and non-instantiations in both cases.

We performed similar experiments which illustrate the distance in words between a member/subset and its set, normalised by the number of words in the sentence pair containing the instantiation. The results are shown in Tables 3.21 and 3.22 for set members and subsets respectively. We also performed the same experiments but instead measured the distance in characters, with similar results. Our experiment shows that noun phrases participating in instantiations are, on average, more proximate than other NPs.

Category	**Set Member**	**Other Sing-Plur pair**	**Total**
Set First	816	22 566	23 382
Set Second	661	23 562	24 223
Total	1 477	46 128	47 605

Table 3.19: Ordering of NPs in intersentential set member entity instantiations

Category	**Subset**	**Other Plur-Plur pair**	**Total**
Set First	368	14 737	15 105
Set Second	273	15 056	15 329
Total	641	29 793	30 434

Table 3.20: Ordering of NPs in intersentential subset entity instantiations

Metric	**Set member**	**Other Sing-Plur pair**
Min	0.01	0.00
Max	0.95	0.96
Mean	0.39	0.42
Median	0.39	0.42
Standard Deviation	0.21	0.22

Table 3.21: Distribution of intersentential set member entity instantiations by normalised distance in words between noun phrases.

Metric	**Subset**	**Other Plur-Plur pair**
Min	0.01	0.00
Max	0.91	0.96
Mean	0.40	0.42
Median	0.39	0.42
Standard Deviation	0.21	0.22

Table 3.22: Distribution of intersentential subset entity instantiations by normalised distance in words between noun phrases.

3.8.4.2 Noun phrase categorisation

We also wished to examine the types of NPs that participated in entity instantiations. We did this in three stages. Firstly, we examined the type of set member/subset NP, in terms of the part of speech of its head. Secondly, we examined the modification level of set member/subset NPs, and finally we examined the type of the set NPs from which members and subsets were drawn.

Set member and subset NP types. We extracted the head word of each of our set member/subset NPs, using dependency parse trees generated from the gold standard Penn Treebank trees of the sentences using the Penn Converter tool (Johansson and Nugues, 2007). The headword is calculated by examining the dependency parse of the NP, and selecting the word which is *not* dependent on another word in the NP[7]. Based on the head word, we classified our NPs into the following categories:

Name. The head word is in a named entity, or its POS tag begins with NNP, and is therefore a proper noun. Named entity data was generated using the Stanford Named Entity Recognizer (Finkel et al., 2005).

Pronoun. The POS tag of the head word begins with PRP.

Common Noun. The POS tag of the head word begins with NN (and does *not* begin with NNP).

Numeric. The POS tag of the head word is CD, or the head word contains '%', '$' or ''.

Other. The head word does not fit into any of the above categories. This includes adjectives (e.g. *'the biggest'*), determiners (e.g. *'this', 'that'*), and gerunds (e.g. *'the rewriting', 'mourning for the victims'*).

The results are shown for set members and subsets in Table 3.23 and 3.24 respectively. We find our set member distribution to be significantly different to other singular NPs ($\chi^2 = 593, 4$ *d.f.,* $p < 0.0001$), featuring a much higher proportion of names and pronouns than non-instantiation singular NPs. Our subset distribution is also significantly different to other plural NPs ($\chi^2 = 29, 4$ *d.f.,* $p < 0.0001$), with a slightly smaller proportion of common nouns. When we compare set members with subsets, we see a clear difference — set members are very often names, subsets are much more likely to be headed by common nouns.

[7] See Section 4.1.1.3 for a further description of this process.

NP Type	**Set member NP**		**Other singular NP**	
Name	716	(48.48%)	11900	(25.80%)
Pronoun	228	(15.44%)	3527	(7.65%)
Common Noun	471	(31.89%)	27080	(58.71%)
Numeric	32	(2.17%)	2313	(5.01%)
Other	30	(2.03%)	1308	(2.84%)
Total	1477	(100.00%)	46128	(100.00%)

Table 3.23: Intersentential set member NP categorisation.

NP Type	**Subset NP**		**Other singular NP**	
Name	41	(6.40%)	1213	(4.07%)
Pronoun	65	(10.14%)	2725	(9.15%)
Common Noun	490	(76.44%)	24706	(82.93%)
Numeric	13	(2.03%)	403	(1.35%)
Other	32	(4.99%)	746	(2.50%)
Total	641	(100.00%)	29793	(100.00%)

Table 3.24: Intersentential subset NP categorisation.

Set member and subset NP modification. We record the modification level of set member and subset NPs. For each NP we again extract the head word, and then look for words *within* the NP that depend upon it, excluding links which represent name-internal links, such as titles and post-honorifics, as well as punctuation and possessive suffixes. We classify the NPs into one of four categories:

Pre. The NP head is pre-modified.

Post. The NP head is post-modified.

Both. The NP head is both pre- and post-modified.

Neither. The NP head has no pre- or post-modification.

The results are shown for set members and subsets in Table 3.25 and 3.26, respectively. We find significant differences between the distribution of instantiation NPs and non-instantiation NPs in both cases ($\chi^2 = 179, 3$ *d.f.,* $p < 0.0001$ for set members, $\chi^2 = 23, 3$ *d.f.,* $p < 0.0001$ for subsets).

We notice that set member NPs are more often post modified or without modification than their non-instantiation counterparts. Subset NPs more often have post modification or both pre- and post-modification than non-instantiation plural NPs.

Modification type	**Set member NP**		**Other singular NP**	
Pre	386	(26.13%)	17774	(38.53%)
Post	227	(15.37%)	3933	(8.53%)
Both	225	(15.23%)	8687	(18.83%)
Neither	639	(43.26%)	15734	(34.11%)
Total	1477	(100.00%)	46128	(100.00%)

Table 3.25: Intersentential set member NP modification.

Modification type	**Subset NP**		**Other plural NP**	
Pre	247	(38.53%)	12826	(43.05%)
Post	109	(17.00%)	3957	(13.28%)
Both	162	(25.27%)	5992	(20.11%)
Neither	123	(19.19%)	7018	(23.56%)
Total	641	(100.00%)	29793	(100.00%)

Table 3.26: Intersentential subset NP modification.

Set NP types. We also examine the type of the set from which our set member/subset NPs are drawn. We use the same four categories as for our set member/subset type classification, with the addition of a category for conjunctive noun phrases.

The results are shown for set members and subsets in Table 3.27 and 3.28, respectively. We find that the distribution of set member set NPs is significantly different from non-instantiation set NPs ($\chi^2 = 302,5$ *d.f.*, $p < 0.0001$). For subsets, there is no significant difference between the distributions. We note that conjunctions, pronouns and names occur more commonly as the sets for set members than for subsets.

NP type	**Set member Set NP**		**Non-instantiation Set NP**	
Conjunction	178	(12.05%)	4168	(9.04%)
Name	42	(2.84%)	936	(2.03%)
Pronoun	284	(19.23%)	3690	(8.00%)
Common Noun	903	(61.14%)	35593	(77.16%)
Numeric	11	(0.74%)	630	(1.37%)
Other	59	(3.99%)	1111	(2.41%)
Total	1477	(100.00%)	46128	(100.00%)

Table 3.27: Intersentential set member set NP types

NP type	**Subset Set NP**		**Non-instantiation Set NP**	
Conjunction	57	(8.89%)	2536	(8.51%)
Name	18	(2.81%)	551	(1.85%)
Pronoun	60	(9.36%)	2847	(9.56%)
Common Noun	487	(75.98%)	22685	(76.14%)
Numeric	1	(0.16%)	374	(1.26%)
Other	18	(2.81%)	800	(2.69%)
Total	641	(100.00%)	29793	(100.00%)

Table 3.28: Intersentential subset set NP types

3.9 Conclusion

3.9.1 Summary

In this Chapter, we have described the creation of the first annotated corpus of entity instantiations. The inspiration for our methodology came from the problem of recognising textual entailment, and so we treat the annotation of entity instantiations as an applied task rather than enforcing strict logical rules.

We defined the problem in detail, demonstrating its breadth by means of numerous examples and setting out how our scheme dealt with special cases. We also considered some of the potential difficulties that occurred as a result of our approach.

The texts annotated formed part of the Penn Treebank Wall Street Journal corpus, a resource chosen because of the many other layers of annotation that exist for these texts, including part-of-speech tags, constituency parse trees, discourse relations and coreference. We developed an annotation tool specifically for the task, which displayed each sentence or sentence pair in turn, allowing for efficient annotation.

The reliability of the scheme was tested by two agreement studies, for inter- and intrasentential instantiations, respectively. Five randomly selected texts were independently annotated by two people, and chance-corrected agreement (Cohen, 1960) was calculated. The intersentential study showed reasonable agreement ($\kappa = 0.65$), and the intrasentential study showed higher levels of agreement ($\kappa = 0.75$), suggesting that the intrasentential task may be easier. Three common types of disagreement were found, relating to systematic polysemy, difficulties deciding between set coreference, set overlap and subset relationships and simple annotator errors.

Our gold standard corpus consists of 4,521 instantiations, 2,118 of which are intersentential, and 2,403 which are intrasentential, annotated over a total of 75 texts. We statistically analyse these instantiations in terms of their syntactic arrangement, the ordering of and distance between NPs, the type of NPs involved and their modification. There are

significant differences between the distributions of instantiations and non-instantiations when examined in this manner, suggesting potential features for identifying entity instantiations.

In the next chapter, we turn our attention to the automatic identification of entity instantiations, using the annotations described in this chapter for training and testing a supervised machine learning algorithm.

3.9.2 Future work

In the future, we wish extend the work of this chapter in a number of ways.

Firstly, we would like to increase the size of our annotated corpus. The corpus we introduce in this thesis covers 75 texts, and includes annotation of 4,521 instantiations. However, prior research (Banko and Brill, 2001) has shown that larger amounts of data can be highly beneficial, and aid in the development of better automatic identification methods and more sound statistical analysis.

Producing a corpus of similar dimensions to the PDTB, which covers over 2,000 texts, would be likely to give us more than 100,000 instantiations, allowing for future supervised machine learning to better capture outlier cases and create better rules for automatic identification, and for more revealing statistical analysis. Creating annotated data on that scale takes a great deal of time, effort and money; even a corpus of 150 texts may well give us a better understanding of the phenomenon.

Secondly, we wish to experiment with texts of different genres and sources. We decided to annotate WSJ texts, as they were also annotated as part of several other corpora, including the Penn Treebank, Penn Discourse Treebank, OntoNotes, PropBank and NomBank. This overlap with existing annotations facilitates easy study of the interaction between entity instantiations and other phenomena, and gives us the opportunity to develop features for machine learning which use these other annotations.

Whilst the texts annotated were not solely news reports, and include both essays and summaries (see Section 3.8.1 for a discussion of the distribution of genres in the corpus), their origin means they share some common drawbacks. For instance, the texts are over 20 years old, meaning they describe people and events that are unlikely to be prominent in modern knowledge bases, and many of them focus on economic matters, which rely on a knowledge of financial jargon to understand. More modern texts, meant for a more general audience, could be more straightforward to annotate.

However, the tendency of the articles to focus on real-world objects and things makes the identification of sets, members and subsets more straightforward than might be the

case in other genres. Identifying instantiations within a philosophy text, or a novel, could raise additional challenges, and it would certainly be interesting to see how the current annotation scheme fared in these circumstances.

Additionally, the texts are written in formal English, suitable for publication in a newspaper. They are well formatted and spelt, and written in grammatically correct English. Dealing with entity instantiations in less formal settings, such as web pages, blog posts, or tweets — which are generated online in vast amounts every day — could be harder, but also may offer more future applications for our work.

Thirdly, we imposed some restrictions on our annotation process, to reduce annotator effort. This included limiting the set NPs to plural NPs using the process specified in Section 3.5.3, which reduces the chances of relationships such as meronymy, employment or location being mistakenly marked as instantiations, and avoids some difficult decisions about whether an NP such as *'the parliament'* or *'the team'* is just a set of entities or is somehow more than the sum of its parts. However, this leads is the potential omission of some valid entity instantiations. One possible solution would be to identify singular NPs that can act as sets, by means of a manually generated list, or by developing an automatic algorithm, possibly based on lexicosyntactic patterns.

The annotation was also restricted to between adjacent sentences and within sentences. To explore the impact of this, three texts were annotated without restrictions. Although a significant number of additional entity instantiations were identified, annotating in this way was difficult — keeping track of sets and their members across a text of non-trivial length proved problematic. In the future, we wish to develop a better system for the annotation of unrestricted entity instantiations that will allow for reliable, replicable annotation.

We also simplify our annotation by including generics as possible sets. This approach causes few problems, possibly because of the aforementioned tendency of the source materials to focus on real-world objects. Should we choose to annotate texts where this caused a problem, we could use the methods of Reiter and Frank (2010), who use supervised machine learning to identify generic noun phrases, as a pre-processing step for our annotation.

Chapter 4

Machine Learning of Entity Instantiations

Having established in Chapter 3 that entity instantiations can be identified reliably by human annotators, we then experimented with automatic identification, using supervised machine learning.

4.1 Feature Design

We designed features to establish if an instantiation exists between pair of NPs. Specifically, we wish to learn whether a plural-singular NP pair represents a set member instantiation, and whether a plural-plural NP pair represents a subset instantiation.

The features described in this section were used for *both* intersentential and intrasentential instantiations. Extra features developed specifically for intrasentential instantiations are described in a separate section, Section 4.2.

The features we designed fall into five broad categories; *surface*, *salience*, *syntactic*, *contextual* and *knowledge*. These categories contain both features that pertain to a single NP, and those that represent cross-NP relationships.

All features that comprised a list of tokens — such as those described in Sections 4.1.1.1, 4.1.1.2, 4.1.1.3 and 4.1.1.4 — were included as counts of unigrams and bigrams. A full table of features is shown in Section 4.1.6.

4.1.1 Surface features

Our surface features are intended to capture relationships between the surface forms of the tokens involved in each noun phrase. This category of features also includes information regarding the part of speech of the tokens. We use gold standard tokenisation and POS tag data from the Penn Treebank throughout.

4.1.1.1 N-grams

We have two feature groups; one for the tokens of each of the two NPs involved in the potential instantiation. We use these features to capture recurring words and phrases which often signal the presence of an instantiation, including *'many', 'one of'* and *'some'*.

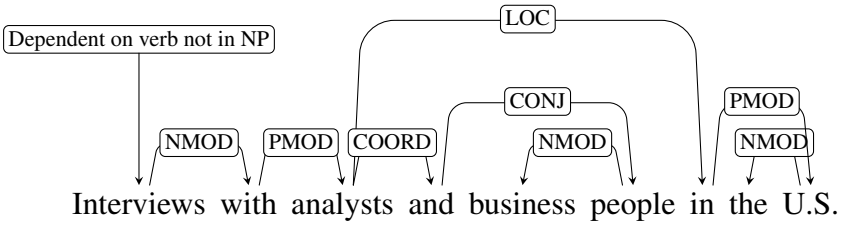
4.1.1.2 Part-of-speech

Similarly to the unigrams, we have two features, one for the POS tags of each of the two NPs. We use these to try capture common POS patterns in our NPs which may signal the presence or absence of an instantiation.

4.1.1.3 Head words

We derive the head word of each NP, and include these as features. To derive the head word, we use dependency parse trees, generated from the gold standard Penn Treebank trees of the sentences using the Penn Converter tool (Johansson and Nugues, 2007).

The headword is calculated by examining the dependency parse of the NP, and selecting the word which is *not* dependent on another word in the NP. For example, in the dependency parse below the word *Interviews* is dependent on a node not in the NP and is therefore the head of the phrase.



We include head words separately from the n-grams, hypothesising that these words may be more influential in deciding the nature of the relationship between the NPs.

4.1.1.4 Lemmas

We used WordNet's Morphy (Fellbaum, 1998) program to generate lemmas of each token, and included these as two features; one for each of the two NPs. Words that are not in WordNet are included in their original inflected form. Our reason for including lemmatised forms is that they make spotting related plural/singular word pairs such as *worker* and *workers* easier, as *workers* would be included as *worker*.

4.1.1.5 Head word lemmas

We also include lemmatised versions of head words as features.

4.1.1.6 Levenshtein's distance

We calculate Levenshtein's distance (Levenshtein, 1966) between the unigrams of both NPs, the lemmas of both NPs and the heads of both NPs. Levenshtein's distance is a metric which measures the minimum number of edits (substitutions, deletions or insertions) required to transform one string into another.

For example, to transform the string 'cat' into the string 'smart' we must perform the following 3 edits, giving us a Levenshtein distance of 3:

1. Insert 'r' (ca**r**t).
2. Substitute 'c' for 's' (**s**art).
3. Insert 'm' (s**m**art).

We hypothesise that the Levenshtein distance between pairs of head words could be especially useful at discovering pairs such as *'funds'* and *'fund'* (see Example 4.1), and could help capture relationships between pairs that were unable to be lemmatised by Morphy. Although head words are useful, the lack of context can be problematic. Examples 4.2 and 4.3 show situations in which head words may be more or less useful respectively.

(4.1) a. **Some European funds** recently have skyrocketed.
b. *Spain Fund* has surged to a startling 120% premium.

(4.2) a. **Many workers** are opposed to the regulations.
b. *Some workers* told us that that they are contemplating a strike.

(4.3) a. **Labour MPs** are voting for the measure.
b. *Liberal MPs* are split on the issue.

4.1.1.7 Distance

We calculate the distance in the text between the two NPs, both in terms of characters and words. We also include separate versions of these features normalised by the total number of characters and words in the containing sentences. We hypothesise that two NPs that are relatively close to each other in the text may be more likely to be part of an instantiation.

4.1.1.8 Ordering

We include a boolean feature which represents the ordering of the two NPs — set followed by set member/subset or set member/subset followed by set. We hypothesised that 'reversed' instantiations may behave differently than regular instantiations, and included a feature to make the difference explicit to the learner.

4.1.2 Salience features

As an indicator of the salience of each NP, we use the features described below.

4.1.2.1 Grammatical role

Using the dependency parse data described in Section 4.1.1.3, we extract the grammatical role of each NP. We hypothesise that NPs that fulfil important grammatical roles in a sentence, such as the subject or object, are more likely to be participants in an instantiation.

4.1.2.2 Mention count

We include the length of the coreference chain of the entity prior to this mention in the document, and the overall length of the coreference chain of the entity in the document. We expect that repeatedly mentioned, salient entities are more likely to be part of instantiations than rarely mentioned entities.

We separately include whether this is the first mention of the entity or not, as we hypothesise that those entities that are not especially salient may be introduced in a text by means of an instantiation, as in Example 4.4.

(4.4) a. Already, scientists are developing tests based on **the newly identified genes** that, for the first time, can predict whether an otherwise healthy individual is likely to get cancer.

b. "It's a super-exciting set of discoveries," says Bert Vogelstein, a Johns Hopkins University researcher who has just found *a gene pivotal to the triggering of colon cancer*.

We use coreference data from the OntoNotes Release 3.0 corpus (Weischedel et al., 2011) to calculate mention data.

4.1.3 Syntactic features

We include five syntactic features, detailed below.

4.1.3.1 Syntactic parallelism

We compare the grammatical role (see Section 4.1.2.1) of each NP, hypothesising that NPs that play the same role in their respective sentence may be more likely to be in an instantiation. We include a binary feature which is True if the grammatical role of the two NPs is identical. This has been shown to be a useful feature in anaphora resolution (Mitkov, 1999).

4.1.3.2 Modification

The modification type includes values that represent apposition, conjunction, pre modification and bare nouns, again derived from dependency parse data. We include any of the types that apply to the NP as a feature string, from which unigrams and bigrams are drawn.

Our intuition is that set members and subsets are often more heavily modified than the sets that they are part of, as in Example 4.5.

(4.5) a. *footballers* → *Premiership footballers playing for top 4 clubs*,

b. *European countries* → *European nations that use the Euro*.

4.1.4 Context features

We include several contextual features, hypothesising that NPs that occur in similar contexts may be more likely to be entity instantiations.

4.1.4.1 Verb semantics

We hypothesise that the verb on which an NP depends provides important context. Our intuition is that NPs that depend on similar types of verbs are more likely to participate in an instantiation. We note examples such as Example 4.6 which has two similar verbs, *'surge'* and *'skyrocket'*.

(4.6) a. **Some European funds** recently have skyrocketed.

b. *Spain Fund* has surged to a startling 120% premium.

We retrieve the Levin class (Levin, 1993) of the verb on which each NP depends, as well as the verb itself. We use this verb classification as a simple method of including a measure of verb similarity. There are many more sophisticated methods of calculating verb similarity, such as Chklovski and Pantel (2004) and Yang and Powers (2006), which we would like to implement in future work.

4.1.4.2 Quotations

We calculate a binary feature which represents whether each NP is in a quotation. In our examination of the source texts we discovered a number of examples where instantiations were part of sentences involving quotations. Although, as in Example 4.7, the set member can often occur outside the quotation as an attribution, set members can and do occur within quotations. Examples 4.8 and 4.9 illustrate this.

(4.7) a. **American Express, Kraft General Foods, and Mattel executives** said the move won't affect their relationships with the ad agency.

b. "General Foods's relationships with its agencies are based on the agencies' work, and will continue to be," said *David Hurwitt, a vice president of Kraft General Foods*.

(4.8) a. Industry executives say that although **the two executives** used to clash more frequently, the WPP takeover brought **them** closer together.

b. "*I*'m the guy who made him head of New York, head of the U.S., president of North America, and recommended *him* {to Mr. Sorrell} as my successor."

(4.9) a. **Some in the industry** are skeptical.

b. "*I* find it hard to conceive of people switching over to CNN for what, at least in the public's mind, is the same news," says Reuven Frank, the former two-time president of NBC News and creator of the Huntley-Brinkley Report.

4.1.4.3 Discourse relations

We hypothesise that contextual discourse relations can aid the disambiguation of entity instantiations. In cases such as Example 4.10, the presence of the discourse connective *'however'*, which represents a contrast discourse relation, is useful in establishing that no instantiation is present.

(4.10) a. **Some workers** are opposed to strike action.

b. *David Jones*, however, is willing to put his job on the line for the cause. *(Not an instantiation.)*

As we intended to use our entity instantiation classifier to aid implicit discourse relation classification in the future (see Chapter 5), we felt it important to refrain from directly employing gold standard PDTB (Prasad et al., 2008) annotations at this stage. We instead include an approximation of the discourse relations present.

We approximate by extracting a list of the explicit connectives relation in the PDTB, along with the discourse relation that most frequently corresponds to each connective. We then search the sentence(s) which contain the two NPs for the presence of any of the connectives in our extracted lists, and include two features — a list of connectives found, and a list of their corresponding relation.

For example, the sentence pair in Example 4.11 would lead to the connective list [*'still', 'if', 'as'*], and relation list [*'comparison.contrast', 'contingency.condition', 'temporal.synchrony'*].

(4.11) a. In ending *Hungary's part of the project*, Parliament authorized Prime Minister Miklos Nemeth to modify a 1977 agreement with Czechoslovakia, which still wants the dam to be built.

b. Mr. Nemeth said in parliament that Czechoslovakia and Hungary would suffer environmental damage if **the twin dams** were built as planned.

Most explicit connectives are unambiguous in terms of the 4 main relation types — Pitler et al. (2008) induce a classifier with the explicit connective of a discourse relation as the sole feature which obtains an accuracy of 93.09%.

We instead map to the 20 relation types which comprise the second level of the hierarchy of PDTB relations (see Appendix D for full hierarchy), for which connectives are more ambiguous. However, they do provide a more useful and expressive set of discourse relations than just considering the top level.

4.1.5 Knowledge-based features

We include knowledge-based features, on the basis that world knowledge about the entities is important in establishing the presence of an instantiation. The strategy of incorporating world knowledge to improve classification is a commonly found one in several related NLP problems.

In the problem of anaphora resolution (see Section 2.3), Poesio et al. (2004), Markert and Nissim (2005) and Markert et al. (2003) are amongst those who use knowledge sources to improve classification. In relation extraction (see Section 2.1.3.1), Zhou et al. (2005) employ WordNet, Chan and Roth (2010) use Wikipedia queries and Sun et al. (2011) use large-scale word clusters to improve performance. In coreference resolution, Harabagiu et al. (2001) use WordNet relations, Daumé III and Marcu (2005) use both WordNet and lists of IS-A relations harvested from the web, Yang and Su (2007) use automatically created patterns and Ponzetto and Strube (2006) use both WordNet and features extracted from Wikipedia glosses and category listings.

Entity instantiations are often cases of hyponymy, and relating established hyponyms from knowledge sources to entity instantiations is likely to be very useful. We chose to adopt 3 particular methods for this task; WordNet, Pattern-based Hyponyms and Animacy matching.

4.1.5.1 WordNet

WordNet (Fellbaum, 1998) is a widely used lexical database, in which words are organised into sets of synonyms, referred to as synsets, and relations exist between synsets representing phenomena such as hyponyms, hypernyms, meronyms and holonyms. As a hand-created resource, WordNet has the benefit of highly accurate relations, and good coverage of common nouns, but it does not cover most named entities.

We use WordNet to establish whether the head words of NPs that are *not* named entities are synonyms or hyponyms, in an effort to identify pairs such as *'offers'* and *'bids'* in Example 4.12.

(4.12) a. **Bids totalling $515 million** were submitted.

b. *Accepted offers* ranged from 8.38% to 8.395%

4.1.5.2 Pattern-based hyponyms

We take our inspiration from Hearst (1992), in which patterns are used to extract hyponyms from corpora. In Hearst (1992), patterns such as *'<NP> and other <NP>'*, *'<NP> or other <NP>'* and *'such <NP> as <NP>'* are used to automatically harvest hyponyms from Grolier's American Academic Encyclopedia (Grolier, 1990).

In Markert and Nissim (2005), this pattern extraction is applied for finding the relatedness of NPs, with the purpose of resolving non-pronominal anaphora. Instead of using a regular corpus of English, the pattern extraction is done from the World Wide Web, using

the Google search engine to calculate hit counts. To estimate the relative likelihood of two NPs being hyponyms they use a scoring system based on Mutual Information.

We also use Google for discovering potential set membership and subset relations, and a similar Pointwise Mutual Information (PMI) type measure to indicate the strength of the relationship.

We employ the pattern "*X* and other *Y*", where *X* is a potential set member or subset and *Y* is a potential set. We use the following formula to calculate the value of our feature:

$$\text{G-PMI}(X,Y) = \frac{hits(\text{“}X \text{ and other } Y\text{”})}{hits(\text{“}X\text{”}) \times hits(\text{“and other } Y\text{”})}$$

We expand our queries to include the NE type of named entity NPs, and conjunctions and appositions of the head NP. As in Markert and Nissim (2005), when querying using the NE type, we change the query structure to reflect the fact that the NE type is a hypernym, rather than hyponym, of the set NP. The maximal value is taken, and included as a numerical feature.

4.1.5.3 Animacy

Instantiations are, by definition, almost always between a set and a subset/set member that are of the same type. For example, one organisation drawn from a set of organisations, or a subset of a set of persons. We attempt to establish whether the animacy of the two NPs match, reasoning that pairs of NPs that do not have the same animacy are not of the same type and therefore highly unlikely to participate in an Entity Instantiation.

We use a list of animate pronouns and lists of animate and inanimate words distributed as part of the Stanford Deterministic Coreference Resolution System (Ji and Lin, 2009; Lee et al., 2011), and named entity information generated by the Stanford Named Entity Recognizer (Finkel et al., 2005) to ascertain the animacy of each NP. Our feature has three possible values; Match if the two NPs have the same animacy, No Match if they do not, and Not Present if we cannot calculate the animacy of one of the NPs. Not Present occurs in only 3.30% of pairs.

4.1.6 Full feature list

Our full feature set is listed in the table below. The value *text* in the 'Feature type' column refers to those features that are presented to the learner as a list of tokens, which are then converted into unigrams and bigrams. A list of values for the 'Feature type' column denotes a restricted set of values from which the feature value must be drawn.

We also include an analysis of the numerical features, showing statistics which describe the distribution of the values each feature takes, as Table 4.2. The unbalanced, intersentential data was used to calculate these statistics.[1]

Feature name	**Feature type**	**Example Value**
surface backwards	binary	True
surface member/subset unigrams	text	the eagerness of chinese trade officials
surface set unigrams	text	foreign loans
surface unigram min edit	integer	33
surface member/subset unigram lemmas	text	the eagerness of chinese trade official
surface set unigram lemmas	text	foreign loan
surface unigram lemma min edit	integer	33
surface member/subset POS	text	dt nn in jj nn nns
surface set POS	text	jj nns
surface member/subset headword	text	eagerness
surface set headword	text	loans
surface headword min edit	integer	11
surface member/subset headword lemma	text	eagerness
surface set headword lemma	text	loan
surface headword lemma min edit	integer	13
surface member/subset head POS	text	nn
surface set head POS	text	nns
surface distance words	integer	8
surface distance chars	integer	43
surface distance words normalised	continuous	0.163265306122
surface distance chars normalised	continuous	0.153024911032
salience member/subset grammatical role	text	SBJ
salience set grammatical role	text	PMOD

[1]This data set is described in Section 4.3.

Feature name	**Feature type**	**Example Value**
salience member/subset mention count	integer	1
salience set mention count	integer	1
salience member/subset prior mention count	integer	0
salience set prior mention count	integer	0
salience member/subset local first mention	first, not-first, non-coreferent	non-coreferent
salience set local first mention	first, not-first, non-coreferent	non-coreferent
syntax grammatical role retained	False,True	False
syntax member/subset premodification	text	NMOD
syntax member/subset postmodification	text	NMOD
syntax set premodification	text	NMOD
syntax set postmodification	text	
contextual member/subset dependent verb	text	estimate
contextual member/subset dependent verb levin	text	54.4
contextual set dependent verb	text	have
contextual set dependent verb levin	text	7.7
contextual member/subset has quotation	False, True	False
contextual set has quotation	False, True	False
contextual member/subset in quotation	False, True	False
contextual set in quotation	False, True	False
contextual discourse connectives	text	when
contextual discourse relations	text, restricted to the 20 discourse relation sub-types	temporal.synchrony

Feature name	Feature type	Example Value
worldknowledge wordnet syns	False, True	False
worldknowledge wordnet hypo	False, True	False
worldknowledge wordnet compatible	False, True	False
worldknowledge google pmi	continuous	0.0
worldknowledge animacy	False, True, None	True

4.2 Intrasentential Features — Tree Kernels

We employed tree kernels — a method for learning directly from tree structures — for the classification of our intrasentential instantiations. All the features discussed so far are presented to the machine learner as a vector of features. Figure 4.1 shows an example feature vector.

> [True, mr. freeman, american express representatives, 25, mr. freeman, american express representative, 24, nnp nnp, nnp nnp nns, freeman, representatives, 19, freeman, representative, 17, nnp, nns, 29, 157, 0.644444444444, 0.564748201439, SBJ, SBJ, True, True, 3, 1, 0, 0, True, , , NMOD, , say, 2.1 37.7, influence, , False, False, False, False, , , False, False, False, 0.000895551794238, False, False, True, 5, 1, 10000.0, 0, first, non-coreferent, set_member.]

Figure 4.1: A vector of features, as presented to the ICSIBoost machine learner.

In contrast, tree kernels learn directly from structured data, in this case in the form of trees, by learning common subtrees and tree fragments (Collins and Duffy, 2002). In practical terms, using tree kernels is a two step process:

Tree representation design. This step involves deciding on the portion of the tree that is presented to the tree kernel learning algorithm, and is analogous to the design of features in traditional feature vector based machine learning. This step is necessary because it is often not optimal to present the learner with the entire tree of the sentence containing the phenomenon being identified (Bunescu and Mooney, 2005). Instead, one can present the learner with an appropriate portion of the sentence, in order to reduce noise and increase

Feature name	**Max**	**Min**	**Mean**	**Quartiles**	**Standard Deviation**
surface unigram min edit	423	0	35.15	$Q_1 = 14$, $Q_2 = 23$, $Q_3 = 44$	34.12
surface unigram lemma min edit	404	0	34.15	$Q_1 = 13$, $Q_2 = 22$, $Q_3 = 42$	33.20
surface headword min edit	35	0	9.96	$Q_1 = 7$, $Q_2 = 9$, $Q_3 = 12$	4.04
surface headword lemma min edit	35	0	9.48	$Q_1 = 7$, $Q_2 = 9$, $Q_3 = 11$	4.04
surface distance words	126	0	25.35	$Q_1 = 14$, $Q_2 = 23$, $Q_3 = 34$	15.97
surface distance chars	697	1	136.53	$Q_1 = 70$, $Q_2 = 124$, $Q_3 = 187$	89.42
surface distance words normalised	0.96	0.00	0.42	$Q_1 = 0.25$, $Q_2 = 0.42$, $Q_3 = 0.58$	0.22
surface distance chars normalised	0.98	0.00	0.42	$Q_1 = 0.25$, $Q_2 = 0.42$, $Q_3 = 0.59$	0.23
salience member/subset mention count	216	1	6.41	$Q_1 = 1$, $Q_2 = 1$, $Q_3 = 1$	21.51
salience set mention count	72	1	1.95	$Q_1 = 1$, $Q_2 = 1$, $Q_3 = 1$	4.20
salience member/subset prior mention count	211	0	2.67	$Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$	12.31
salience set prior mention count	69	0	0.51	$Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$	2.81
worldknowledge google pmi	915.72	0.00	0.07	$Q_1 = 0.0$, $Q_2 = 0.0$, $Q_3 = 0.0$	4.57

Table 4.2: Statistical analysis of the numerical features included in the full feature set.

the likelihood that useful patterns are identified. We discuss possible representations and their use in other problems, and also outline the representation we use, in Section 4.2.1.

Tree kernel choice. A number of methods of comparing the trees and searching for common subtrees are available. This step is analogous to the selection of an appropriate machine learning algorithm in traditional feature vector based machine learning. In Section 4.2.2, we discuss these algorithms, showing examples of the types of subtrees extracted by them.

The use of tree kernels is common in RE, with Zelenko et al. (2003), Culotta and Sorensen (2004), Bunescu and Mooney (2005), Zhang et al. (2006), Zhou et al. (2007) and Swampillai and Stevenson (2011) all applying variants of this technique. Whilst some information about entity instantiations — or indeed any other phenomenon — are best represented in a feature vector form, such as the number of prior mentions or the Levin verb class, it can be difficult to adequately capture information about the syntactic structure in a feature vector format. The use of tree kernels to learn directly from parse trees allows us to instead capture similarity between the syntactic structures of the phenomenon directly (Grishman, 2012).

We used this approach solely for the intrasentential instantiations because syntactic parse trees cover only a single sentence. This means that intersentential instantiations are *not* covered as part of a single structure, and some mechanism (such as merging the trees of two adjacent sentences under a single root node, as in Swampillai and Stevenson (2011)) must be used to create a single, artificial structure. Although Swampillai and Stevenson (2011) have some success with this approach, we find this artificial merging of sentences to be both theoretically dubious and likely to introduce noise, and therefore avoid it.

We also hypothesised, based on our observations in Section 3.8.4, that syntactic patterns play a stronger role in the recognition of intrasentential instantiations than intersentential instantiations.

4.2.1 Tree representation

In using tree kernels, we seek to automatically discover pertinent patterns in constituency parse trees that make it more or less likely that an instantiation is present. We know both NPs are contained within a single sentence, and therefore the simplest way to do this is to pass the tree kernel learner the parse tree of the entire containing sentence, without pre-processing it in any way. However, if one considers an example sentence such as Figure 4.2, which contains an instantiation between the two NPs indicated with

circles, there is a great deal of syntactic information that is likely to be unrelated to the phenomenon — and is therefore noise. To encourage the learner to discover pertinent patterns, and to prevent the learner having to consider patterns in the sentence parse that are distant from the phenomenon and unlikely to be helpful, we present our learner with a smaller portion of the containing sentence.

This technique of presenting a tree kernel learner with a fragment of a sentence rather than a whole sentence was shown to be effective for RE in Bunescu and Mooney (2005), albeit with a dependency, rather than a constituency parse tree. The fragment of the sentence that they present to the learner is the shortest dependency path between the two entities. Subsequent work in RE using constituency parses, including Zhang et al. (2006), Zhou et al. (2007), Jiang and Zhai (2007) and Swampillai and Stevenson (2011), has chosen to present sentence fragments based around the shortest path between the two entities to the learner, rather than whole sentences, and achieved good results with this technique.

In particular, Zhang et al. (2006) experiment with five different sentence fragments, with varying degrees of included context. The description of these five representations is reproduced below. We also include the diagrams used to illustrate these representations as Figure 4.3.

> **Minimum Complete Tree (MCT):** the complete sub-tree rooted by the nearest common ancestor of the two entities under consideration.
>
> **Path-enclosed Tree (PT):** the smallest common sub-tree including the two entities. In other words, the sub-tree is enclosed by the shortest path linking the two entities in the parse tree (this path is also commonly-used as the path tree feature in the feature-based methods).
>
> **Context-Sensitive Path Tree (CPT):** the PT extended with the 1st left word of entity 1 and the 1st right word of entity 2.
>
> **Flattened Path-enclosed Tree (FPT):** the PT with the single in and out arcs of non-terminal nodes (except POS nodes) removed.
>
> **Flattened CPT (FCPT):** the CPT with the single in and out arcs of non-terminal nodes (except POS nodes) removed.
>
> (Zhang et al. (2006))

They experimented with the trees on the ACE 2003 data set[2], and achieved the results

[2]See Section 2.1.3 for a discussion of the ACE RE data sets.

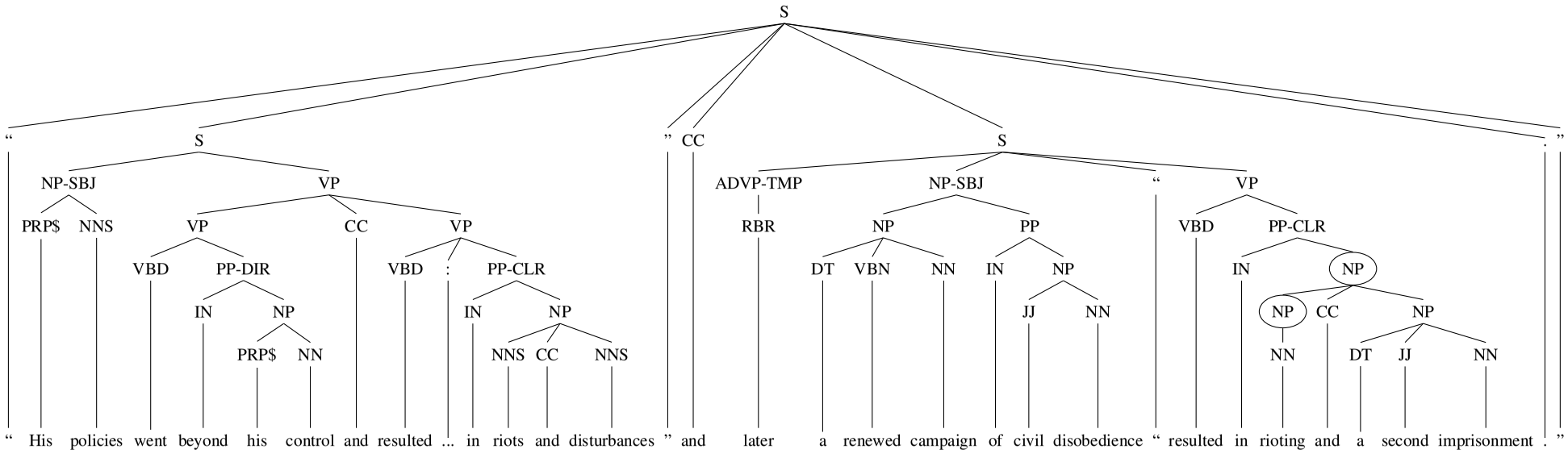


Figure 4.2: An example of large sentence parse tree with a relatively local entity instantiation.

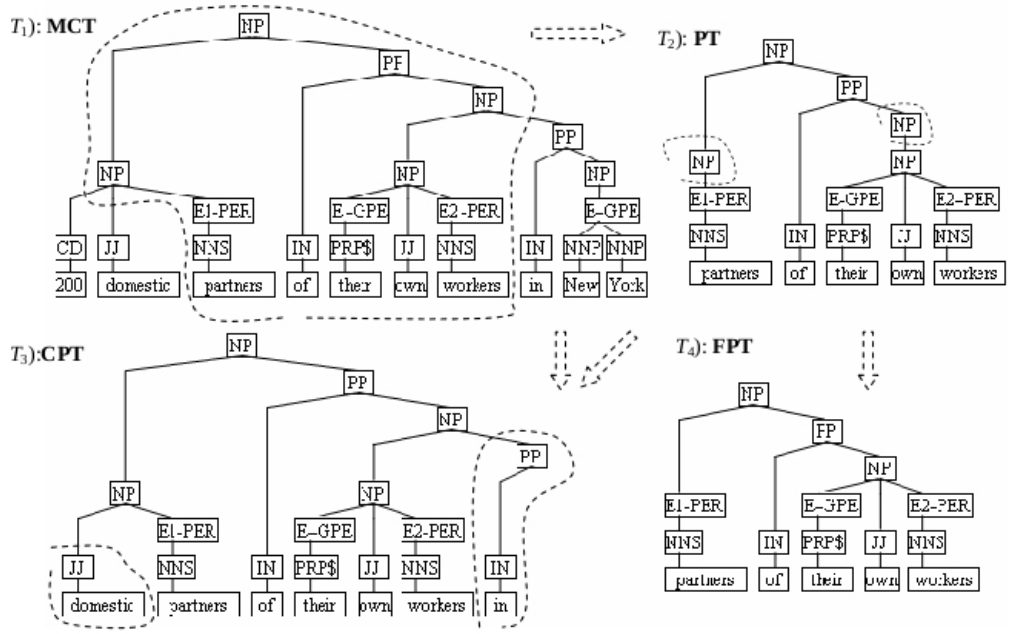


Figure 4.3: The five tree representations used in Zhang et al. (2006), based on the sentence "...provide benefits to 200 domestic partners of their own workers in New York". "Partners" and "workers" are the two entities in question.

Tree representation	Precision (%)	Recall (%)	F-Score
Minimum Complete Tree (MCT)	77.5	38.4	51.3
Path-enclosed Tree (PT)	72.8	53.8	61.9
Context-Sensitive PT (CPT)	75.9	48.6	59.2
Flattened PT	72.7	51.7	60.4
Flattened CPT	76.1	47.2	58.2

Table 4.3: Results from Zhang et al. (2006).

shown in Table 4.3. Based on these results, they observe that the MCT, which has the most contextual data, performs substantially worse than any of the other representations. In contrast, the PT, with the least contextual information, performs best. The fact that the MCT is high in precision, but low in recall, leads them to suggest that the extra contextual data leads to overfitting. They also note that flattening the trees — removing non-terminal nodes with a single in and single out arc — decreases performance, and therefore non-terminals provide useful information.

Further to this, Swampillai and Stevenson (2011) experimented with a more minimal representation, the Shortest Path Tree (SPT). Given two entities, e_1 and e_2, along with their nearest common ancestor, C, the SPT is the conjunction of the shortest path from e_1 to C and the shortest path from e_2 to C. This is illustrated in Figure 4.4, using the same

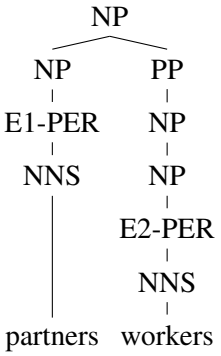


Figure 4.4: The SPT representation of the sentence from Figure 4.3.

example sentence as in Figure 4.3. They used both the SPT and the PT, which they refer to as the Shortest Path Enclosed Tree (SPET)[3], and find that on their data set, the SPT performs better.

We also note that Swampillai and Stevenson (2011), in contrast to Zhang et al. (2006), do not appear to add labels to the tree to indicate the location of the entities under consideration. Zhang et al. (2006) use the labels `E1` and `E2`, appended with the entity type (e.g. `-PER` for person) to explicitly mark the entities.

Whilst there are differences between the RE task and that of identifying entity instantiations, which are outlined in Section 2.1.3.4, we feel that identifying *intrasentential* entity instantiations is closely related to RE, and therefore the conclusions drawn in the RE literature are likely to be applicable to our problem. In particular, we demonstrated in Section 3.8.3 that the majority of intrasentential instantiations annotated consist of examples where the two NPs are nested in some fashion, and therefore likely to be proximate.

Therefore, we used the two tree representations that perform best in Zhang et al. (2006) and Swampillai and Stevenson (2011) — SPET and SPT. In our implementation we opted to follow Zhang et al. (2006) and explicitly indicate the NPs under consideration. As the considered items are always NPs[4], which is not the case in RE, we replace the node label of the subtree that represents the set member/subset NP with the node TREE1, and the node label of the subtree that represents the set NP with TREE2, rather than introducing extra nodes to the tree.

For clarity, we repeat the definitions and show further examples using entity instan-

[3]Henceforth, we use the term SPET rather than PT. They are, however, identical.

[4]Whilst all the considered items are NPs, the PTB does attach suffixes to the labels of certain types of NP, such as `-SBJ` for those NPs that are subjects, or `-TMP` for temporal NPs. Although these specific labels are lost in this process, the effects are mitigated by the fact that our flat features capture the grammatical role of each NP. We also note that some of these suffixes are reserved for NPs that are removed in our annotation pre-processing because they cannot be mentions, such as `NP-PRD` which represents predicate NPs.

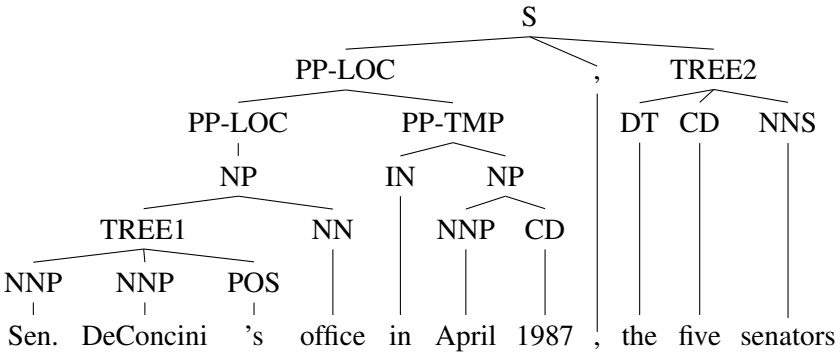


Figure 4.5: The SPET representation derived from Example 4.13.

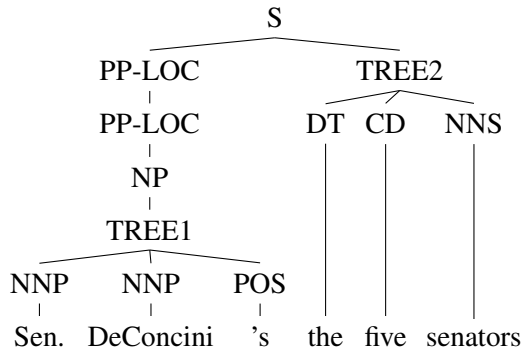


Figure 4.6: The SPT representation derived from Example 4.13.

tiations. The SPET is the shortest path between the two NPs, inclusive of all nodes in between. SPT is the the shortest path between the two NPs, exclusive of all nodes in between. Example 4.13 shows an example sentence with two NPs underlined. Figure 4.5 shows the SPET that connects them, and Figure 4.6 shows the SPT tree that connects them.

(4.13) In a highly unusual meeting in Sen. DeConcini's office in April 1987, the five senators asked federal regulators to ease up on Lincoln.

A point not explicitly considered in the RE literature is the inclusion of the leaf nodes, which represent the words. In a situation where we are trying to avoid noise and overfitting, most of the particular words used are unlikely to be repeated in another example, and are therefore unhelpful as training data. On the other hand, some words belonging to closed classes may occur commonly enough to to provide useful training data. We therefore experimented with two variations in the lexicalisation of these trees; full delexicalisation, in which all leaf nodes are removed, and partial delexicalisation, in which terminal nodes representing nouns are removed.

Henceforth, the abbreviations P and F are appended to SPT or SPET to represent the lexicalisation used. SPTF corresponds to 'SPT, fully delexicalised', SPETP corresponds to 'SPET, partially delexicalised', and so on.

4.2.2 Tree kernel algorithms

In the previous Section we considered the tree fragment that would be passed to the tree kernel learner. In this Section, we instead focus on how tree kernels actually work, and the sort of patterns they learn.

A *kernel function* is a method of measuring the similarity between a pair of objects (Grishman, 2012). In feature vector based machine learning, a kernel function measures the similarity between the feature vectors, but machine learning algorithms such as Support Vector Machines (SVMs) allow for the creation of specialised kernels to measure similarity between objects other than feature vectors. For example, in addition to *tree kernels* which measure the similarity of tree structures (Vishwanathan and Smola, 2002; Collins and Duffy, 2002; Moschitti, 2006a), *subsequence* kernels have been created to measure similarity between strings of text (Lodhi et al., 2002; Bunescu and Mooney, 2006).

At least three tree kernel algorithms have been postulated. All measure similarity by counting the number of common substructures present between two trees, but differ in the type of internal substructures considered:

Subtree kernel (ST). The ST counts common *subtrees* to measure similarity (Vishwanathan and Smola, 2002). These subtrees must contain all the descendants of a given node, down to the leaves (Moschitti, 2006a).

Subset tree kernel (SST). In contrast to the ST, the SST can make use of internal subtrees, which do not include the leaf nodes. However, each node in the subset tree must either have *none* or *all* its immediate children included. In Example 4.7(b), `(VP (V NP))` would be a valid subset tree, `(VP (V))` would not (Collins and Duffy, 2002; Moschitti, 2006a).

Partial tree kernel (PT). The PT further relaxes the constraints on the internal substructures considered, allowing for the generation of substructures that correspond to partial grammar rules. In other words, the tree `(VP (V))`, which is disallowed by the SST, is a valid substructure in the PT (Moschitti, 2006a).

Examples of these three tree kernels, reproduced from Moschitti (2006a), are shown in Figure 4.7.

In our experiments we use the SST, for a number of reasons. Firstly, Moschitti (2006a) compare the performance of these three tree kernels in three different problems: classifying predicate-argument structures in FrameNet, classifying predicate-argument structures in PropBank and in the task of Question Classification, in which questions are classified into one of 6 coarse-grained classes that represent the type of response required to the question. In all these experiments the ST performs worst, and the SST generally outperforms the PT, especially when using constituency rather than dependency trees.

Secondly, the SST has been applied to RE successfully by both Zhang et al. (2006) and Zhou et al. (2007)[5]. Thirdly, an implementation of the SST is available as part of SVM-LIGHT-TK (Moschitti, 2006b; Joachims, 1999), but an implementation of the PT is not part of this toolkit. SVM-LIGHT-TK also has the useful capability of allowing us to combine tree kernels with traditional unstructured feature kernels.

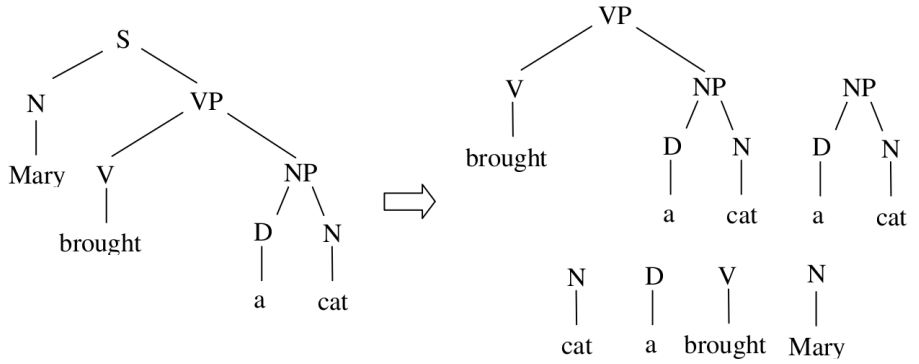
4.2.3 Application of tree kernels to intrasentential entity instantiations

Given our choice of tree representation and tree kernel algorithm, we highlight some common syntactic patterns within intrasentential entity instantiations that we hope the tree kernels will capture. Of course, part of the attraction of tree kernels is that they are also likely to identify useful common substructures outwith the ones we have identified and detail here, and so this list is not exhaustive, but instead serves as an indication of the motivation behind their use.

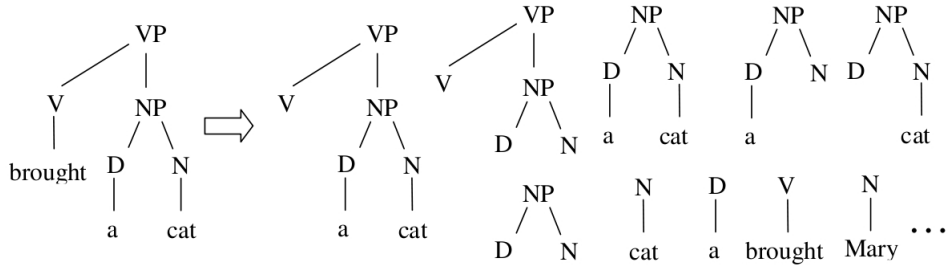
Simple conjunctive phrases. A number of intrasentential entity instantiations occur as part of conjunctive phrases, in which the entire phrase represents a set, and each of the nested NPs is a member or subset of that set. Figure 4.8 shows two examples of this type of phrase. Our expectation is that the common occurrences of subset trees indicating conjunctive phrases, such as `(TREE2 (TREE1 CC NP))`, `(TREE2 (NP CC TREE1))` and `(TREE2 (NP , NP CC TREE1))` will lead to the identification of entity instantiations.

Noun phrases with prepositional phrases. Another common syntactic pattern in intrasentential entity instantiations is the introduction of a list of members/subsets as part of

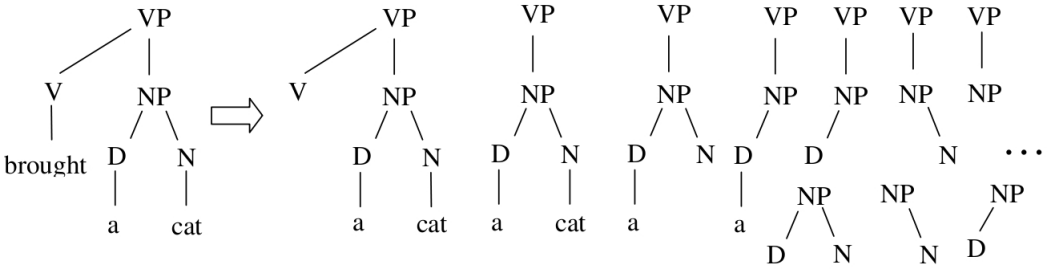
[5]Zhou et al. (2007) extend the SST to consider ancestral contextual information. However, they do still rely on subset trees.



(a) A syntactic parse tree and its subtrees (STs).



(b) A syntactic parse tree and some of its subset trees (SSTs).



(c) A syntactic parse tree and some of its partial trees (PTs).

Figure 4.7: Examples of the substructures extracted and compared by three tree kernel algorithms: the ST, SST and PT, from Moschitti (2006a).

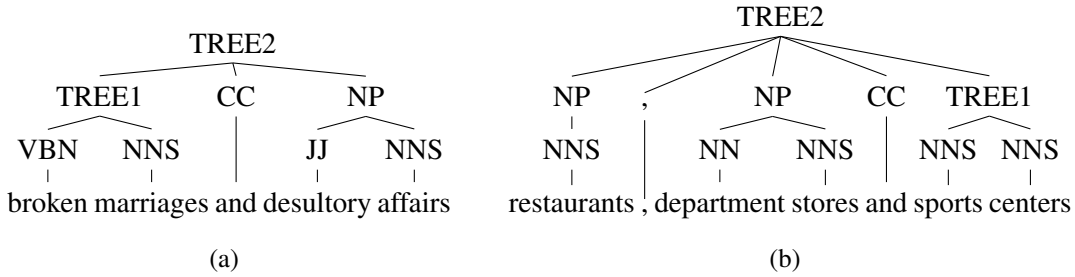


Figure 4.8: Example constituency parse trees of simple conjunctive phrases.

a prepositional phrase, typically *'such as'* or *'like'*. Figure 4.9 shows three examples that follow this pattern. The tree kernel based learner should identify common subset trees such as `(TREE2 (NP PP))`, `(TREE2 NP (PP IN (NP NP CC TREE1)))` and `(PP (IN like) NP)` in these cases.

Nested sets. A third syntactic pattern occurs in situations where the set is nested within the member, and usually takes the form *'X of the Y'*, where *Y* is the set. Figure 4.10 shows three examples of this type of phrase. The subset tree `(TREE1 NP (PP IN TREE2))`, which is common to all three examples, should be learned by the tree kernel, along with the subset tree `(PP (IN of) TREE2)`.

We believe that syntactic relationship alone is not sufficient to identify all intrasentential entity instantiations. For example, the tree in Figure 4.11 is very similar syntactically to the trees in Figure 4.9 — the difference being the use of *'to'* rather than *'like'* or *'such as'*. As such, a tree kernel learner may mistakenly classify this as an entity instantiation. However, the lexical knowledge that *'to'* is unlikely to be an indicator of an entity instantiation, which is captured by our flat unigram features, would help in this instance. Additionally, knowing that *'Boston'* is a location, and is unlikely to be related to *'good extensions'* could also helpful in disambiguating the instantiation. In cases such as these, we suggest that both tree kernels and flat features have a role to play in classification.

Similarly, in situations where the NPs under consideration are distant, it seems unlikely that there are common subset trees that would make instantiations more or less likely. Also, in cases where the NPs are distant, the SPT and SPET are necessarily larger, and therefore introduce further noise. For example, in Figure 4.12, the SPET linking the two NPs exhibits a complex syntactic structure, and it is difficult to see any patterns that could be learned from such an example.

In summary, we employ tree kernels to automatically identify common patterns within

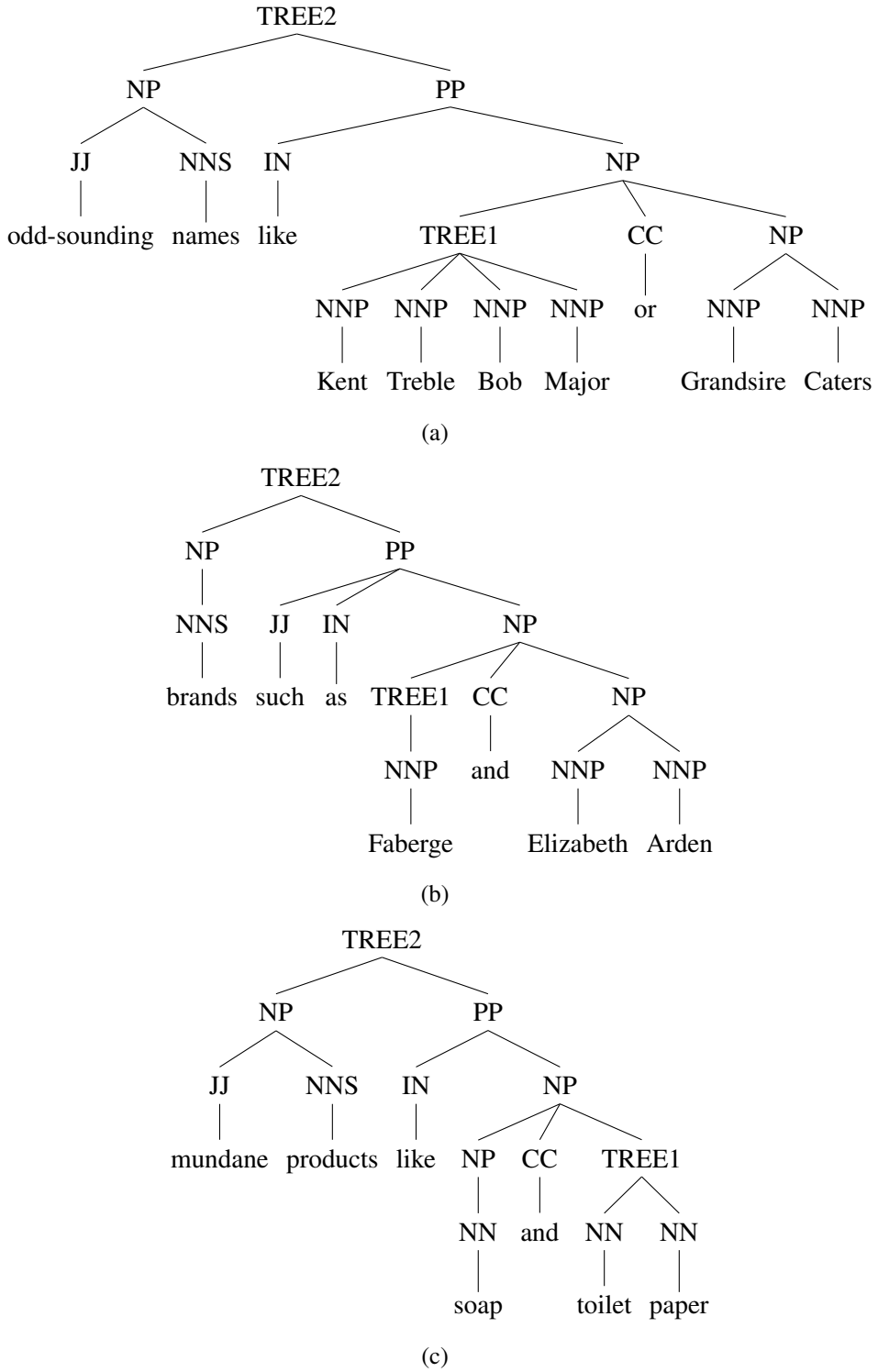
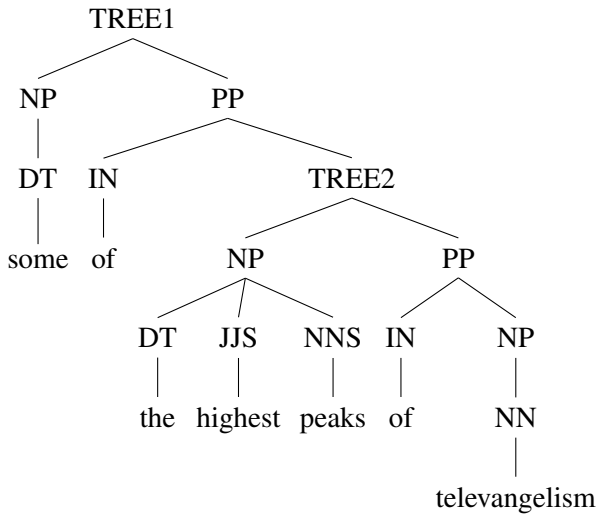
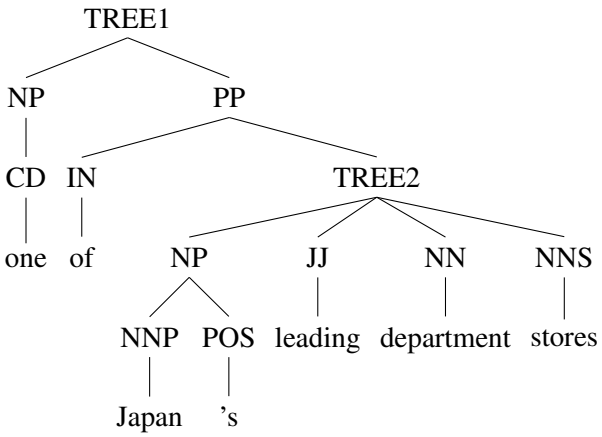


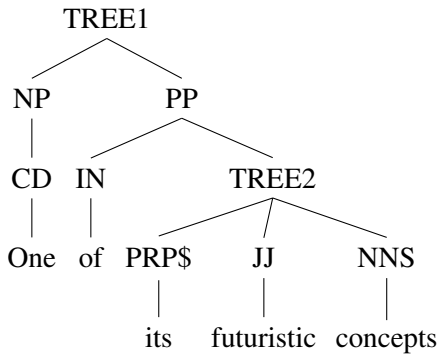
Figure 4.9: Example constituency parse trees comprising noun phrases with prepositional phrases.



(a)



(b)



(c)

Figure 4.10: Example constituency parse trees of nested sets.

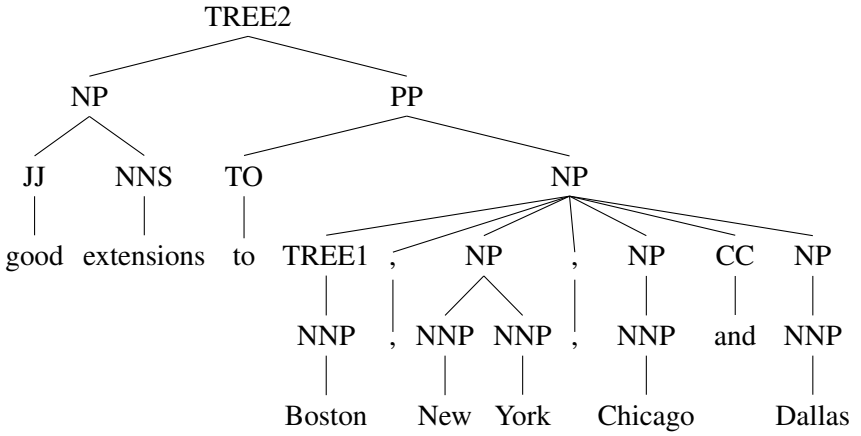


Figure 4.11: A tree similar to those in Figure 4.9, but that does not represent an entity instantiation.

intrasentential entity instantiations. Our intuition is that many intrasentential entity instantiations follow common syntactic patterns, making tree kernels suitable. Rather than learning from trees of the whole sentence containing the instantiation, we choose to follow the methods employed by a variety of research in RE, in which a fragment of the sentence tree is instead extracted and presented to the learner. The tree kernel learns common substructures within the trees presented to it, which means explicitly designing a multitude of syntactic features is unnecessary. However, we note that syntax is not the sole factor for deciding the presence of an entity instantiation, and we intend to combine our tree kernels with the flat features outlined in Section 4.1.

4.3 Experimental Setup

For these experiments, we consider the problem of identifying set membership separately to that of identifying subsets. We therefore divide our data set into two; plural-plural noun phrase pairs that are labelled either *subset* or *no-instantiation* and plural-singular noun phrase pairs that are labelled either *set member* or *no-instantiation*. We use the same feature set (i.e. the one described in Section 4.1) for both.

We also considered the problems of intersentential and intrasentential instantiations separately. Our reasoning was that intrasentential instantiations were a sufficiently different phenomenon, that occurred in patterns not found in intersentential instantiations, such as those described in Section 3.8.4. Our intuition was that syntax played a stronger role in identifying intrasentential instantiations, hence the development of the tree kernel methods described in Section 4.2.

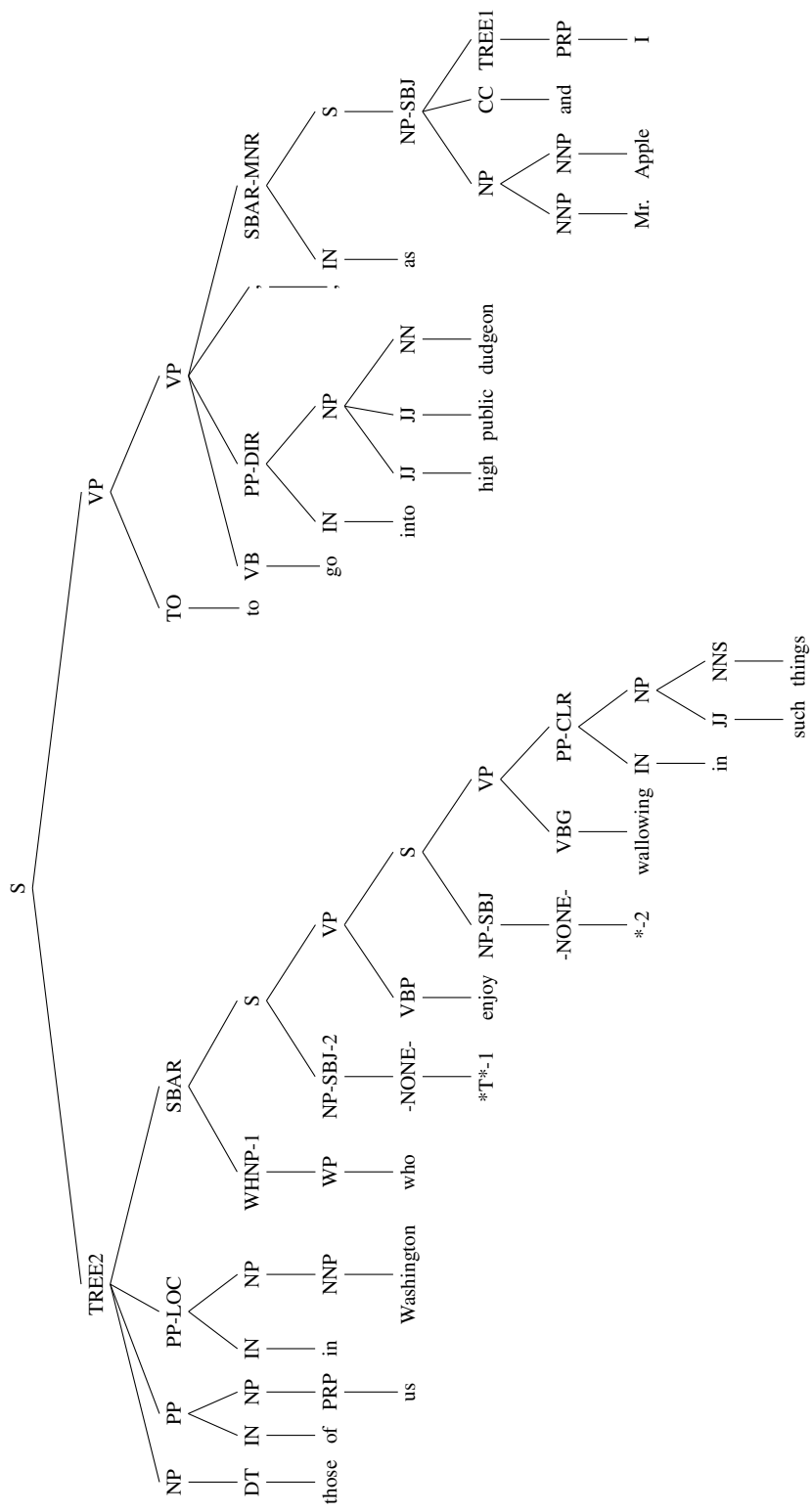


Figure 4.12: An example of a distant entity instantiation in a syntactically complex sentence, which is unlikely to be identified by tree kernel learning.

Set member/Subset	**Balanced**	**# of NP pairs**	**# Positive (%)**
Set member	Unbalanced	25901	1538 (5.9%)
Subset	Unbalanced	17893	865 (4.8%)
Set member	Balanced	3076	1538 (50%)
Subset	Balanced	1730	865 (50%)

Table 4.4: The size and distribution of intrasentential machine learning data sets.

Set member/Subset	**Balanced**	**# of NP pairs**	**% Positive**
Set member	Unbalanced	47605	1477 (3.1%)
Subset	Unbalanced	30434	641 (2.1%)
Set member	Balanced	2954	1477 (50%)
Subset	Balanced	1282	641 (50%)

Table 4.5: The size and distribution of intersentential machine learning data sets.

Due to the nature of the annotation study, there are many more pairs of candidates between which no entity instantiation has been annotated than those that have. As discussed in Chapter 3, only 3.7% of the 121,833 pairs of candidates in the corpus have a set member or subset annotation. Considering this heavy skew, we experimented with *balanced* data sets — i.e. data sets in which the numbers of entity instantiations and non-instantiations are equal — as well as the original data sets. To produce these data sets, we used random sub-sampling. These sets were then used for *both* training and testing.

Tables 4.4 and 4.5 show the size and distribution of the intra- and intersentential data sets respectively.

We apply 10-fold cross-validation for testing and training in all our experiments. We also keep pairs of NPs from the same text in the same fold, to avoid over-training based on specific topical unigrams that may occur in a single text.

Initially, we use the machine learner ICSIBoost (Favre et al., 2007). ICSIBoost is an open source implementation of Boostexter (Schapire and Singer, 2000), an algorithm which combines simple 'rules-of-thumb' — in this case, decision stumps — to produce a classifier. We chose this algorithm on the basis that it has been applied successfully to a wide variety of NLP problems, including sentiment analysis (Wilson et al., 2009, 2004), discourse chunking (Sporleder and Lapata, 2005), paragraph and sentence segmentation (Cuendet et al., 2007; Favre et al., 2008; Sporleder and Lapata, 2006) and dialog act segmentation (Kolář, 2008).

ICSIBoost allows us to specify several options. For our experiments, we specify 500 rounds of boosting and use the "ngram" expert with a window size of two — i.e. all textual features are included as unigrams and bigrams.

		Gold Standard	
		True	**False**
Classifier Prediction	**Positive**	True Positive	False Positive
	Negative	False Negative	True Negative

Table 4.6: Confusion matrix

To include tree features for intrasentential learning we used SVM-LIGHT-TK (Moschitti, 2006b), an extension to SVM^{light}(Joachims, 1999). We include evaluation of intrasentential flat features using SVM^{light} as well as ICSIBoost. Combined evaluation is done using SVM^{light} alone.

4.4 Evaluation

4.4.1 Evaluation measures

We employ several evaluation measures to assess the performance of our classifiers, all of which are standard measures used in the NLP field. We measure our performance against the gold standard annotations created in Chapter 3. As shown in Table 4.6 there are 4 possibilities in our binary classification problem:

True positive The classifier correctly predicts an instantiation. (TP)

False positive The classifier incorrectly predicts an instantiation when one is not present. (FP)

False negative The classifier fails to predict an instantiation where one exists. (FN)

True negative The classifier correctly predicts that no instantiation is present. (TN)

The first measure we use is classification accuracy:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$$

Accuracy is a measure that tells us what fraction of the data set was classified correctly. However, it can be misleading for data sets with heavy skew. For example, in a situation with a data set comprised of 5% positive examples and 95% negative examples a classifier which solely predicted negatively would score an accuracy of 95%. This high accuracy does not reflect the performance of the classifier on the phenomenon we are actually interested in learning about; the positive examples.

We also employ precision, recall and F-score:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

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

Precision represents the fraction of positive predictions that are true positives. Recall is the fraction of true positives that are identified by the classifier. Both measures are very useful, but can be misleading in certain contexts. A classifier that only once predicts positive in a data set of 1000, and on that one prediction is correct would have a precision of 1, but be almost useless. A classifier that always predicts positive will have a recall of 1, but also be almost useless.

F-score is the harmonic mean of these two measures, which gives a reflection of the balance between precision and recall exhibited by the classifier, and a better idea of overall performance than either measure alone.

We calculate the Receiver Operating Characteristic (ROC) curve, and the area under this curve (AUC). The ROC is based upon the idea that our classifier outputs a numeric value for each prediction, such as a confidence score or probability, and that we may use this score to vary the threshold at which predictions are considered positive. In Table 4.7 we show an example of a variable threshold and the classifications of instances at each threshold.

The ROC is a plot of True Positive Rate against False Positive Rate as the classification threshold is varied across its range. True Positive Rate is identical to recall. False Positive Rate is defined below:

$$False\ Positive\ Rate = \frac{FP}{FP+TN}$$

The plot of the graph $y = x$ is often displayed on ROC curves as it represents the line generated by a classifier making random guesses. A perfect classifier would be represented on the graph at the co-ordinates $(0,1)$ and the worst possible classifier would be at $(1,0)$). Classifiers represented by curves above $y = x$ are useful, and the closer they pass to $(1,0)$ the better they are. Classifiers represented by curves below $y = x$ are unhelpful.

The AUC gives a useful single numerical value to compare the ROC performance of classifiers. We calculate ROC graphs and AUC using the algorithms described in Fawcett (2006).

		Label with Threshold at:		
Instance No.	**Probability**	**0.8**	**0.7**	**0.6**
1	0.95	P	P	P
2	0.85	P	P	P
3	0.75	N	P	P
4	0.65	N	N	P
5	0.63	N	N	P
6	0.59	N	N	N
7	0.50	N	N	N

Table 4.7: Classification with changing thresholds

For the highly skewed data sets we also include Balanced Error Rate (BER), which is the average of the error rate over the two classes (positive and negative). This gives us a more accurate reflection of the error rate. As it is a form of error rate, smaller numbers indicate better performance.

$$\textit{Balanced Error Rate} = \frac{1}{2} \times \left(\frac{FP}{TP+FP} + \frac{FN}{TN+FN} \right)$$

By calculating all of these measures for each of our classifiers we hope to give a representative picture of their performance, and a clear appraisal of their strengths and weaknesses.

In our discussions of results on the balanced data sets, where data imbalance is not an issue, we focus on the *accuracy* of the classifiers. When discussing performance on the original, unbalanced data sets, where accuracy can be misleading, we also note the *precision, recall* and *F-score* of the classifiers.

We also note at this point that Chapter 5 details an application of our entity instantiation classifier — that of classifying discourse relations. This forms a further, extrinsic, evaluation of our classifier.

Included with all classification results is a baseline algorithm called *Unigram*. This algorithm, as the name suggests, is a learner that solely uses the 2 features that represent the unigrams of each NP for classification. We also include a baseline called *Majority*, which is simply the prediction of the majority class for all instances. These baselines are intended to give a reflection of how our approaches compare to these much simpler algorithms, and to represent how difficult the task is.

To directly compare the performance of algorithms we test the statistical significance of the difference between them, using McNemar's χ^2 test (McNemar, 1947). This calculates the probability that the differences in classification output of the algorithm are due to chance — the lower the probability level, the more significant the difference. In

Sections 4.4.2 and 4.4.3, when we use the word *significant* we mean a difference in performance between two algorithms for which the probability level is less than 0.05.[6]

4.4.2 Intrasentential evaluation

4.4.2.1 Evaluation of flat feature performance on balanced data set

Table 4.8 shows the results of our intrasentential classifier on the balanced data set (see Section 4.3 for explanation of balanced data set creation).

Set members. Our full flat feature set (i.e. *excluding* tree features), on the set member data set, scores an accuracy of 87.6% using ICSIBoost, a significant increase in accuracy of 9 percentage points over the Unigram baseline, and 37 percentage points over the Majority baseline. Our classifier exhibits high levels of precision (0.933), as well as very good recall (0.810).

We experimented with linear and polynomial kernel options for flat features in SVM^{light}. The linear kernel performed significantly better than the polynomial kernel, with an increase in accuracy of 3.9%. The polynomial kernel gives a higher precision classifier, but at the cost of a large drop in recall. On the basis of this, all our further experiments with flat features in SVM^{light} were performed with the linear kernel. Both the linear and polynomial kernels were significantly better than the two baseline algorithms.

We can also see that the SVM classifier has a slightly lower accuracy than the ICSIBoost classifier on this problem, by 0.4% — this difference is not significant however. This pattern is repeated across our intrasentential experiments; ICSIBoost has a slight, but usually not statistically significant, edge for flat features.

To further investigate the utility of our features, we performed a feature ablation. A feature ablation is a study in which features, or groups of features in this case, are systematically removed from the feature set, in order to ascertain the contribution of the omitted features to the overall performance. We removed each feature group in turn, using both ICSIBoost and SVM^{light}, and the results are shown in Table 4.8.

We first consider the ICSIBoost ablation. Firstly, we note that removing our Surface feature group lowers accuracy significantly, showing that our surface features have an impact on classification, and that the words, lemmas and POS tag features included in this group are predictive of instantiations.

Removing the Salience feature group gives a significant reduction in accuracy of 3.1%. This increase demonstrates that knowledge of the salience of an entity in a text is helpful

[6] In addition, the detailed tables of results show two levels of significance, $p < 0.05$ and $p < 0.001$.

in establishing the presence of a set member instantiation.

Omission of the Contextual feature group gives a rise in performance, though not significantly. Our hypothesis that contextual features aid performance remains unproven for intrasentential instantiations. One reason that may explain this is that our discourse relation feature, which approximates the presence of discourse relations by identifying possible connectives and mapping them to their most likely relation, is not a strong enough approximation of discourse context. We took the decision to use this approximation based on our intention to use our classifier as part of a discourse relation classification algorithm. Bearing in mind this future application and therefore deciding against using gold standard discourse relation annotation may have had a detrimental effect on this feature.

The removal of either Syntax or World Knowledge features had little effect on performance. The lack of success of the Syntax feature group suggests that identical grammatical roles of NPs and modification level is not predictive of entity instantiations in this case. Our success with tree kernel features, discussed below, suggests that syntax does play an important role in the discovery of intrasentential instantiations, and it seems that our choice of flat syntax features were not useful.

One reason why our World Knowledge feature group had little effect was that our WordNet features had quite low hit rates. This was to be expected — WordNet deals with mostly common nouns and we had many named entities that would not appear.

The SVM classifier produces results on the data which show a similar trend, with one notable exception — in this case our World Knowledge features do make a difference and removing them leads to a significant reduction in accuracy of 0.7%.

Subsets. On the subset data set, our full feature set scores an accuracy of 84.5%, improving accuracy significantly over the Majority and Unigram baselines by 34.5% and 6.8% respectively. We note that the subset problem appears harder on the balanced data — a reduction in accuracy of between 1% and 4% occurs for the Unigram baseline and all algorithms which use the full feature set.

In our subset feature ablation, only the removal of Surface features gives a statistically significant drop in performance. This may be partially due to the fact that the subset data set is smaller, making statistically significant differences hard to attain.

4.4.2.2 Evaluation of tree kernel performance on balanced data set

To ascertain the utility of our tree kernels on this data set, we experimented with various combinations of the 4 kernels. The results are shown in Table 4.9.

	Set Members					Subsets				
ICSIBoost, Flat Features										
Feature set	**Accuracy**	**P**	**R**	**F**	**AUC**	**Accuracy**	**P**	**R**	**F**	**AUC**
Majority	50.0%	—	—	—	—	50.0%	—	—	—	—
Unigrams	78.5%$^{\beta}$	0.796	0.767	0.781	0.831	77.7%$^{\beta}$	0.795	0.743	0.768	0.805
All	87.6%	0.933	0.810	0.867	0.935	84.5%$^{\alpha}$	0.895	0.780	0.834	0.904
All - Surface	83.5%$^{\beta}$	0.866	0.792	0.827	0.905	78.6%$^{\beta}$	0.810	0.748	0.778	0.856
All - Salience	84.5%$^{\beta}$	0.898	0.777	0.833	0.905	83.4%	0.870	0.784	0.825	0.886
All - Syntax	87.5%	0.939	0.803	0.866	0.932	84.5%	0.882	0.795	0.836	0.899
All - Contextual	87.8%$^{\alpha}$	0.932	0.816	0.870	0.931	84.5%	0.897	0.778	0.833	0.895
All - World Knowledge	87.6%	0.934	0.809	0.867	0.929	84.5%	0.893	0.783	0.834	0.901
SVM, Flat Features										
All features, Linear	87.2%	0.910	0.826	0.866	0.939	84.3%$^{\zeta}$	0.888	0.786	0.834	0.898
All features, Polynomial	83.3%$^{\eta}$	0.922	0.728	0.814	0.927	79.5%$^{\eta}$	0.957	0.618	0.751	0.898
All features - Surface (Lin)	85.4% $^{\eta}$	0.892	0.807	0.847	0.909	79.5%$^{\eta}$	0.831	0.742	0.784	0.873
All features - Salience (Lin)	84.9%$^{\eta}$	0.890	0.796	0.840	0.912	83.6%	0.880	0.778	0.826	0.896
All features - Syntax (Lin)	87.6%$^{\zeta}$	0.913	0.830	0.870	0.937	83.9%	0.875	0.791	0.831	0.911
All features - Contextual (Lin)	87.4%	0.915	0.824	0.867	0.939	84.0%	0.899	0.768	0.828	0.904
All features - World Knowledge (Lin)	86.5%$^{\theta}$	0.899	0.822	0.859	0.927	84.2%	0.883	0.788	0.833	0.906

Table 4.8: Results of the intrasentential classifier on the balanced data set — flat features

$^{\alpha}$ ICSIBoost Algorithm with highest accuracy
$^{\beta}$ Significantly *worse* than $^{\alpha}$, significance $p < 0.001$, McNemar's χ^2 test.
$^{\zeta}$ SVM flat-feature algorithm with highest accuracy
$^{\eta}$ Significantly *worse* than $^{\zeta}$, significance $p < 0.001$, McNemar's χ^2 test.
$^{\theta}$ Significantly *worse* than $^{\zeta}$, significance $p < 0.05$, McNemar's χ^2 test.
$^{\iota}$ Significantly *better* than $^{\gamma}$ ($p < 0.001$) and $^{\zeta}$ ($p < 0.05$)
$^{\kappa}$ Significantly *better* than $^{\gamma}$ ($p < 0.001$), $^{\zeta}$ ($p < 0.05$) and $^{\alpha}$ ($p < 0.05$)

	Set Members					Subsets				
Feature set	**Accuracy**	**P**	**R**	**F**	**AUC**	**Accuracy**	**P**	**R**	**F**	**AUC**
Tree Kernel Features										
SPTP + SPTF + SPETP + SPETF	86.8%$^{\varepsilon}$	0.897	0.831	0.863	0.927	84.1%	0.863	0.810	0.836	0.896
SPTP	87.1%	0.897	0.838	0.867	0.927	83.6%	0.856	0.807	0.831	0.899
SPTF	86.5%$^{\varepsilon}$	0.889	0.835	0.861	0.922	83.5%$^{\varepsilon}$	0.856	0.805	0.830	0.894
SPETP	85.2%$^{\varepsilon}$	0.881	0.815	0.847	0.917	81.9%$^{\delta}$	0.845	0.782	0.812	0.883
SPETF	85.0%$^{\varepsilon}$	0.889	0.800	0.842	0.912	81.2%$^{\delta}$	0.847	0.761	0.801	0.875
SPTF + SPETP + SPETF	86.7%	0.897	0.830	0.862	0.925	84.3%$^{\gamma}$	0.866	0.813	0.838	0.895
SPTP + SPETF + SPETP	86.8%	0.898	0.831	0.863	0.926	84.3%	0.868	0.808	0.837	0.896
SPTP + SPTF + SPETF	87.1%	0.898	0.836	0.866	0.927	83.7%$^{\varepsilon}$	0.859	0.807	0.832	0.895
SPTP + SPTF + SPETP	87.2%$^{\gamma}$	0.899	0.837	0.867	0.928	84.0%	0.864	0.807	0.834	0.898
Combination kernels										
All Trees, All features (Lin)	88.8%$^{\iota}$	0.919	0.851	0.884	0.941	86.0%$^{\iota}$	0.898	0.812	0.852	0.915
SPTF + SPTP + SPETP + All - Syntax	89.1%$^{\iota}$	0.922	0.855	0.887	0.942	85.8%$^{\iota}$	0.888	0.821	0.853	0.917

Table 4.9: Results of the intrasentential classifier on the balanced data set — tree kernels and combined kernels

$^{\gamma}$ Tree Kernel Algorithm with highest accuracy
$^{\delta}$ Significantly *worse* than $^{\gamma}$, significance $p < 0.001$, McNemar's χ^2 test.
$^{\varepsilon}$ Significantly *worse* than $^{\gamma}$, significance $p < 0.05$, McNemar's χ^2 test.
$^{\iota}$ Significantly *better* than $^{\gamma}$ ($p < 0.001$) and $^{\zeta}$, the best performing SVM flat feature algorithm in Table 4.8 ($p < 0.05$).

Set members. Firstly, we note that our best performing tree kernel combination for the set member data set has an accuracy of 87.2%, within a percentage point of both the best ICSIBoost flat feature combination (87.8%) and the best SVM flat feature combination (87.6%).

When testing the tree kernels in isolation, we found that the SPTP kernel performed best, with the difference between it and the SPETP and SPETF kernels being significant. However, the difference between the SPTP and SPTF kernel was not significant.

We also combined all 4 kernels. This gave similar performance to the SPTP kernel — the difference was not significant. We then removed each kernel in turn, in a fashion similar to the feature ablation we performed with our flat features (see Section 4.4.2.1). This process gave us our best performing tree kernel for set members, SPTP + SPTF + SPETP. This attained an accuracy of 87.2%, but was not significantly different from any of the other ablation combinations.

Subsets. For the subset data, our best tree kernel combination scores the same accuracy as its best SVM flat feature counterpart, and just 0.2% lower than the best ICSIBoost feature combination. When we take into account the comparative complexity of the flat features and the much longer development time needed, the performance of the tree kernels is impressive. When tested in isolation the SPTP performed significantly better than SPETP and SPETF, but the difference between SPTP and SPTF was not significant.

Again, the combination of all 4 kernels was not significantly different to the SPTP alone. When we removed each kernel, we found the best performing subset tree kernel was SPTF + SPETP + SPETF, with an accuracy of 84.3%. This was significantly better than the SPTP + SPTF + SPETF combination, but not the other two combinations.

The difference between partially and fully delexicalised versions of the same subtree feature was never significant for subsets or set members. The extra terminal nodes included made no significant difference. This suggests one of three possibilities. Firstly, structural features of the trees, rather than their leaves, may be important for entity instantiation identification. Secondly, the information needed may be encapsulated by the parts-of-speech, which are included, making the words unnecessary. Finally, it may simply be that more data is required for a lexicalised version of the tree representation to be useful.

4.4.2.3 Evaluation of combination kernels on balanced data set

We combined flat features and tree kernels, in an attempt to further improve performance. The results are shown in Table 4.9.

Set members. For set members, combining all flat features and all tree kernels scored 88.8%, significantly better than both the best performing flat features in isolation and the best performing tree kernels in isolation. Combining the best performing flat feature set and the best performing tree kernel combination scored 89.1%.

On the subset data, combining all flat features and all tree kernels scored 86.0%, significantly better than the best performing SVM flat feature and SVM tree kernel classifiers. Combining the best flat features and best tree kernels gave 85.8%, also significantly better than the best performing SVM flat feature and SVM tree kernel classifiers.

4.4.2.4 Evaluation of flat feature performance on unbalanced data set

Having established the utility of our features on a balanced data set, we next applied the same algorithms to our unbalanced data sets. The results for flat features and tree/combination kernels are shown in Tables 4.10 and 4.11 respectively.

Baselines. The first thing we note is the performance of our Unigram baseline. Unigram scores lower than predicting the majority for both set membership and subset instantiations, but still attains high accuracies of 92.7% and 94.0%. The other metrics, however, suggest that the performance of this baseline is not impressive. On the set member data, it scores low precision (0.286), recall (0.157) and F-Score (0.203), and a high balanced error rate (0.434). The subset Unigram has similarly weak performance.

Full feature set. Our full flat feature set with ICSIBoost beats the baseline significantly for both set members and subsets. Not only do the full feature sets have higher F-Score, they have vastly improved precision, recall and accuracy when compared to the Unigram baseline. The balanced error rate is also greatly reduced.

As in Section 4.4.2.1, we next perform a feature ablation.

Contextual and World Knowledge features. Again, we find that our Contextual and World Knowledge feature groups do not have a positive effect on classification accuracy. For set members, removing the World Knowledge feature group gives us a significantly better classifier than using all features; for subsets removing the Contextual feature group gives a significantly better classifier than using all features.

Surface features. Surface features remain important for both set members and subsets on the unbalanced data. Removal of this feature group led to significant drops in accu-

racy, as well as large drops in precision and recall. Examining the rules generated by ICSIBoost suggests a number of reasons for this. Part-of-Speech based features have an impact, as the classifier learns which POS tags are more or less likely in instantiations. Common unigrams, such as *'such'*, *'we'* and *'and'* are also identified by these features, and additionally the classifier learns rules suggesting that a very low minimum edit distance between head word lemmas is indicative of an instantiation.

Salience features. In Section 4.4.2.1, we found that removal of Salience features had a significant negative effect on the performance of set membership classification but not subset classification. On the unbalanced data, we find that removal of Salience features is significant for both.

Syntax features. Syntax features were not found to be significantly helpful on the balanced data, but here they make a significant difference to both set members and subsets. It may be that the significance of these features are amplified by the larger data set.

Using the SVM classifier rather than the ICSIBoost classifier gives some variation in our findings. We still see that Surface and Salience features are significant, but Syntax no longer is significant for set members. Our SVM classifier performs worse when World Knowledge features are removed, the opposite effect when compared to ICSIBoost.

4.4.2.5 Evaluation of tree kernel performance on unbalanced data set

We performed the same combination of tree kernels as detailed in Section 4.4.2.2. We notice that on this bigger data set, the performance of our various tree kernel combinations are much closer. There are no significant differences between the performance of each tree kernel combination; there seems to be no difference between partial and full lexicalisation or between including or omitting intervening context in terms of accuracy. This again suggests that a few structural features that all 4 representations have in common are important.

In general, the tree kernels have a lower F-Score than the unstructured features, because they have a lower recall. However, they do produce a classifiers with higher precision. All of our tree kernels beat the Unigram baseline significantly.

4.4.2.6 Evaluation of combined kernel performance on unbalanced data set

We next combined flat feature and tree kernels on the unbalanced data set. We combined all trees and all flat features, as well as the most accurate flat feature combination (All -

	Set Members						Subsets					
Feature set	**Accuracy**	**P**	**R**	**F**	**AUC**	**BER**	**Accuracy**	**P**	**R**	**F**	**AUC**	**BER**
Majority	94.1%	—	—	—	—	—	95.2%	—	—	—	—	—
Unigrams	92.7%$^{\beta}$	0.286	0.157	0.203	0.826	0.434	94.0%$^{\beta}$	0.231	0.106	0.146	0.780	0.456
ICSIBoost, Flat Features												
All	97.1%$^{\gamma}$	0.849	0.616	0.714	0.937	0.196	97.1%$^{\gamma}$	0.816	0.525	0.639	0.907	0.241
All - Surface	95.9%$^{\beta}$	0.735	0.492	0.590	0.897	0.260	95.5%$^{\beta}$	0.556	0.333	0.416	0.848	0.340
All - Salience	96.2%$^{\beta}$	0.831	0.450	0.584	0.905	0.278	96.6%$^{\beta}$	0.768	0.410	0.535	0.891	0.298
All - Syntax	97.0%$^{\beta}$	0.838	0.604	0.702	0.932	0.202	96.8%$^{\beta}$	0.793	0.471	0.591	0.907	0.268
All - Contextual	97.1%	0.854	0.627	0.723	0.934	0.190	97.3%$^{\alpha}$	0.842	0.553	0.667	0.898	0.226
All - World Knowledge	97.2%$^{\alpha}$	0.854	0.631	0.726	0.932	0.188	97.2%	0.841	0.527	0.648	0.909	0.239
SVM, Flat Features												
All features, Linear	96.9%	0.847	0.578	0.687	0.936	0.214	96.8%$^{\eta}$	0.842	0.425	0.565	0.907	0.289
All features, Polynomial	95.7%$^{\zeta}$	0.919	0.303	0.456	0.935	0.349	95.5%$^{\zeta}$	0.900	0.08	0.152	0.903	0.459
All features - Surface (Lin)	96.2%$^{\eta}$	0.805	0.475	0.597	0.898	0.266	96.1%$^{\zeta}$	0.774	0.282	0.414	0.840	0.361
All features - Salience (Lin)	95.7%$^{\zeta}$	0.836	0.337	0.481	0.910	0.333	96.2%$^{\zeta}$	0.867	0.265	0.406	0.895	0.369
All features - Syntax (Lin)	96.6%	0.835	0.538	0.654	0.933	0.234	96.1%$^{\zeta}$	0.791	0.271	0.404	0.907	0.366
All features - Contextual (Lin)	97.0%$^{\varepsilon}$	0.849	0.597	0.701	0.937	0.205	97.0%$^{\varepsilon}$	0.834	0.471	0.602	0.903	0.267
All features - World Knowledge (Lin)	96.7%$^{\eta}$	0.834	0.552	0.665	0.925	0.227	96.6%$^{\zeta}$	0.852	0.788	0.833	0.906	0.324

Table 4.10: Results of the intrasentential classifier on the original data set — flat features

$^{\alpha}$ ICSIBoost Algorithm with highest accuracy
$^{\beta}$ Significantly *worse* than $^{\alpha}$, significance $p < 0.001$, McNemar's χ^2 test.
$^{\gamma}$ Significantly *worse* than $^{\alpha}$, significance $p < 0.05$, McNemar's χ^2 test.
$^{\varepsilon}$ SVM flat-feature algorithm with highest accuracy
$^{\zeta}$ Significantly *worse* than $^{\varepsilon}$, significance $p < 0.001$, McNemar's χ^2 test.
$^{\eta}$ Significantly *worse* than $^{\varepsilon}$, significance $p < 0.05$, McNemar's χ^2 test.

	Set Members						Subsets					
Feature set	**Acc.**	**P**	**R**	**F**	**AUC**	**BER**	**Acc.**	**P**	**R**	**F**	**AUC**	**BER**
Tree Kernel Features												
SPTP + SPTF + SPETP + SPETF	96.7%	0.894	0.495	0.637	0.925	0.254	96.7%	0.937	0.342	0.501	0.907	0.329
SPTP	96.7%	0.922	0.479	0.631	0.921	0.262	96.7%	0.942	0.339	0.498	0.902	0.331
SPTF	96.7%	0.922	0.479	0.631	0.921	0.262	96.7%	0.942	0.339	0.498	0.902	0.331
SPETP	96.3%	0.942	0.409	0.570	0.918	0.296	96.7%	0.942	0.336	0.496	0.902	0.332
SPETF	96.3%	0.937	0.404	0.565	0.913	0.298	96.7%	0.945	0.335	0.495	0.894	0.333
SPTF + SPETP + SPETF	96.6%	0.897	0.486	0.630	0.924	0.259	96.7%	0.940	0.345	0.504	0.905	0.328
SPTP + SPETF + SPETP	96.7%$^{\delta}$	0.914	0.491	0.638	0.924	0.256	96.7%	0.937	0.343	0.503	0.906	0.329
SPTP + SPTF + SPETF	96.6%	0.892	0.492	0.634	0.925	0.256	96.7%$^{\delta}$	0.940	0.345	0.504	0.907	0.328
SPTP + SPTF + SPETP	96.7%	0.908	0.494	0.640	0.927	0.255	96.7%	0.934	0.343	0.502	0.908	0.329
Combination kernels												
All Trees, All features (Lin)	97.0%	0.884	0.579	0.699	0.941	0.213	97.2%$^{\theta}$	0.934	0.461	0.618	0.923	0.270
SPTF + SPTP + SPETF + All - Contextual	97.1%$^{\theta}$	0.889	0.591	0.710	0.942	0.207	97.3%$^{\theta}$	0.936	0.476	0.631	0.920	0.263
SPTP + SPETF + SPETP + All - Contextual	97.1%	0.886	0.586	0.705	0.942	0.209	97.3%$^{\theta}$	0.935	0.479	0.633	0.921	0.262

Table 4.11: Results of the intrasentential classifier on the original data set — tree and combination kernels

$^{\delta}$ Tree Kernel Algorithm with highest accuracy

$^{\theta}$ Significantly *better* than best performing SVM flat feature algorithm, $^{\varepsilon}$ from Table 4.10 ($p < 0.05$) and $^{\delta}$ ($p < 0.001$)

Contextual) and the most accurate tree kernel algorithm (SPTF + SPTP + SPETF for set members, SPTP + SPETF + SPETP for subsets).

For subsets, both combinations were significantly better than both the best SVM flat feature classifier and the best tree kernel classifier in isolation.

With set members, only the the most accurate flat feature combination and the most accurate tree kernel algorithm gave an improvement over the best SVM flat feature classifier and the best tree kernel classifier in isolation.

4.4.2.7 Intrasentential summary results

As our intrasentential study includes two data sets, two relations, five categories of flat feature and four tree kernels, we include a short summary in Table 4.12 to make clear the important results.

4.4.2.8 Error analysis

We analyse the errors our algorithm makes in two ways, by breaking down the performance according to the syntactic relationship between the NPs, and by manually inspecting the errors. Our error analysis is based on performance on the original, unbalanced data sets.

Syntactic relationship breakdown. Firstly, we show the performance of our best algorithms with respect to the syntactic relationship between the two NPs. We use the syntactic relationship categories detailed in Section 3.8.3. Tables 4.13 and 4.14 show the results for set members and subsets, respectively.

For both set members and subsets, we record our highest F-scores by far on instantiations where the set is the syntactic parent of the set member or subset. As this category contains the vast majority of positive examples of intrasentential instantiations, it makes sense that our machine learner deals with this case most effectively.

Manual inspection. We manually inspect the errors of our best performing algorithms, looking for common trends within both the false positives and false negatives the algorithm produces. Due to the skewed nature of the data, the algorithm tends to under-predict slightly, with false positives being rarer than false negatives. Within the results of classifiers we find the three re-occurring types of false positives.

Firstly, we see several false positives where the NPs have identical head words or head word lemmas but are not in an instantiation, such as Example 4.14.

Instantiation Type	**Corpus**	**Best Flat Feature combination**	**Sig. Dif. when removed**	**Best TK combination**	**Combination kernel with significant improvement**
Set members	Balanced	All - Contextual (87.8%)	Surface (-4.3%) Salience (-3.3%)	SPTP + SPTF + SPETP (87.2%)	All TK, All Features (88.8%)
	Unbalanced	All - Knowledge (97.2%)	Surface (-1.3%) Salience (-1.0%) Syntax (-0.2%)	SPTP + SPETF + SPETP (96.7%)	Best TK, Best Features (97.1%)
Subsets	Balanced	All (84.5%)	Surface (-5.9%)	SPTF + SPETP + SPETF (86.7%)	All TK, All Features (86.0%)
	Unbalanced	All - Contextual (97.3%)	Surface (-1.8%) Salience (-0.7%) Syntax (-0.5%)	SPTP + SPTF + SPETF (96.7%)	All TK, All Features (97.2%) Best TK, Best Features (97.3%)

Table 4.12: Intrasentential Summary Results.

	Actual Label Distribution			Algorithm performance			
Syntactic Relationship	**Set Member**	**Other Sing-Plur pair**	**Total**	**Acc.**	**P**	**R**	**F**
Set NP Parent	1 065	2 294	3 359	91.40%	0.906	0.813	0.857
Member NP Parent	55	1 843	1 898	97.26%	0.588	0.182	0.278
Same Clause	84	7 068	7 152	98.74%	0.125	0.012	0.022
Different Clause	334	13 158	13492	97.69%	0.762	0.096	0.170

Table 4.13: Breakdown of best performing intrasentential set member algorithm by syntactic relationship.

	Actual Label Distribution			Algorithm performance			
Syntactic Relationship	**Subset**	**Other Plur-Plur pair**	**Total**	**Acc.**	**P**	**R**	**F**
Set NP Parent	615	1 489	2 104	88.74%	0.938	0.659	0.774
Subset NP Parent	85	1 991	2 076	96.24%	0.889	0.094	0.170
Same Clause	90	4 945	5 035	98.21%	—	0.000	—
Different Clause	75	8 603	8 678	99.14%	0.500	0.013	0.026

Table 4.14: Breakdown of best performing intrasentential subset algorithm by syntactic relationship.

(4.14) Soon the studio is producing a $40 million picture called *"Tet, the Motion Picture,"* to distinguish it from **"Tet, the Offensive," as well as "Tet, the Book" and "Tet, the Album."**

Secondly, and more numerously, false positives occur where the erroneously identified member is nested within the set but is not an instantiation. Examples of this include Examples 4.15 and 4.16. A subset of these cases exist where the set is negated, such as Example 4.17, suggesting a feature which explicitly identifies negated sets might be a useful addition.

(4.15) **good extensions to Boston, *New York* and Dallas**

(4.16) **F-16 Fighting Falcon and Mirage 2000 combat aircraft, produced by the U.S. based General Dynamics Corp. and *France's Avions Marcel Dassault*, respectively.**

(4.17) **Neither the opposition nor *the LDP***

Thirdly, we see a number of cases that appear to be valid instantiations but that have

not been identified by the annotation process. These false positives are not down to errors in the machine learning algorithm.

In terms of false negatives, we see the following 4 categories.

Firstly, related to the false positives with nested NPs, we see false negatives where the nesting relationship is not straightforward, such as Examples 4.18, 4.19, 4.20 and 4.21. Although our tree kernel methods perform well, this suggests that either a better tree kernel representation might be useful in disambiguating these cases, or that tree kernels alone are not enough to disambiguate some entity instantiations and additional flat features are needed to aid them.

(4.18) **The places renowned for breeding bunco, like the Miami neighborhood known as the "Maggot Mile" and *Las Vegas's flashy strip of casinos***

(4.19) *Gulbuddin Hekhmatyar, perhaps the most hated and feared of **the extremists***

(4.20) *Planar Systems Inc. of Beaverton, Ore., the largest of **these firms***

(4.21) **catastrophic illnesses and conditions such as *cancer*, heart attacks, renal failure and kidney transplants.**

Secondly, we see false negatives where the set or occasionally the member/subset are pronouns, illustrated by Examples 4.22, 4.23, 4.24 and 4.25. This suggests that using coreference data to resolve these pronouns, and additionally including data about the antecedent, might be useful feature additions.

(4.22) In addition, *Sen. McCain* last week disclosed that he belatedly had paid $13,433 to American Continental as reimbursement for trips **he and his family** took aboard the corporate jet to Mr. Keating's vacation home at Cat Cay, the Bahamas, from 1984 through 1986.

(4.23) Last summer, in response to congressional criticism, *the State Department* and the CIA said **they** had resumed military aid to the resistance months after it was cut off; but it is not clear how much is being sent or when it will arrive.

(4.24) "**We** don't see a domestic source for some of our {HDTV} requirements, and that's a source of concern," says *Michael Kelly, director of DARPA's defense manufacturing office.*

(4.25) All of which has enabled **those of us in Washington who enjoy wallowing in such things** to go into high public dudgeon, as Mr. Apple and *I* did the other night on ABC's "Nightline."

Thirdly, we see situations where the head words are related, but the relationship is not identified. In Examples 4.26 and 4.28, the two head words are related by hyponymy — a share issue is a type of security and missiles are types of weapons. In Example 4.27, there is a metonymic relationship between Beijing and China, and in Example 4.29 *'carriers'* and *'Airlines'* are synonyms in this context. This suggests that better word similarity metrics could improve classification.

(4.26) *The largest issue* was a $4 billion offering of **auto-loan securities** by General Motors Acceptance Corp. in 1986.

(4.27) In a sign of easing tension between **Beijing and Hong Kong**, *China* said it will again take back illegal immigrants caught crossing into the British colony.

(4.28) This includes what Deputy Foreign Minister Yuli Vorontsov fetchingly called **"new peaceful long-range weapons,"** including *more than 800 SCUD missiles*.

(4.29) Mr. Eddington sees alliances with **other carriers** – particularly Cathay's recent link with *AMR Corp.'s American Airlines* – as an important part of Cathay's strategy.

Finally, we see cases where world knowledge, knowledge from elsewhere in the document, or logical inference is required to interpret the instantiation. In most of these cases, a human reader could probably infer the link between member/subset and set without world knowledge. In Example 4.30, one needs to know that *'Sen. DeConcini'* is a senator, and that he is also a member of *'the five senators'* that were previously mentioned in the document — though the fact that the meeting was in his office might allow us to infer his set membership. In Example 4.31, one needs to know a *party* is a participant in talks, and to infer that the fact *'De Beers'* and *'the union'* are making offers means they are participants. To interpret Example 4.32, we need to either know from earlier on in the document that *'Hurricane Hugo'* is a disaster, or infer it from the sentence. The relationship between the NPs in Example 4.33 is unlikely to be contained in any knowledge base, but instead should be inferred from the context.

(4.30) In a highly unusual meeting in *Sen. DeConcini*'s office in April 1987, **the five senators** asked federal regulators to ease up on Lincoln.

(4.31) Before **the two parties** resumed talks last week, *De Beers* offered 17% and *the union* wanted 37.6%.

(4.32) The funds are in addition to $1.1 billion appropriated last month to assist in the recovery from *Hugo*, bringing the total for **the two disasters** to nearly $4 billion in unanticipated spending.

(4.33) *The art of change-ringing* is peculiar to the English, and, like **most English peculiarities**, unintelligible to the rest of the world.

These knowledge-based and inferential cases are the hardest to suggest a simple feature to identify. One possible solution might be to run a relation extraction algorithm on the documents, to try and establish a document-level knowledge base for the text, as some of the entities involved are unlikely to be salient enough to be contained within knowledge bases such as Freebase or Wikipedia.

4.4.3 Intersentential evaluation

4.4.3.1 Evaluation of performance on balanced data set

The results of our intersentential classifiers on the sub-sampled data are shown in Table 4.15.

Baselines. The results of the Unigram baseline show the problem is significantly different, and considerably harder, than the intrasentential problem. For set members and subsets, this baseline has a reduction in accuracy of 15% and 25% respectively over their intrasentential counterparts.

Full feature set. Using our full feature set gives us accuracies of 69.77% and 61.31% for set members and subsets respectively. This is a significant increase over the baseline in both cases.

Feature ablation. We performed a feature ablation study, removing each group of features from our model in turn, the results of which are also present in Table 4.15. From our feature ablation, we can draw a number of conclusions.

Firstly, it is clear set members and subsets perform quite differently — the highest scoring set member algorithm increases the accuracy by more than 7 percentage points over its subset counter part. This suggests that either the subset problem is harder, or that the feature set is less suitable for this problem. It is also clear that the different categories of features have a different impact on each problem.

However, several observations hold across both problems. Firstly, the Syntax features are not helpful in identifying instantiations and in fact removing them improves performance. Secondly, the removal of Contextual features gives a non-significant performance drop, suggesting that these features are also of little help. Finally, Salience features are significantly helpful, and especially so for subsets — their removal leads to a large drop in performance.

For set members, we see that the World Knowledge features are good for identifying instantiations. Upon further investigation, we discovered that our Google PMI feature is the most effective of this feature group, with large PMI values often being indicative of instantiations. This contrasts with results of the ICSIBoost set member classifier on intrasentential instantiations, where the removal of world knowledge features is *not* significant.

For subsets, the results show that only Salience features significantly reduce classification performance when removed. However, all feature configurations comfortably beat the unigram baseline.

4.4.3.2 Evaluation of performance on unbalanced data set

Secondly, we experimented with the original, highly skewed data. Our initial attempts with training on the original data resulted in a classifier that almost never predicted an instantiation, so we experimented with some simple techniques to improve recall. These comprised randomly sub-sampling the negative examples so that they made up 50% or 75% of the training data, and oversampling the positive examples in the training data by a factor of 10, 20 or 40. The results of these experiments are shown in Table 4.16. As before, results for the Unigram baseline are also included.

Learning from the original, highly skewed data is much more difficult than the balanced data used in Section 4.4.3.1. None of these methods lead to a more accurate algorithm, and our highest F-scores are 0.1938 and 0.1414 for set members and subsets, respectively. The AUC shows very little change over the different algorithms, which suggests that these sampling methods just shift the classification boundary, rather than making a meaningful change in the learning process.

We temper these disappointing results with the knowledge that learning from data with this sort of distribution is difficult, regardless of the domain. In future we intend to use techniques such as SMOTE (Chawla et al., 2002) and One-Sided Selection (Kubat and Matwin, 1997) to address this heavy skew.

Additionally, ROC curves were produced for each sampling method, and are included as Figures 4.13 and 4.14. We see from the ROC curves, and the AUCs reported in Ta-

	Set Members					Subsets				
Feature set	**Accuracy**	**P**	**R**	**F**	**AUC**	**Accuracy**	**P**	**R**	**F**	**AUC**
Majority	50.0%	—	—	—	—	50.0%	—	—	—	—
Unigrams	63.13%♢	0.6862	0.4841	0.5677	0.6736	52.96%♢	0.5772	0.2215	0.3202	0.5646
All	69.77%	0.7797	0.5511	0.6458	0.7636	61.31%	0.6944	0.4040	0.5108	0.6704
All - Surface	67.37%♠	0.7067	0.5938	0.6453	0.7307	62.32%	0.6717	0.4821	0.5613	0.6833
All - Salience	68.04%♠	0.7695	0.5152	0.6172	0.7488	59.98%♢	0.6749	0.3853	0.4906	0.6540
All - Syntax	69.94%♣	0.7745	0.5626	0.6518	0.7644	62.32%♣	0.7026	0.4275	0.5315	0.6654
All - Contextual	68.68%	0.7680	0.5355	0.6310	0.7603	61.00%	0.6932	0.3947	0.5030	0.6710
All - World Knowledge	68.01%♠	0.7687	0.5152	0.6169	0.7494	60.53%	0.6692	0.4165	0.5135	0.6683

Table 4.15: Results of the intersentential classifier on the balanced data set

♣ Algorithm with highest accuracy
♠ Significantly *worse* than ♣, significance $p < 0.005$, McNemar's χ^2 test.
♢ Significantly *worse* than ♣, significance $p < 0.001$, McNemar's χ^2 test.

	Set Members					Subsets				
Method	**Accuracy**	**P**	**R**	**F**	**AUC**	**Accuracy**	**P**	**R**	**F**	**AUC**
Original Set	96.90%	0.5080	0.0643	0.1142	0.7719	97.76%	0.1639	0.0156	0.0285	0.7286
Undersampling 50/50	82.05%	0.1012	0.6073	0.1735	0.7814	80.26%	0.0507	0.4727	0.0916	0.7246
Undersampling 75/25	93.41%	0.1825	0.3230	0.2332	0.7798	93.69%	0.0966	0.2387	0.1375	0.7256
Oversampling x10	95.25%	0.2258	0.2180	0.2218	0.7717	96.90%	0.1475	0.0982	0.1180	0.7267
Oversampling x20	93.67%	0.1935	0.3283	0.2435	0.7705	96.25%	0.1400	0.1513	0.1454	0.7092
Oversampling x40	90.42%	0.1462	0.4312	0.2184	0.7639	95.38%	0.1274	0.2044	0.1570	0.7142

Table 4.16: Results of the intersentential classifier on the original data set

	Actual Label Distribution			Algorithm performance			
NP type	**Set Member**	**Other Sing-Plur pair**	**Total**	**Acc.**	**P**	**R**	**F**
Name	716	406	1122	67.74%	0.8000	0.6592	0.7228
Pronoun	228	111	339	66.08%	0.7729	0.7018	0.7356
Common Noun	471	835	1306	72.97%	0.7418	0.3843	0.5063
Numeric	32	73	105	72.38%	0.5714	0.3750	0.4528
Other	30	52	82	64.63%	0.5455	0.2000	0.2927

Table 4.17: Breakdown of best performing intersentential set member algorithm by set member NP type.

ble 4.16, that none of the sampling methods drastically change the curve or the area under the curve. These sampling methods just move the decision boundary, changing the balance between precision and recall.

4.4.3.3 Error analysis

As in Section 4.4.2.8, we analyse the errors of our algorithm in two ways; by breaking down the performance according to the NP type of the set member/subset, and by manually inspecting the errors. Our error analysis is based on performance on the balanced data set.

NP type breakdown. Firstly, we show the performance of our best algorithms with respect to the category of the set member/subset NP. We use the NP type categories detailed in Section 3.8.4.2. Tables 4.17 and 4.18 show the results for set members and subsets, respectively.

For set members, accuracy remains reasonably constant across the 5 categories. However, we see a drop in large drop in recall for common nouns, numerics and others. In the case of numeric and other, this can be explained by the relatively small number of instances, but the algorithm finds classification of common nouns harder than pronouns and names.

For subsets, common nouns make up the vast majority of instantiations. The classifier has good precision on this category, but has recall of only 0.44.

Manual inspection. A manual inspection of the false positives shows two main categories where the algorithm over predicts, NP pairs where one of the participants is a pronoun, and NP pairs where the error could be avoided by a stronger features indicating

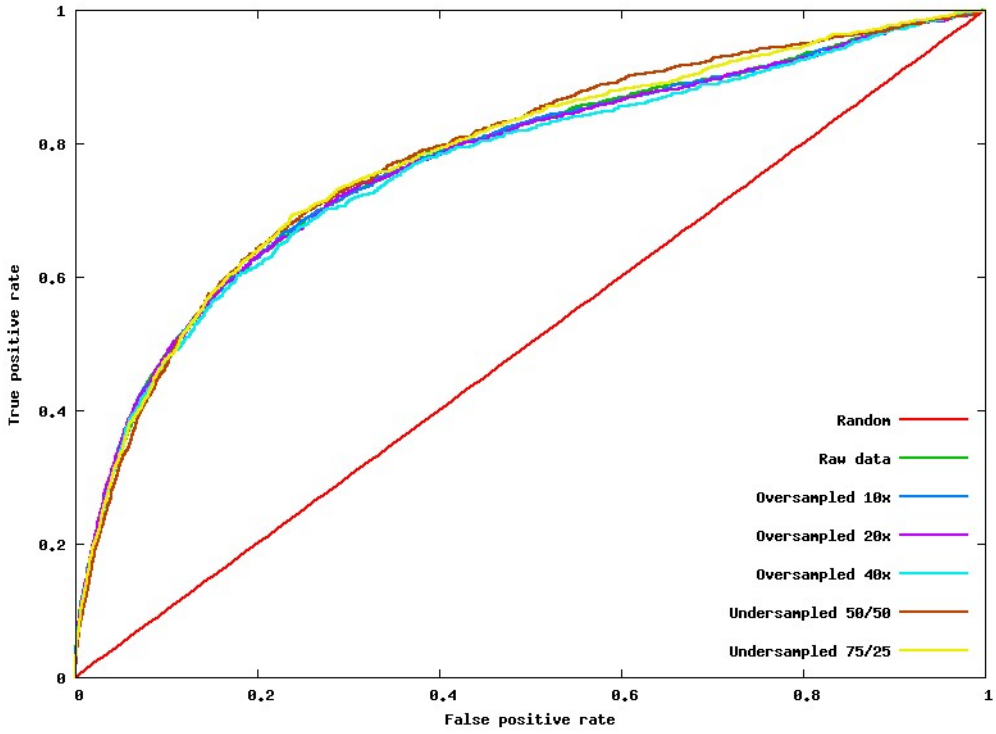


Figure 4.13: ROC curve: Set members

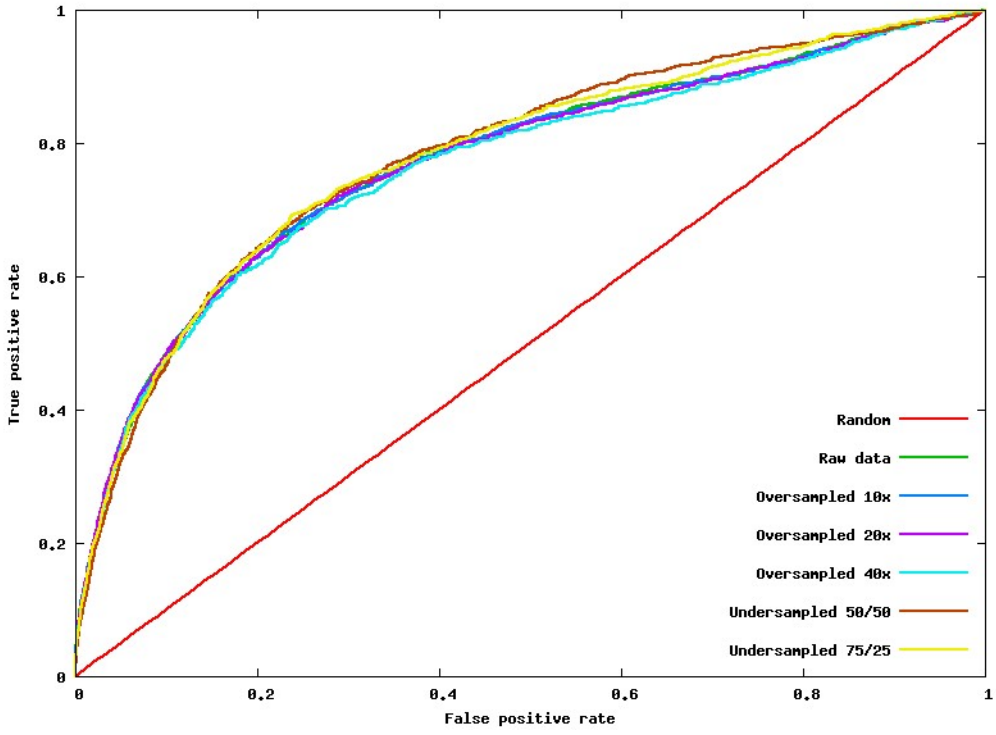


Figure 4.14: ROC curve: Subsets

	Actual Label Distribution			Algorithm performance			
Syntactic Relationship	**Subset**	**Other Plur-Plur pair**	**Total**	**Acc.**	**P**	**R**	**F**
Name	41	29	70	55.71%	0.7083	0.4146	0.5231
Pronoun	65	67	132	51.52%	0.5238	0.1692	0.2558
Common Noun	490	525	1015	64.14%	0.7032	0.4449	0.5450
Numeric	13	6	19	42.11%	0.6667	0.3077	0.4211
Other	32	14	46	71.74%	0.8276	0.7500	0.7869

Table 4.18: Breakdown of best performing intersentential subset algorithm by subset NP type.

mismatching entity types.

Examples 4.34, 4.35 and 4.36 show examples of pronoun related error. As suggested in Section 4.4.2.8, some of these errors could be avoided by using coreference data to identify the antecedents of these anaphora, and involving it in the feature generation.

(4.34) a. "I think {Mr. Phillips} is going to need *some help*.

b. I think they need creative leadership, and I don't think **they** have it," said Emma Hill, an analyst with Wertheim & Co.

(4.35) a. "I guess I might have asked Beauregard to leave, but he drops so many good names, **we** decided to let him stay," says Steven Greenberg, publisher of Fame.

b. "After all, *Warhol* was the ultimate namedropper, dropping five a day in his diaries.

(4.36) a. Another argument of the environmentalists is that if substitutes are available, why not use **them**?

b. *Mr. Teagan* cites a list of substitutes but none, so far, match the nonflammable, nontoxic CFCs.

In the second category, we see examples where the false positive NPs are of different entity types. In Example 4.37, the set NP refers to journalistic output, and the member NP refers to a person, and in Example 4.38, the set NP refers to people, and the member NP refers to a location. Similarly, in Example 4.39, the set NP is an organisation and the member NP is a person. These errors suggest that features which identify the type of each NP more specifically, and indicate if the types do not match, could be helpful.

(4.37) a. That's such a departure from the past that many in the industry are skeptical CNN will follow through with its investigative commitment, especially after it sees the cost of producing **in-depth pieces**.

b. "They've never shown any inclination to spend money on production," says *Michael Mosettig, a senior producer with MacNeil-Lehrer NewsHour, who notes that CNN is indispensable to his job*.

(4.38) a. Moreover, **both men** have hewn to a similar hard-line philosophy.

b. Notably, one of Mr. Krenz's few official visits overseas came a few months ago, when he visited *China* after the massacre in Beijing.

(4.39) a. Common Cause asked **both the Senate Ethics Committee and the Justice Department** to investigate $1 million in political gifts by Arizona businessman Charles Keating to five U.S. senators who interceded with thrift-industry regulators for him.

b. *Mr. Keating* is currently the subject of a $1.1 billion federal anti-racketeering lawsuit accusing him of bleeding off assets of a California thrift he controlled, Lincoln Savings & Loan Association, and driving it into insolvency.

In terms of false negatives, we again see difficulties with pronouns, as in Examples 4.40, 4.41 and 4.42

(4.40) a. Imprisoned by the Nazis during World War II for his political beliefs, Mr. Honecker typified the postwar generation of **committed Communist leaders** in Eastern Europe who took their cues from Moscow.

b. *He* was a "socialist warrior" who felt rankled by West Germany's enormous postwar prosperity and the Bonn government's steadfast refusal to recognize the legitimacy of his state.

(4.41) a. And **surprising numbers of small investors** seem to be adapting to greater stock market volatility and say they can live with program trading.

b. Glenn Britta, a 25-year-old New York financial analyst who plays options for his personal account, says *he* is "factoring" the market's volatility "into investment decisions." He adds that program trading "increases liquidity in the market.

(4.42) a. He hurt *himself* further this summer by bringing homosexual issues into the debate; and by wavering on this issue and abortion, he has weakened his credibility in what is already a mean-spirited campaign on both sides.

b. Elected to Congress in 1978, the 48-year-old Mr. Courter is part of a generation of **young conservatives who were once very much in the lead of the rightward shift under Mr. Reagan.**

Secondly, despite a good recall on named entity based set members, our algorithm under predicts somewhat. It is likely that further training data could aid this problem. Examples 4.43 and 4.44 are amongst those that are missed by our classifier.

(4.43) a. Indeed, according to West German government sources, he was one of **the leaders in the power struggle that toppled Mr. Honecker**.

b. In recent days, *Mr. Krenz* has sought to project a kinder image.

(4.44) a. He also will sit on the company's corporate planning and policy committee, made up of **the top corporate and operating executives**.

b. *Mr. Roman*'s departure isn't expected to have any enormous repercussions at Ogilvy.

Finally we see cases where conjunctions and appositions to the head of the NP provide important context for identifying the instantiation. Although some of our world knowledge features do use the conjunctions and appositions, these cases suggest that simply including their head words as features might also be useful. Example 4.45 shows a false negative where the conjunction of the set NP helps identify the instantiation. Example 4.46 shows a false negative where the apposition of the set member NP helps identify the instantiation.

(4.45) a. **Five states – Oregon, Rhode Island, New Hampshire, Iowa and Wisconsin** – passed bills to boost the minimum wage, but measures in 19 other states were defeated.

b. *Oregon*'s rate will rise to $4.75 an hour, the nation's highest, in Jan. 1, 1991.

(4.46) a. But **other analysts** said that having Mr. Phillips succeed Mr. Roman would make for a smooth transition.

b. "Graham Phillips has been there a long time, knows the culture well, is aggressive, and apparently gets along well with" Mr. Sorrell, said *Andrew Wallach, an analyst with Drexel Burnham Lambert.*

4.5 Conclusion

In this Chapter we have explored the machine learning of entity instantiations. We developed a feature set appropriate to the problem, with features reflecting several categories we felt were important to the problem, namely Surface, Salience, Syntax, Contextual and World Knowledge. This feature set is used for both intrasentential and intersentential entity instantiation recognition.

For the problem of intrasentential entity instantiations, we supplement this feature set with tree kernels — a method of learning directly from tree structures. In our case we provide the learner with sub-trees that encapsulate the shortest path between two NPs that are in a possible instantiation.

We chose to treat intrasentential and intersentential entity instantiations as distinct classification problems. We also chose to separate the identification of set membership entity instantiations, which only occur between a singular NP and a plural NP, and subset entity instantiations, which only occur between a pair of plural NPs. Both of these decisions are well justified in light of the classification results. Despite the similarities between the tasks, our results reflect the fact each of them performs quite differently.

Due to the nature of the phenomenon, we had many more negative examples than positive examples, for both inter and intrasentential entity instantiations. To better establish the utility of our features, we constructed balanced data sets with equal numbers of positive and negative examples and analysed our results on these before progressing to apply our algorithms to the original, highly skewed data.

Our results on the intrasentential data were very positive. When training and testing on the balanced data sets, both ICSIBoost and SVM classifiers using flat features made highly significant increases in performance over the Unigram baseline. Tree kernels in isolation also showed highly significant increases in performance over the Unigram baseline, and combining both tree kernels and flat features gave a further significant improvement. Our best set member classifier scored an accuracy of 89.1%, and our best subset classifier scored an accuracy of 86.0%.

On the full, unbalanced data set, tree kernels, flat features and combinations of both approaches again show highly significant improvements over the Unigram baseline, despite the skewed distribution. Our best classifiers scored accuracies of 97.1% and 97.3% for set members and subsets respectively.

We found the classification of intersentential entity instantiations more challenging — on a balanced data set our highest set member accuracy was 69.9% and our highest subset accuracy was 61.3%, both significantly better than the Unigram baseline.

On unbalanced intersentential data, we found that our classifier was unable to beat a baseline of predicting the majority. We experimented with some simple sampling techniques, but none of these improved classification accuracy.

The results of our machine learning experiments confirm some of the hypotheses we formulated in Section 1.4.2. We confirm that a supervised machine learning approach can be used to automatically identify entity instantiations from texts with a reasonable degree of accuracy. Additionally, we find our hypotheses regarding the importance of surface forms, salience features, world knowledge features and syntactic relationships proved.

4.6 Future Work

There are several ideas we wish to explore in the future, and that we hope would further improve our classifiers.

4.6.1 Feature selection

In this Chapter we have compared and contrasted the relative performance of features by conducting feature ablations, in which features, or in our case groups of features, are systematically *removed* from the model. Whilst this method has merit, and gave useful information about the utility of our broad groups of features, there are a number of more sophisticated metrics which could be used to judge feature performance.

One group of methods rank the utility of features, and then apply a threshold to remove low scoring features (Yang and Pedersen, 1997). One widely used metric for this feature ranking is *information gain*, which is a measure of the reduction in entropy — and therefore gain in information – given by a feature (Mitchell, 1997). Other options include using *mutual information* or the χ^2 statistic to measure the association between classes and features (Yang and Pedersen, 1997).

In addition to judging the merit of features individually, it can also be beneficial to consider subsets of features. A particular feature may be helpful in isolation, but can be redundant in the face of an alternative feature which represents the phenomenon better. For example, it might be that our feature that represents the head words of the NPs is unnecessary because of our feature that represents head word lemmas. Clearly, exhaustively searching through all features to find the best performing subset is impractical for a large feature set, so one can employ a more efficient search strategy, such as the Best-First algorithm (Kohavi and John, 1997), a genetic algorithm (Yang and Honavar, 1998) or Correlation-based Feature Selection (CFS) (Hall, 1999).

Rather than finding the best performing features, one may instead represent the feature space in fewer dimensions by performing a transformation that combines features. Principal Components Analysis, one such algorithm, does this by using eigenvectors to map the data to a lower dimensional space which still represents a specified amount of the variance of the data (Jolliffe, 2002).

It is our intention in future to employ some of these methods in order to better understand the utility and redundancy of our feature set, and therefore guide the development of new features whilst maintaining a compact representation of the problem.

4.6.2 Additional features

Machine-generated POS, parses and coreference. Our features rely on gold standard data from a number of sources. We use gold standard tokenisation, POS tags and parse trees from the PTB. Our dependency parses are derived from gold standard PTB parses, and our coreference data is from OntoNotes. It would be interesting to see how reliant our classifier is on this gold standard data — can similar results be achieved with automatically created equivalents?

Word similarity. To discover cross argument lexical relationships we currently use Levenshtein's distance, coupled with Levin's verb classes (Levin, 1993) and WordNet lookups. There are other, more sophisticated, methods of calculating verb similarity, such as Chklovski and Pantel (2004) and Yang and Powers (2006), which we would like to implement in future.

World knowledge features. In Section 2.2, we discuss several methods for extracting context-independent relations, some of which use many patterns to identify relations, and include algorithms which automatically identify new patterns and evaluate their utility. However, our feature set uses a single one of Hearst (1992)'s patterns. This single pattern gives good results — using more patterns, or employing at least one of these more complex methods to better identify those entity instantiations that are based upon context-independent, well-known relationships (e.g. $France \in EU\ Countries$) is likely to improve results.

4.6.3 Kernel methods

Due to the strong syntactic relationship between the participants of intrasentential entity instantiations, we employed tree kernels to learn directly from constituency parse trees.

The tree representation we used is based on that of Zhang et al. (2006) and Swampillai and Stevenson (2011), coupled with the SubSet Tree (SST) kernel learner that is part of SVM-LIGHT-TK (Moschitti, 2006b), and it performs well. However, there are a number of other options that are worth exploring for tree kernel based classification.

Firstly, one might try using a different tree kernel. The SST kernel (Collins and Duffy, 2002) allows learning from more generalised, leaf-less internal sub-trees, when compared to the SubTree (ST) kernel, which learns from sub-trees containing all the descendants of the target root node until the leaves. There is however, an even more generalised option which may be useful, the Partial Tree (PT) kernel (Moschitti, 2006a), which allows learning from tree fragments which do not necessarily conform to production rules. These even more general substructures may lead to better tree kernel classification results, and this generality may also open up the possibility of using tree kernels for intersentential learning.

Secondly, one might try learning from a structure other than a constituency parse tree. Dependency trees are commonly used in relation extraction (Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; Zhao and Grishman, 2005), and could prove an interesting alternative, or addition, to our constituency tree kernel classifier.

Thirdly, one could apply tree kernels to intersentential entity instantiations. We have not done so in this thesis because our intuition was that syntactic relationships were much more important for intrasentential instantiations. However, tree kernels could also be useful to some degree for intersentential instantiations. At least two options are feasible; following Swampillai and Stevenson (2011) and joining together the trees of two sentences under a new node, or creating two individual kernels, one for the set-containing sentence and one for the set member/subset-containing sentence, before summing the results.

4.6.4 Machine learning techniques

Dealing with highly skewed data. Due to the nature of the phenomenon, our annotation resulted in a very skewed data set. Learning from skewed data is *hard* — the tendency of most classification algorithms is to predict the class which is the overwhelming majority in these cases. We experimented with some basic sampling techniques to try and improve results, which increased recall at the cost of reduced accuracy and precision.

There are other, more sophisticated ways of learning from skewed data, including SMOTE (Chawla et al., 2002) and One-Sided Selection (Kubat and Matwin, 1997). In the future, we intend to apply these techniques to our intersentential data, in an effort to improve recall without sacrificing quite so much accuracy and precision.

Joint learning. In this thesis we consider the classification of entity instantiations as a *local* problem — each instance is treated separately, and although we include some contextual features, the classification of other proximate entity instantiations does not affect the process. However, our intuition suggests that a global classification approach might produce better results. In an example such as Example 4.47, knowing that *'the UK'* is an instantiation of *'Several countries'* means that the entities it appears in conjunction with, *'France'* and *'Spain'* are more likely to also be instantiations of the same set. At an even simpler level, knowing that a set has had one instantiation drawn from it may make it more likely to be used again.

(4.47) a. **Several countries** attended the conference, including *the UK*, France and Spain.

b. Iceland didn't turn up.

Given these considerations, we might find representing instantiations as a network or *graph* useful. In a model such as this, the nodes would be classified by taking into account the local probability that each NP pair is an instantiation, as well as the global links between NP pairs.

This sort of learning also provides the potential to learn inter- and intrasentential and set member and subset relation simultaneously, which could certainly be advantageous One potential tool for carrying out such learning experiments is NetKit-SRL (Macskassy and Provost, 2007).

Chapter 5

Entity Instantiations and Discourse Relations

5.1 Introduction

We hypothesise the existence of a connection between entity instantiations and *discourse relations*. Discourse relations are binary relations which connect *abstract objects* — clauses and sentences which represent events, states, and propositions, in contrast to the entity relations we have considered so far in this thesis (Asher, 1993).

For example, in the sentence in Example 5.1 the clauses *'John ordered the fish'* and *'Mary preferred the chicken'* are the arguments of a contrast discourse relation. Example 5.2 shows a cause discourse relation between *'John ordered the fish'* and *'he liked the lemon sauce it came with'*. The clauses and sentences which take part in discourse relations are referred to as *arguments*.

(5.1) John ordered the fish, but Mary preferred the chicken.

(5.2) John ordered the fish because he liked the lemon sauce it came with.

Discourse relations are often signalled by a *connective*, which is a single word or phrase which expresses the relation. These relations are referred to as *explicit*. In Example 5.1, the connective *'but'* signals the contrast, and in Example 5.2, the connective *'because'* indicates the causal relation.

Discourse relations can also occur without a connective, and are instead understood *implicitly*. In Example 5.3, an implicit cause relation exists between the two sentences. In Example 5.4, an implicit expansion relation is present.

(5.3) John had no room for desert. He'd eaten far too much already.

(5.4) John hated his dessert. It was the worst he'd ever eaten.

There are various theories of discourse (Mann and Thompson, 1988; Lascarides and Asher, 2007; Hobbs, 1979; Sanders et al., 1992; Halliday and Hasan, 1976) which define taxonomies of discourse relations, and we discuss the differences between these schemes in Section 5.2. Once such taxonomy, and the one that we follow in this thesis, is the one used in the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), which defines a hierarchy of relations. The four types of relation which comprise the top level of this hierarchy are briefly described below, and the full hierarchy is present as Appendix D:

Temporal Used to label examples of temporal ordering. E.g. *'Add the eggs, then stir in the butter'*.

Contingency Used to label causal or conditional relations. E.g. *'The mixture will split if heated too quickly'*.

Comparison Used to label contrast or concession. E.g. *'John walked. Gary took the bus'*.

Expansion Used to label elaboration of ideas. E.g. *'This food is rubbish. I wouldn't even feed it to my dog.'*

In general, the automatic classification of explicit discourse relations is easier than implicit discourse relations, because explicit connectives are relatively unambiguous. In Pitler et al. (2008), a decision tree classifier using only the connective as a feature is used to classify explicit relations into one of the top 4 classes of the PDTB hierarchy, achieving an accuracy of 93.09%[1]. The current state-of-the-art implicit discourse relation classifier, however, scores an accuracy of 57.55%.

Implicit discourse relations are challenging to classify for a number of reasons. Their interpretation can require world knowledge — in Example 5.5 one needs to know the connection between rain and umbrella to interpret that this relation is a cause. Also, a

[1]In languages other than English, explicit relations can be more ambiguous. For example, Al-Saif and Markert (2011) performed the same experiment as Pitler et al. (2008) but in Arabic, and achieved a significantly lower accuracy of 82.7%.

degree of logical inference may be required — in Example 5.6, one must deduce that cooking the dishes mentioned in the first argument would be worthy of praise to interpret the causal relation[2].

(5.5) It was raining. I got out my umbrella.

(5.6) Nine of the hottest chefs in town fed the executives Indiana duckling mousseline, lobster consomme, and veal mignon. The executives gave the chefs a standing ovation.

The specific hypothesis that underpins this chapter is that the presence of an entity instantiation is predictive of the discourse relation *Expansion.Instantiation*. The discourse relation Expansion.Instantiation is defined as follows:

> "The tag "Instantiation" is used when the connective indicates that Arg1 evokes a set and Arg2 describes it in further detail. It may be a set of events, a set of reasons, or a generic set of events, behaviors, attitudes, etc. Typical connectives often tagged as Instantiation are *for example*, *for instance* and *specifically*."
>
> (The Penn Discourse Treebank 2.0 Annotation Manual, The PDTB Research Group, 2008, p. 34)

We observed that instantiations of events, reasons, behaviours and attitudes often co-occur with instantiations of entities. For example, in the discourse relation in Example 5.7 the event of 'The Eurovision Song Contest being boring' is a part of the set of events 'attempts to produce pan-European TV programs resulting in disappointment'. Concurrently, the entity 'The Eurovision Song Contest' participates in an entity instantiation with ' "pan-European" TV programs'.

(5.7) a. Attempts to produce **"pan-European" TV programs** have generally resulted in disappointment.

b. *The Eurovision Song Contest, one such program,* has been described as the world's most boring TV show.

We focus solely on implicit discourse relations in this thesis. The main reason for this is that the majority of explicit examples of the Expansion.Instantiation relation in the PDTB are unambiguous. 194 of the 302 examples are marked by the connective *'for*

[2]Example 5.6 is adapted from file `wsj_0010` of the PDTB

example', which is used to mark a relation other than Expansion.Instantiation just twice in the corpus. A further 98 examples are marked with the connection *'for instance'*, which never signals a relation other than Expansion.Instantiation. Whilst the remaining 10 examples in the corpus use more ambiguous connectives, the connective is a very good predictor of explicit Expansion.Instantiation relations in the vast majority of cases, and therefore we concentrate solely on implicitly understood Expansion.Instantiation relations.

In the remainder of this Chapter, we discuss some background literature related to discourse relations, before presenting a corpus study confirming a strong correlation between entity instantiations and the Expansion.Instantiation discourse relation. We then develop a strong baseline discourse relation classifier, based on the feature set described in Sporleder and Lascarides (2008), and subsequently augment it with gold standard entity instantiation data and machine-generated entity instantiation data.

5.2 Background

In this chapter we use entity instantiations to aid the classification of discourse relations. Although discourse relations are not the primary focus of this thesis, in this Section we present a short summary of the field and survey of some current state of the art methods for discourse relation classification.

The study of discourse relations has been undertaken in many languages, such as German (Versley, 2011), Turkish (Zeyrek and Webber, 2008), Hindi (Oza et al., 2009), Danish, Italian and Spanish (Buch-Kromann and Korzen, 2010). In this Section we focus solely on methods that relate to English.

5.2.1 Theories of discourse

There are a variety of views about the organisation and structure of discourse, which have developed over time in to a number of theories of discourse representation.

A variety of linguistic literature (Grimes, 1975; Longacre, 1976; Crothers, 1979; Fillmore, 1981; Mann and Thompson, 1986; Halliday and Hasan, 1976) identified the existence of discourse relations, although relations were referred to by differing names such as rhetorical predicates, coherence relations and conjunctive relations. Each author formulated a slightly different taxonomy of relations, based upon their observations.

One of the first full theories of discourse relations was presented in Hobbs (1985). In the paper, the author describes a full taxonomy of relations, along with a framework

that suggests how the relations may be identified. Also described is a hierarchical, tree-like structure in which relations exist not only between spans of text, but between other relations.

Over the last twenty years, possibly the most dominant theory of discourse structure has been Rhetorical Structure Theory (RST). In RST a discourse is represented as a hierarchical tree, of which the leaves are elementary discourse units (edus). Edus are non-overlapping text segments which cover the whole of a text. Also, RST introduces *nuclearity*, which is the idea that most relations takes place between a more important *nucleus* and a less important *satellite*. However, some relations, such as *contrast*, take place between equally important edus.

As alluded to above, the precise set of relations used in a given theory are often chosen somewhat arbitrarily. Conversely, in RST the set of relations can be defined according to the task. This second approach is no more satisfactory — given the seemingly incoherent example below, taken from Knott and Dale (1994), one could define the relation *inform-accident-and-mention-fruit* to connect them.

> *John broke his leg. I like plums.*

To address this situation, Knott and Dale (1994) attempt to motivate a set of coherence relations by combining linguistic phenomena and psychological motivations. This involved organising discourse connectives according to whether they were always, sometimes or never substitutable.

There are also shortcomings with RST's binary tree based representation of discourse, most notably the issue of crossed dependencies. Consider the example below, taken from Wolf et al. (2003):

> *There is a Eurocity train on Platform 1. Its destination is Rome. There is another Eurocity on Platform 2. Its destination is Zürich.*

The first and third sentences participate in a parallel relation and the second and fourth provide a contrast, which cannot be represented as a binary tree. Other theories of discourse organisation have attempted to redress this, and other perceived issues with RST.

Segmented Discourse Representation Theory (SDRT) (Asher and Lascarides, 2003) is one such theory, which uses acyclic graphs rather than trees. Similarly to RST, SDRT allows for relations to occur between relations, and assumes non-overlapping edus that have total coverage of the text.

The Discourse Graphbank (Wolf and Gibson, 2005) is a project which uses another scheme based on acyclic graphs. Again, non-overlapping edus are assumed, but a relation

named *same*, which implies the continuation of a text span rather than a discourse relation, is used to bypass the strict non-overlapping.

Other work has used a more lexically-grounded approach (Webber et al., 1999, 2003), using a discourse-level Lexicalised Tree-Adjoining Grammar (DLTAG) to produce trees which represent the structure of single relations, along with the discourse connectives which signal them.

This work led to the ethos of the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), in which no higher structure of relations is annotated and discourse relations are treated as atomic units. The arguments of explicit discourse relations need not be between adjacent spans, and they may overlap. Implicit relations are annotated between adjacent sentences, though the arguments do not need to be the whole of each sentence. Implicit relations across paragraphs, within sentences or between non-adjacent sentence, and between sentences already connected by an explicit relation are not annotated due to time and resource constraints. The PDTB is described further in Section 5.2.2.

Other research has focused on the existence and the use of connectives in text, rather than the formulation of taxonomies of discourse relations or development of theories of discourse organisation. In Fraser (1999) an effort is made to define exactly what constitutes a connective. In Hirschberg and Litman (1994), the authors use prosodical and part-of-speech information to discover connectives in speech and text.

The use of connectives for connecting speech acts rather than propositions is discussed in Van Dijk (1979), and Cohen (1984) use connectives to augment the building of argument understanding trees.

5.2.2 Corpora

Research into automatic approaches for discourse relation classification has depended heavily on corpora annotated with discourse structure, for both the training of supervised machine learning algorithms and the evaluation of methods for classification.

Early work such as Kurohashi and Nagao (1994) and Marcu (1997) created small corpora of RST trees, ranging from 5 to 9 texts to demonstrate their algorithms. One of the first corpora which allowed for some meaningful evaluation contained 90 texts and was again used to demonstrate an algorithm, in Marcu (1999).

These were followed by a publicly available corpus of RST trees known as the RST Discourse Treebank (RST-DT) (Carlson et al., 2002). This corpus contains 385 annotated discourse trees (1 per text), and over 176,000 words of text. The texts are a subset of the Wall Street Journal texts that make up the Penn Treebank (PTB) (Marcus et al., 1993).

The Discourse Graphbank (Wolf and Gibson, 2005) is a publicly available corpus, based on the acyclic graphs described in Section 5.2.1. It comprises 135 texts from Associated Press newswire and the WSJ, which form part of the TIPSTER corpus (Harman and Liberman, 1993).

The PDTB (Prasad et al., 2008) is another publicly available corpus annotated with discourse relations. The texts used are also a subset of the WSJ texts that form the PTB. It attempts to be theory neutral, by merely annotating relations between spans of text rather than building larger trees or graphs of discourse. The spans of text do not have to cover the complete text, and may overlap.

The PDTB contains 2,159 annotated texts — considerably more than the RST-DT, leading to a total of 40,600 annotated relations. Both Explicit — those marked with a connective such as *but, and, or because* — and Implicit relations are annotated. We use the PDTB as the basis of our research in this Chapter, both because it is the largest annotated corpus of discourse relations currently available, and because it overlaps with our corpus of entity instantiations.

5.2.3 Automatic approaches

A number of automatic approaches to the classification and identification of discourse relations have been postulated. These approaches may be organised into at least 4 categories; those which aim to parse a text into a hierarchical structure of RST trees, those which attempt to classify PDTB style atomic discourse relations without any assumed hierarchy, those which categorise connectives, and those which attempt to identify the extent of the arguments of discourse relations.

The last two categories, approaches which categorise connectives, such as Hutchinson (2004a), Hutchinson (2004b), Hutchinson (2005a) and Hutchinson (2005b), and approaches which identify the arguments of relations, such as Wellner and Pustejovsky (2007), Elwell and Baldridge (2008), Prasad et al. (2010) and Ghosh et al. (2012), are only tangentially related to our problem and are therefore not discussed further in this Section. We instead focus on work which is related to the classification of discourse relations into types, either dealing with them in a stand-alone atomic fashion, or classifying them as part of the process of constructing a hierarchical discourse structure.

Hierarchical discourse parsing algorithms. Firstly, we note that the intention of hierarchical discourse parsers is to produce a representation which comprehensively covers the whole text. This means that these methods treat both explicit and implicit relations,

without distinguishing between the two. Therefore, their performance on implicit relations alone is unclear, and difficult to compare to methods which focus solely on implicit relations.

Initially, algorithms based upon hand coded rules were proposed for the parsing of text into RST trees, and the labelling of relations (Kurohashi and Nagao, 1994; Corston-Oliver, 1998; Marcu, 1997). However, the lack of annotated discourse corpora at the time means that these approaches were evaluated on inadequately sized data sets, and no firm conclusion could be made about their usefulness.

The first algorithm that used machine learning methods and a reasonably sized corpus was that of Marcu (1999). The paper presents a discourse segmenter, which learns to divide the text into spans between which relations exist, and a shift-reduce discourse parser which learns to build an RST tree from a series of shift and reduce operations. The algorithms are trained and tested on a corpus of 90 texts, each containing an RST tree which represents the document, and perform modestly. The segmenter has high precision, but the best recall reported is 75.4%. One must also note that in RST there are no gaps or overlaps between spans, making the task somewhat less difficult. The labelling of relations achieves a best precision and recall of 72.4% and 62.8%, but performance drops as low as 13.0% and 34.3% depending on the corpus used. Learning curve experiments suggested that a larger amount of training data would benefit performance.

In Soricut and Marcu (2003), the authors experiment with sentence-level discourse parsing — RST trees that only span a single sentence. They use the RST-DT (Carlson et al., 2002) as their corpus. Again, the task is split into two; discourse segmentation and discourse parsing. The segmenter improves over previous efforts by using syntactic information, gaining an F-Score of 85.4%. The parser is a probabilistic, bottom up search algorithm which has an F-Score of 49%. An interesting finding is that the quality of the segmentation is vital — using perfect segmentation causes the F-Score to rise to 63.8%.

Other discourse parsers include that of Hernault et al. (2010), who employ a pair of SVM classifiers for their HILDA parser. The first classifier decides whether two input sub-trees have a connecting node, and the second labels the relation and its nuclearity. They attain an F-Score of 54.8 on the test subset of the RST-DT. They also test on 10 doubly annotated RST-DT texts, allowing for comparison with human agreement. On this set, they achieve an F-Score of 55.1, compared with a human agreement score of 65.3.

Feng and Hirst (2012) build upon the HILDA parser, following the same two step method but suggesting additional features for each classifier. These features include cue phrases, word pairs, nearby discourse relations, discourse production rules and semantic

similarity metrics. They achieve a macro-averaged F-Score of 0.440 and an accuracy of 65.3%, compared to the 35.8% accuracy of a majority baseline.

Baldridge and Lascarides (2005) and Baldridge et al. (2007) use the framework of SDRT, rather than RST. In Baldridge and Lascarides (2005) they modify the head-driven lexicalised syntactic parser of Collins (2003) to work for dialogues, and gain a highest F-Score of 46.3, compared to inter-annotator agreement F-Score of 53.7. In Baldridge et al. (2007), they experiment on newswire text, this time adapting a dependency parse framework.

Discourse relations without hierarchy. Many authors have focused on the identification of single discourse relations, without attempting to create some hierarchical tree or graph framework to contain them.

One early machine learning attempt that did not use RST was that of Marcu and Echihabi (2001). In an effort to negate the lack of a large annotated corpus of implicit relations at the time, they attempted to automatically harvest training instances and classify them into one of Contrast, Cause, Elaboration, Condition, No-Relation-Same-Text, No-Relation-Different-Text. Hand-written patterns were used to find unambiguous explicit relations, signalled by connectives. The connectives are then removed to create implicit training examples.

The implicit training examples are used to train a word pair model. Given two spans the features are the word pairs which are the result of the Cartesian product between the two spans — $(w_i, w_j) \in W_1 \times W_2$. A 6-way classifier achieved an accuracy of 49%, and individual binary classifiers (e.g. Contrast vs $\neg$ Contrast) reached accuracies of over 85%.

These results suggest that this data-intensive approach with simple features is very useful for classifying implicit relations. However, further work (Sporleder and Lascarides, 2008; Sporleder, 2007) has found that these automatically generated training examples are not representative of naturally occurring implicit relations. In Sporleder and Lascarides (2008), the author experiments with training on automatically collected examples and testing on naturally occurring data and achieves an accuracy of 24.5%. In Sporleder (2007), the author uses the same model and experiments with augmenting natural training data with these automatic examples, and is unable to achieve any increase in performance.

The first work in classifying the discourse relations of the PDTB was Pitler et al. (2008). In this paper, a decision tree classifier using only the connective as a feature is used to classify *explicit* relations into one of the top 4 classes of the hierarchy (see Appendix D). This approach achieved an accuracy of 93.09%. The paper also presents significance tests and perplexity calculations, suggesting that sequences of discourse re-

lations might be helpful for classification.

The high accuracy of Pitler et al. (2008)'s algorithm means that most subsequent work has focused on implicit relations, rather than attempting to improve on an already very good explicit classifier.

Pitler et al. (2009) concentrated on classifying implicit relations in the PDTB. This paper compared word pair features with a variety of other features. Word pairs are actually found to be more useful in capturing function word co-occurrences rather than content word pairs such as $(popular, oblivion)$ or $(rain, rot)$ which suggest contrast and cause respectively. This could be due to data sparseness — function words will occur much more often than other useful pairings. Various binary classifiers are employed, and it is found that polarity is more useful for classifying Contingencies than Contrasts, and that the tokens that are found at the beginning and end of sentences are particularly useful. However, none of these features lead to a performance better than the baseline of assigning the majority class.

Following this, a number of other research has considered implicit relations in the PDTB. Lin et al. (2009) focus on the 11 most frequent members of the second level of the PDTB hierarchy. They use features describing nearby discourse relations, constituency and dependency parse tree production rules, and word pairs. Their maximum entropy classifier scores an accuracy of 40.2%, compared with a majority baseline accuracy of 26.1% and a random baseline accuracy of 9.1%.

As explicit relations are much easier to identify, Zhou et al. (2010) propose using a language model to predict the connective between the arguments of two implicit arguments, and include that as a feature. They use the top level of the hierarchy, and create binary relation classifiers. They achieve an 3% improvement in F-score over Pitler et al. (2009). These scores are further improved by Park and Cardie (2012), who use a greedy feature selection algorithm. Their highest scoring binary classifier, Expansion vs All gains an F-score of 79.22, with their lowest scoring classifier, Temporal vs All, still beating Pitler et al. (2009) and Zhou et al. (2010) with an F-score of 26.57.

Wang et al. (2010) employ tree kernels and temporal ordering information to classify both explicit and implicit discourse relations. They experiment with three tree kernels; one which represents the minimal syntactic structure which covers the arguments and connective if applicable, one which includes the first level children of intervening nodes, and one which includes all intervening nodes except the leaf nodes which represent the words. Their temporal ordering features extract events from each argument of a discourse relation, and calculates the order in which they occurred. The intuition is that in causal relations, the cause event usually happens temporally before the effect event. Their best

tree kernel — the one that includes first level intervening children — improves implicit relation accuracy by 9.7% over a simple baseline feature set, and the temporal ordering information adds 3.6% accuracy to the baseline. In combination, they score an accuracy of 40%, compared to a baseline performance of 29%. All experiments create 4-way classifiers based on the top level of the PDTB hierarchy.

Hong et al. (2012) use the web to aid discourse relation classification. They mine the web for argument pairs similar to the discourse relation arguments they are trying to classify, and extract the most frequently occurring connective for the top ranked similar argument pairs. This connective is then used as a major feature in the classification process, along with methods for filtering out *pseudo-cues* — results returned by the mining process that are not actually discourse connectives. Their algorithm scores 57.55% accuracy on four-way classification, and is the current state-of-the-art for implicit discourse relation classification.

This work closest to the work we present in this chapter is Louis et al. (2010b). They use entity features, including coreference, but *not* entity instantiations, for discourse relation classification. Their features include the grammatical role, information status, syntactic realisation and modification level of the entities involved. They develop binary classifiers at the top level of the PDTB hierarchy, and find that although their entity features beat the random baseline, simple word pair features perform better that the entity features.

Our work differs in a number of ways; Louis et al. (2010b) create binary classifiers for the top level of the PDTB hierarchy, whereas we focus on a single discourse relation, Louis et al. (2010b) employ features which detail coreferent entities, whereas we instead use entity instantiations. Finally, Louis et al. (2010b) present what is essentially a negative result, whereas we find gold standard entity instantiation information improves the classification of the Expansion.Instantiation discourse relation.

Differences between prior automatic discourse relation classification research and our work. In this Section we have summarised some recent relevant work in discourse relation classification. A number of themes run through this work; the use of word pairs (Marcu and Echihabi, 2001; Pitler et al., 2009; Lin et al., 2009), the prediction of the likely explicit connective (Zhou et al., 2010; Hong et al., 2012) and the use of tree production rules or kernels (Lin et al., 2009; Wang et al., 2010; Park and Cardie, 2012).

Our hypothesis is distinct from this work in at least two ways. Firstly, and uniquely, we use non-coreferent entity relationships as features for discourse relation classification, hypothesising that the relationship between the entities occurring within the arguments of

a discourse relation is important for its disambiguation. The only other discourse relation research which considers cross-argument entity relationships is Louis et al. (2010b), who only consider coreferent entities and find they do not perform better than word pairs.

Secondly, rather than creating 4-way classifiers, we narrow our scope and focus on one particular discourse relation — *Expansion.Instantiation*.

5.3 The Interaction between Discourse Relations and Entity Instantiations

Given our hypothesis connecting discourse relations and entity instantiations, we explored the relationship between the two phenomena in two ways.

Firstly, we calculated the correlation between the instantiations annotated in Chapter 3 and the relations in the PDTB, compiling statistics which describe the number of discourse relations with which each entity instantiation co-occurs.

Secondly, we annotated a set of PDTB discourse relations for the presence of entity instantiations. This annotated set shows a strong correlation between the discourse relation Expansion.Instantiation and entity instantiations. We use this set as the basis for our learning experiments that comprise the subsequent Sections of this chapter.

5.3.1 Entity instantiation and discourse relation co-occurrence

In our annotation study, described in full in Chapter 3, we intentionally selected texts which were part of the PDTB. This allows us to calculate statistics which describe the co-occurrence of entity instantiations and discourse relations.

Extent of overlap. The full PDTB spans 2,159 files, and comprises 40,600 annotated relations. The portion of the PDTB that overlaps with our entity instantiation corpus consists of 75 files and 4,182 relations. The distribution of the whole corpus and the overlapping portion are shown in Table 5.1. Explicit and Implicit relations have been discussed previously in this Chapter. However, the PDTB annotates several other related phenomena. AltLex relations, an abbreviation of *'Alternative Lexicalisation'*, are defined as follows:

> "...Cases where a discourse relation is inferred between adjacent sentences but where providing an Implicit connective leads to *redundancy in the expression of the relation*. This is because the relation is *alternatively lexicalized* by

Relation Type	**# in 75 EI texts**	**# in full PDTB**
Explicit	2 005 (47.94%)	18 459 (45.47%)
Implicit	1 714 (40.98%)	16 053 (39.54%)
AltLex	54 (1.29%)	624 (1.5%)
EntRel	384 (9.18%)	5 210 (12.83%)
NoRel	25 (0.60%)	254 (0.63%)
Total	4 182 (100.00%)	40 600 (100.00%)

Table 5.1: Distribution of discourse relations in 75 entity instantiation texts.

some "non-connective expression"."

(The Penn Discourse Treebank 2.0 Annotation Manual, The PDTB Research Group, 2008, p. 22)

These are closely related to implicit discourse relations, and as such we make no further distinction between them in this Chapter. Also annotated are Entity Relations (EntRel), where an entity-based coherence relations exist between two sentences, but no discourse relation exists, and No Relations (NoRel) where no discourse relation exists between two consecutive, paragraph-internal sentences. We do not include these non-relations in our study.

The distribution of the overlapping portion is a broadly representative sample of the full PDTB corpus, suggesting that conclusions drawn from this study are likely to be applicable on a larger corpus.

Entity instantiation and discourse relation co-occurrence — all relations. As defined in this Thesis, an entity instantiation exists between of two participant NPs. Discourse relations consist of two arguments, each of which spans one or more clauses and can cross sentence boundaries, as well as a connective for explicit relations. Due both to the restrictions placed on instantiations, and the restrictions placed on connectives in the PDTB, connectives and entity instantiations do not overlap. Similarly, the restrictions on the participants of instantiations and of discourse relation arguments mean that entity instantiation NPs can occur within the arguments of discourse relations, but *not* vice versa.

These restrictions leave us with three possible overlap scenarios:

Nesting An entity instantiation is nested within a discourse relation if both of its NPs are within a single argument of the discourse relation.

Spanning An entity instantiation spans a discourse relation if each of its NPs are in different arguments of the same discourse relation

Partial Overlap An entity instantiation partially overlaps with a discourse relation if exactly one of its NPs is within the argument of exactly one discourse relation argument.

The three types of overlap are shown in Examples 5.8, 5.9 and 5.10, respectively. Discourse relation arguments are delimited by square brackets, connectives are underlined.

(5.8) [The removal of Mr. Honecker was apparently the result of bitter infighting within the top ranks of the Communist party]. [According to West German government sources, ***Mr. Honecker*** **and several senior Politburo members** fought over the last week to delay any decisions about a leadership change. But, with public demonstrations in the country growing in size and intensity, Mr. Honecker and several key allies lost out in this battle].

(5.9) [Mr. Mason said that many Jewish voters feel guilty toward **blacks**, so they support black candidates uncritically]. [He said that *many black voters* feel bitter about racial discrimination, so they, too, support black candidates uncritically].

(5.10) [Third, the theory suggests why legislators who pay too much attention to national policy making relative to local benefit-seeking have lower security in office]. For example, [first-term members of the House, once the most vulnerable of **incumbents**, have become virtually immune to defeat]. The one exception to this recent trend was the defeat of *13 of the 52 freshman Republicans brought into office in 1980 by the Reagan revolution and running for re-election in 1982*.

To count co-occurrences we compare every positive and negative instantiation instance to every discourse relation within a given text, and therefore the total number of comparisons over a set of T texts is:

$$\sum_{t \in T} \big(\text{intersentential\_NP\_pairs}(t) + \text{intrasentential\_NP\_pairs}(t)) \times \text{discourse\_relations}(t)\big)$$

Each comparison has 3 possible outcomes; spanning, nesting or no co-occurrence. We classify partial overlaps as no co-occurrence, on the basis that they are a weaker link between the discourse relation and entity instantiation than the other two overlaps. Clearly, no co-occurrence is by far the most numerous category — a given entity instantiation will not co-occur with the majority of discourse relations in a text.

Table 5.2 shows the co-occurrence between all relations (implicit, explicit) and all instantiations (set members, subsets, intersentential, intrasentential). The difference in

Co-occurrence type	**Instantiation NP pair**		**Non-instantiation NP pair**	
Spans DR	1 023	(0.4%)	35 884	(0.6%)
Nested in DR	2 863	(1.1%)	48 141	(0.7%)
No co-occurrence	255 824	(98.5%)	6 350 327	(98.7%)
Total	259 710	(100.0%)	6 434 352	(100.0%)

Table 5.2: Entity instantiation and discourse relation co-occurrence: all instantiations, all relations.

the distribution of instantiation and non-instantiation NP pairs is significantly different ($\chi^2 = 533, p < 0.0001$). A greater percentage of instantiation NP pairs are nested within discourse relations than for non-instantiation NP pairs, when we consider all relations.

In Table 5.3, we perform the same calculations, but this time record whether the overlapping entity instantiation is intra- or intersentential. The results illustrate the fact that intrasentential entity instantiations are generally nested within discourse relations, and intersentential instantiations generally span discourse relations.

Table 5.4 shows the co-occurrence of set members and subsets with discourse relations. We see that both set members and subsets span discourse relations in similar proportions, with a slightly higher proportion of subsets being nested in a discourse relation.

Focus on implicit relations. As described earlier in this Chapter, explicit relations are much easier to classify than their implicit counterparts, due to the fact that explicit connectives are relatively unambiguous. We therefore recalculate our above statistics to focus on the phenomenon we intend to classify — implicit relations. Table 5.5 shows the distribution of the overlap between implicit and explicit relations. We find a larger proportion of implicit relations overlap with entity instantiations compared to explicit relations, and significant difference between instantiation pairs and non-instantiation pairs that co-occur with implicit instantiations ($\chi^2 = 332, p < 0.0001$).

We again see, in Table 5.6, that intrasentential instantiations are much more often nested than spanning discourse relations, and the converse is true for intersentential instantiations. Similarly, when when we examine the distribution of set members and subsets that overlap with implicit relations (Table 5.7), they display a similar pattern as for all relations — both set members and subsets span discourse relations in similar proportions, with a slightly higher proportion of subsets being nested in a discourse relation.

We also examine the type of the implicit relations that overlap with entity instantiations. Rather than use the very fine-grained entire schema, or the very coarse grained 4 categories at the top of the hierarchy, we use the 20 categories that make up the second

Co-occurrence type	Intra Instantiation NP pair		Intra Non-instantiation NP pair		Inter Instantiation NP pair		Inter Non-instantiation NP pair	
Spans DR	94	(0.1%)	5 851	(0.3%)	929	(0.7%)	30 033	(0.7%)
Nested in DR	2 780	(2.1%)	43 983	(1.9%)	83	(0.1%)	4 158	(0.1%)
No co-occurrence	132 586	(97.9%)	2 225 248	(97.8%)	123 238	(99.2%)	4 125 079	(99.2%)
Total	135 460	(100.0%)	2 275 082	(100.0%)	124 250	(100.0%)	4 159 270	(100.0%)

Table 5.3: Entity instantiation and discourse relation co-occurrence: all relations, inter vs intra breakdown.

Co-occurrence type	Set Member NP pair		Non Set Member NP pair		Subset NP pair		Non Subset NP pair	
Spans DR	675	(0.4%)	21 105	(0.5%)	348	(0.4%)	14 779	(0.6%)
Nested in DR	1 732	(1.0%)	27 898	(0.7%)	1 131	(1.3%)	20 243	(0.8%)
No co-occurrence	171 280	(98.6%)	3 854 493	(98.7%)	84 544	(98.3%)	2 495 834	(98.6%)
Total	173 687	(100.0%)	3 903 496	(100.0%)	86 023	(100.0%)	2 530 856	(100.0%)

Table 5.4: Entity instantiation and discourse relation co-occurrence: all relations, set member vs subset breakdown.

Co-occurrence type	**Instantiation NP pair**		**Non-instantiation NP pair**	
Spans implicit DR	739	(0.6%)	22 848	(0.7%)
Nested in implicit DR	1 712	(1.4%)	27 685	(0.9%)
No co-occurrence with implicit	123 289	(98.1%)	3 059 014	(98.4%)
Implicit total	125 740	(100.0%)	3 109 547	(100.0%)
Spans explicit DR	284	(0.2%)	13 036	(0.4%)
Nested in explicit DR	1 151	(0.9%)	20 456	(0.6%)
No co-occurrence with explicit	132 535	(98.9%)	3 291 313	(99.0%)
Explicit total	133 970	(100.0%)	3 324 805	(100.0%)

Table 5.5: Entity instantiation and discourse relation co-occurrence: all relations, implicit vs explicit breakdown.

level of the hierarchy[3]. We note that 7 types of relation do not occur at all in our study — all of these relations[4] occur 3 or fewer times as implicit relations in the entire corpus, and do not occur at all in the portion of the PDTB which overlaps with our entity instantiation corpus.

The results are shown in Table 5.8. We notice that the relation we hypothesised had a strong link to entity instantiations, *expansion.instantiation*, is one of two relations to be spanned by instantiations in a higher proportion than non-instantiations (0.141% vs 0.072%), with the other relation being the very infrequent *comparison.concession*.

Expansion.Instantiation. Our initial hypothesis, stated in Section 5.1, focused on a single discourse relation, *Expansion.Instantiation*. Therefore, we compute co-occurrence statistics that focus on this relation. Table 5.9 shows the overall co-occurrence between entity instantiations and Expansion.Instantiation. There is a significantly higher proportion of instantiations that co-occur than non-instantiations ($\chi^2 = 112, p < 0.001$).

Again we see a strong relationship between intrasentential instantiations and nesting, and intersentential instantiations and spanning (Table 5.10). However, the difference between the distributions of intrasentential instantiations and non-instantiations is not significant. The difference between the intersentential instantiations and non-instantiations is highly significant ($\chi^2 = 176, p < 0.0001$).

Table 5.11 shows the distributions of set members and subsets that overlap with Expansion.Instantiation relations. As before the distributions are quite similar.

[3]The full hierarchy is shown in Appendix D. The 20 categories we use are the 16 categories of the second level of the hierarchy, along with the 4 of the top level. This is because some discourse relations are unable to be annotated at the finer-grained level and so are given a label from the top level only.

[4]The relations that do not co-occur at all are *comparison.pragmatic_concession, comparison.pragmatic_contrast, contingency, contingency.condition, contingency.pragmatic_condition* and *temporal*.

Co-occurrence type	Intra Instantiation NP pair		Intra Non-instantiation NP pair		Inter Instantiation NP pair		Inter Non-instantiation NP pair	
Spans DR	17	(0.0%)	611	(0.1%)	722	(1.2%)	22 237	(1.1%)
Nested in DR	1 676	(2.6%)	26 010	(2.4%)	36	(0.1%)	1 675	(0.1%)
No co-occurrence	63 641	(97.4%)	1 071 819	(97.6%)	59 648	(98.7%)	1 987 195	(98.8%)
Total	65 334	(100.0%)	1 098 440	(100.0%)	60 406	(100.0%)	2 011 107	(100.0%)

Table 5.6: Entity instantiation and discourse relation co-occurrence: implicit relations, inter vs intra breakdown.

Co-occurrence type	Set Member NP pair		Non Set Member NP pair		Subset NP pair		Non Subset NP pair	
Spans DR	470	(0.6%)	13 435	(0.7%)	269	(0.6%)	9 413	(0.8%)
Nested in DR	1 025	(1.2%)	15 910	(0.8%)	687	(1.7%)	11 775	(1.0%)
No co-occurrence	82 766	(98.2%)	1 852 924	(98.4%)	40 523	(97.7%)	1 206 090	(98.3%)
Total	84261	(100.0%)	1882269	(100.0%)	41479	(100.0%)	1227278	(100.0%)

Table 5.7: Entity instantiation and discourse relation co-occurrence: implicit relations, set member vs subset breakdown.

Co-occurrence type	**Relation**	**Instantiation NP pair**		**Non-instantiation NP pair**	
Spans DR	comparison	0	(0.000%)	209	(0.007%)
	comparison.concession	9	(0.007%)	143	(0.005%)
	comparison.contrast	74	(0.059%)	2479	(0.080%)
	contingency.cause	157	(0.125%)	6914	(0.222%)
	contingency.pragmatic_cause	0	(0.000%)	14	(0.000%)
	expansion	6	(0.005%)	103	(0.003%)
	expansion.alternative	1	(0.001%)	116	(0.004%)
	expansion.conjunction	130	(0.103%)	4988	(0.160%)
	expansion.exception	4	(0.003%)	8	(0.000%)
	expansion.instantiation	177	(0.141%)	2234	(0.072%)
	expansion.list	1	(0.001%)	507	(0.016%)
	expansion.restatement	146	(0.116%)	3817	(0.123%)
	temporal.asynchronous	27	(0.021%)	984	(0.032%)
	temporal.synchrony	7	(0.006%)	332	(0.011%)
Nested in DR	comparison	3	(0.002%)	144	(0.005%)
	comparison.concession	29	(0.023%)	392	(0.013%)
	comparison.contrast	211	(0.168%)	2788	(0.090%)
	contingency.cause	455	(0.362%)	8173	(0.263%)
	contingency.pragmatic_cause	3	(0.002%)	26	(0.001%)
	expansion	1	(0.001%)	90	(0.003%)
	expansion.alternative	9	(0.007%)	141	(0.005%)
	expansion.conjunction	411	(0.327%)	5478	(0.176%)
	expansion.exception	2	(0.002%)	20	(0.001%)
	expansion.instantiation	164	(0.130%)	2535	(0.082%)
	expansion.list	33	(0.026%)	691	(0.022%)
	expansion.restatement	293	(0.233%)	5721	(0.184%)
	temporal.asynchronous	74	(0.059%)	1084	(0.035%)
	temporal.synchrony	24	(0.019%)	402	(0.013%)
No co-occurrence	comparison	357	(0.284%)	12185	(0.392%)
	comparison.concession	1492	(1.187%)	39439	(1.268%)
	comparison.contrast	11905	(9.468%)	300843	(9.675%)
	contingency.cause	38963	(30.987%)	954253	(30.688%)
	contingency.pragmatic_cause	412	(0.328%)	7459	(0.240%)
	expansion	436	(0.347%)	13986	(0.450%)
	expansion.alternative	1601	(1.273%)	37294	(1.199%)
	expansion.conjunction	20470	(16.280%)	505632	(16.261%)
	expansion.exception	34	(0.027%)	2743	(0.088%)
	expansion.instantiation	10978	(8.731%)	273667	(8.801%)
	expansion.list	3211	(2.554%)	79870	(2.569%)
	expansion.restatement	26451	(21.036%)	648151	(20.844%)
	temporal.asynchronous	5562	(4.423%)	147211	(4.734%)
	temporal.synchrony	1417	(1.127%)	36281	(1.167%)
Total	—	125740	(100.000%)	3109547	(100.000%)

Table 5.8: Entity instantiation and discourse relation co-occurrence: implicit relations, relation type breakdown.

Co-occurrence type	Instantiation NP pair		Non-instantiation NP pair	
Spans DR	177	(1.6%)	2 234	(0.8%)
Nested in DR	164	(1.4%)	2 535	(0.9%)
No co-occurrence	10 978	(97.0%)	273 667	(98.3%)
Total	11319	(100.0%)	278 436	(100.0%)

Table 5.9: Entity instantiation and discourse relation co-occurrence: Expansion.Instantiation

Summary. In summary, we find a strong relationship between implicit discourse relations and entity instantiations. Intrasentential instantiations tend to occur nested in the argument of a discourse relations, intersentential instantiations often span the arguments of a discourse relation. Set members and subsets both play a part in this relationship.

When we consider the discourse relation Expansion.Instantiation, we find a strong relationship between the discourse relation and intersentential instantiations which span it.

5.3.2 Annotating discourse relations for the presence of entity instantiations

To confirm the findings of the previous Section — that the Expansion.Instantiation discourse relation and entity instantiations are strongly related — and to provide training and testing data for subsequent supervised machine learning experiments, we annotated a set of discourse relations for the presence of entity instantiations. We felt that this extra annotation was necessary for because there were not enough implicit Expansion.Instantiation relations that overlapped with our current corpus for meaningful machine learning experiments — only 172 occur in our current corpus.

We annotated 1,000 relations, roughly half Expansion.Instantiations and half randomly selected other discourse relations — non-relations were not considered. The annotation was performed using a version of tool described in Chapter 3 modified to display discourse relation arguments, and carried out by a single annotator, the author of this thesis. Specifically, we annotated entity instantiations with the set NP in Arg1 of the discourse relation, and the member or subset NP within Arg2 of the discourse relations, on the basis that Arg1 of the relation is an abstract object representing a set of events or ideas.

Table 5.12 shows the number of instantiations found in the Expansion.Instantiation and Other relation types, and the number of NP pairs each comprises. We employed the Z-test for proportions to ascertain whether the proportion of NPs that were instantiations were significantly different between Expansion.Instantiation relations and Other

Co-occurrence type	Intra Instantiation NP pair		Intra Non-Instantiation NP pair		Inter Instantiation NP pair		Inter Non-Instantiation NP pair	
Spans DR	3	(0.0%)	24	(0.0%)	174	(3.3%)	2 210	(1.2%)
Nested in DR	163	(2.7%)	2 433	(2.5%)	1	(0.0%)	102	(0.1%)
No co-occurrence	5 871	(97.3%)	94 928	(97.5%)	5 107	(96.7%)	178 739	(98.7%)
Total	6 037	(100.0%)	97 385	(100.0%)	5 282	(100.0%)	181 051	(100.0%)

Table 5.10: Entity instantiation and discourse relation co-occurrence: Expansion.Instantiation, inter vs intra breakdown.

Co-occurrence type	Set Member NP pair		Non Set Member NP pair		Subset NP pair		Non Subset NP pair	
Spans DR	128	(1.6%)	1 293	(0.8%)	49	(1.4%)	941	(0.9%)
Nested in DR	119	(1.5%)	1 630	(1.0%)	45	(1.3%)	905	(0.8%)
No co-occurrence	7 620	(96.9%)	168 276	(98.3%)	3 358	(97.3%)	105 391	(98.3%)
Total	7 867	(100.0%)	171 199	(100.0%)	3 452	(100.0%)	107 237	(100.0%)

Table 5.11: Entity instantiation and discourse relation co-occurrence: Expansion.Instantiation, set member vs subset breakdown.

Relation Type	# Annotated	# of NP Pairs	# of Entity Instantiations	# of Plural-Singular NP Pairs	# of Set Members	# of Plural-Plural NP Pairs	# of Subsets
Exp.Instantiation	491	5892	642	4030	474	1862	168
Other	509	5080	233	3190	143	1890	90
All	1000	10972	872	7220	617	3752	258

Table 5.12: Distribution of entity instantiations over discourse relations

Category	Expansion.Instantiation	Other	Total
Has >1 entity instantiation	162	55	217
Has no entity instantiation	329	454	783
Total	491	509	1 000

Table 5.13: Presence of entity instantiations in discourse relations

relations, as shown in columns 3 and 4 of Table 5.12. We found the proportions significantly different, $p < 0.001$, for all instantiations, as well as for set members and subsets in isolation.

Table 5.13 compares the number of Expansion.Instantiation and Other discourse relations that have at least one instantiation. Again there is a significant difference between the distributions of Expansion.Instantiation relations and Other relations ($\chi^2 = 71, p < 0.0001$).

These statistics show a very strong predictive relationship between entity instantiations and Expansion.Instantiation discourse relations — a discourse relation is much more likely to be an Expansion.Instantiation if it contains an entity instantiation.

5.4 Baseline Discourse Relation Classification

We wished to demonstrate that features indicating the presence of entity instantiations improve discourse relation classification. Our discourse relation classifiers use the data that is annotated in Section 5.3.2. The classifiers are trained and tested on implicit discourse relations, and the output of the classifier is binary — Expansion.Instantiation or Other.

It was necessary to create a strong baseline discourse relation classifier, which we refer to as *Multifeature*, to which we could add features based on entity instantiations. In this Section we describe the features used (Section 5.4.1), the experimental set up (Section 5.4.3) and the performance (Section 5.4.4) of the Multifeature baseline classifier.

5.4.1 Feature set

We based our classifier upon the feature set described in Sporleder and Lascarides (2008). It would have been preferable to employ the state-of-the-art classifier, described in Hong et al. (2012), but our work preceded this publication.

The feature set of Sporleder and Lascarides (2008) is composed of 6 categories: Positional Features, Length Features, Lexical Features, Part-of-Speech Features, Temporal Features and Cohesion Features. We also implemented a set of our own, named Additional Features, and omitted some of the original Sporleder and Lascarides (2008) features

we found to be ineffective in preliminary discourse relation classification experiments. Each of these feature categories are explained below, and a full feature list is shown in Section 5.4.2.

Positional features. The positional features comprised a binary feature indicating whether the relation is inter- or intrasentential, and two features which represented whether the relation is near the beginning of near the end of a paragraph. A relation is judged to be "near the beginning" if it occurs in any of the first 25% of sentences in a paragraph containing 4 or more sentences. For sentences containing 3 or less sentences, it is near the beginning if it occurs in the first sentence. Identical rules, but with respect to the last 25% and final sentence were used for the "near the end" feature.

Length features. Two length features were used, one for the length of each argument in words.

Lexical features. Our lexical features comprise the unigrams of the two arguments as separate features. We also included the overlap between the words, stems, lemmas and content words of the two arguments as numerical features.

Part-of-Speech features. Two features represented the sequence of POS tags of each argument, and a numerical feature represented the overlap between them.

Temporal features. The temporal features were based around the extraction of *verbal complexes* and their classification in six ways. Verbal complexes (VCs) — single verbs or pairs of related verbs such as *'is completed'*, *'has been'* or *'will do'* (Lapata and Lascarides, 2004) — were extracted using TGrep2[5], a tree searching utility. Each VC was then classified in the following ways, on a per argument basis:

Finite If the VC is finite, one of {past, present}, $\emptyset$ otherwise.

Non-Finite If the VC is non-finite, one of {infinitive, ing-form, en-form}, $\emptyset$ otherwise.

Modality If the VC has a modal, one of {future, ability, possibility, obligation}, $\emptyset$ otherwise.

Aspect One of {imperfective, perfective, progressive}.

[5]TGrep2 is available from `http://tedlab.mit.edu/~dr/Tgrep2/`.

Voice One of {active, passive}.

Negation One of {affirmative, negative}.

Cohesion features. Our cohesion features consisted of a count of the first, second and third person pronouns of each argument.

Additional features. Our additional features comprised the following:

- We calculated polarity features, based on the subjectivity lexicon described in Wilson et al. (2009). For each word in an argument, we searched the lexicon, using a lemmatised form if the full word did not produce a match. Each word that is in the lexicon, and is therefore subjective, gets a score of 2 if it is *'strongly subjective'* and 0.5 if it is *'weakly subjective'*. Words whose polarity is negative have their score negated. The scores for an argument are summed, and then normalised by the number of subjective words in the argument. The scores were included as a feature, along with a binary feature indicating polarity change between arguments.
- A count of the number of date expressions in each argument.
- Cross argument features counting synonyms, antonyms and other WordNet relations between the words of the arguments.
- A count of the entities in each argument.

5.4.2 Full feature list

The full list of features used for discourse relation classification is listed below. The features marked with an asterisk are repeated for up to nine verbal complexes in each discourse relation argument.

Feature Name	Feature Type	Example Value
arg1words	text	Still , some analysts insisted that the worst of the inflation is behind
arg2words	text	" It increasingly appears that 1987-88 was a temporary inflation blip and not the beginning of a cyclical inflation problem , " argued Edward Yardeni , chief economist at Prudential-Bache Securities Inc. in New York

Feature Name	**Feature Type**	**Example Value**
wordoverlap	continuous	0.128205128205
intersentential	True, False	True
nearbeginning	True, False	True
nearend	True, False	False
lenarg1	integer	13
lenarg2	integer	35
stemoverlap	continuous	0.58064516129
lemmaoverlap	continuous	0.128205128205
cwoverlap	continuous	0.0454545454545
arg1pos	text	RB COMMA DT NNS VBD IN DT JJS IN DT NN VBZ RB
arg2pos	text	“ PRP RB VBZ IN CD VBD DT JJ NN NN CC RB DT NN IN DT JJ NN NN , ” VBD NNP NNP , JJ NN IN NNP NNP NNP IN NNP NNP
arg1pr1	integer	0
arg1pr2	integer	0
arg1pr3	integer	0
arg2pr1	integer	0
arg2pr2	integer	0
arg2pr3	integer	1
arg1vbcp1finite*	0, past, present, none	past
arg1vbcp1nonfinite*	0, infinitive, ing-form, en-form, none	0
arg1vbcp1modality*	0, future, ability, possibility, obligation, none	0
arg1vbcp1aspect*	imperfective, perfective, progressive, none	imperfective
arg1vbcp1voice*	active, passive, none	active
arg1vbcp1negation*	affirmative, negative, none	affirmative

Feature Name	**Feature Type**	**Example Value**
arg2vbcp1finite*	0, past, present, none	present
arg2vbcp1nonfinite*	0, infinitive, ing-form, en-form, none	0
arg2vbcp1modality*	0, future, ability, possibility, obligation, none	0
arg2vbcp1aspect*	imperfective, perfective, progressive, none	imperfective
arg2vbcp1voice*	active, passive, none	active
arg2vbcp1negation*	affirmative, negative, none	affirmative
arg1dates	integer	0
arg2dates	integer	0
arg1polarity	continuous	-0.153846153846
arg2polarity	continuous	-0.0142857142857
polaritychange	True, False	False
vsyn	integer	1
vant	integer	0
vother	integer	0
nsyn	integer	2
nant	integer	0
nother	integer	0
jsyn	integer	0
jant	integer	0
jother	integer	0
rsyn	integer	0
rant	integer	0
rother	integer	0
a1ents	integer	4
a2ents	integer	8

Algorithm	**Accuracy**	**Precision**	**Recall**	**F-Score**
Majority	50.9%	0.0000	0.0000	0.0000
Unigram	64.2%	0.6371	0.6293	0.6332
Multifeature	70.8%	0.7095	0.6863	0.6977

Table 5.15: Baseline classification of Expansion.Instantiation relations.

5.4.3 Experimental set-up

As in our prior experiments, we apply 10-fold cross-validation for testing and training. We also keep relations from the same text in the same fold, to avoid over-fitting based on specific topical unigrams that may occur in a single text.

We use the machine learner ICSIBoost (Favre et al., 2007), which we previously employed in Chapter 4, an open source implementation of Boostexter (Schapire and Singer, 2000) which combines simple 'rules-of-thumb' — in this case, decision stumps — to produce a classifier. For our experiments, we specify 500 rounds of boosting and use the "ngram" expert with a window size of two — i.e. all textual features are included as unigrams and bigrams.

5.4.4 Results

Table 5.15 shows the results of our discourse relation classification baseline, along with a baseline which predicts the majority relation in each fold, and a classifier whose two features are the unigrams of each argument. Precision, Recall and F-Score are reported with respect to the positive class, Expansion.Instantiation. We find that the Multifeature classifier performs significantly better (McNemar's χ^2 test, $p < 0.0001$) than the other two baselines.

5.5 Discourse Relation Classification with Gold Standard Entity Instantiations

Using our strong Multifeature baseline for comparison, we implemented three features based upon the gold standard annotations described in Section 5.3.2:

1. *ent_inst-binary*, a boolean which indicates whether any entity instantiations are present between the two arguments of the discourse relation.

2. *ent_inst-type*, which has four possible values, indicating the presence of a set member, subset, both or neither.

Algorithm	**Accuracy**	**Precision**	**Recall**	**F-Score**
Majority	50.9%	0.0000	0.0000	0.0000
Unigram	64.2%	0.6371	0.6293	0.6332
Multifeature	70.8%	0.7095	0.6863	0.6977
ent_inst-binary	69.9%	0.7004	0.6762	0.6881
ent_inst-type	68.9%	0.7227	0.5947	0.6525
ent_inst-count	69.8%	0.6998	0.6741	0.6867
ent_instantiation + ent_inst-type + ent_inst-count	70.1%	0.7437	0.5967	0.6621
Multifeature + ent_inst-binary	73.5%	0.7316	0.7271	0.7293
Multifeature + ent_inst-type	73.9%	0.7357	0.7312	0.7334
Multifeature + ent_inst-count	71.7%	0.7149	0.7047	0.7239
Multifeature + ent_inst-binary + ent_inst-type + ent_inst-count	73.0%	0.7297	0.7149	0.7222

Table 5.16: Classification of Expansion.Instantiation relations with gold standard entity instantiation features.

3. *ent_inst-count*, an integer count of the number of instantiations spanning the discourse relation.

Initially, we built four classifiers, whose sole features were either *ent_inst-binary*, *ent_inst-type* or *ent_inst-count*, or the combination of the three. We find no significant difference between the four classifiers. We also find that there is no significant difference between any of their performance and the performance of the Multifeature baseline — in other words the connection between entity instantiations and the discourse relation Expansion.Instantiation is so strong that the gold standard features on their own perform as well as the Multifeature algorithm. All four classifiers are significantly better than the Unigram baseline ($p < 0.05$).

Incorporating gold standard knowledge of the presence of an entity instantiation between the arguments of a discourse relation, in the form of the *ent_inst-type* feature, leads to a 2.1% improvement in accuracy over the multi-feature algorithm alone, which is significant at the 0.5% level. Including *ent_inst-binary* also leads to a significant improvement over the Multifeature algorithm. The inclusion of *ent_inst-count*, or the combination of all three features, does not lead to significant performance increase.

5.6 Discourse Relation Classification with Machine Identified Entity Instantiations

In Section 5.5, we demonstrated that the inclusion of gold standard entity instantiation based features lead to significant improvements over the baseline for discourse relation identification. We wished to apply our entity instantiation classifier, detailed in Chapter 4, to automatically identify entity instantiations within discourse relations.

5.6.1 Machine-identified entity instantiation features

We replicated our three gold standard features, described in Section 5.5, but instead calculated their values using our entity instantiation classifier. The three calculated features are referred to as *ML_ent_inst-binary*, *ML_ent_inst-type* and *ML_ent_inst-count*. The process for calculating their values was as follows:

1. For each discourse relation fold:
   (a) Train a binary set member classifier, using gold standard set member data from the other 9 folds.
   (b) Train a binary subset classifier, using gold standard set member data from the other 9 folds.
   (c) For each discourse relation in the test fold:
      i. Extract all singular-plural NP pairs spanning the discourse relation arguments.
      ii. Classify each NP pair as one of {set member, not}.
      iii. Extract all plural-plural NP pairs spanning the discourse relation arguments.
      iv. Classify each NP pair as one of {subset, not}.
      v. Calculate 3 feature values based on classification output.

The features are calculated as for the gold standard features, but based upon the classification output rather than gold standard data. For clarity, we repeat the feature definitions:

1. *ML_ent_inst-binary*, a boolean which indicates whether any entity instantiations are present between the two arguments of the discourse relation.
2. *ML_ent_inst-type*, which has four possible values, indicating the presence of a set member, subset, both or neither.

3. *ML_ent_inst-count*, an integer count of the number of instantiations spanning the discourse relation.

Our feature set for learning the instantiations has three minor differences from the one presented in Chapter 4. The differences are:

1. The omission of the *surface backwards* feature. We only annotated entity instantiations where the set is in Arg1 of the discourse relation and the member/subset is in Arg2, so the feature is not needed.

2. Due to the fact that some of the annotated relations were not part of the OntoNotes corpus, we were unable to use gold standard coreference data. Our salience features were recalculated, approximating coreference by judging two NPs with the same head noun as identical, as in Barzilay and Lapata (2008).

3. The use of the Google Web 1T corpus (Brants and Franz, 2006), rather than the Google search engine for the calculation of PMI feature. This was due to the closure of the University Research Program for Google Search, which provided an API for our queries in Section 4.1.5.2. The very large Web 1T corpus provided similar results.

5.6.2 Results

As for the gold standard features, we first tested the machine-identified features in isolation, before combining them with the Multifeature baseline. The results are shown in Table 5.17.

We find that our machine-identified features do not outperform the unigram or Multifeature baseline when used in isolation, and do not improve the Multifeature baseline when combined with it. Despite the fact that our machine-learned classifier performs reasonably well at identifying whether a pair of noun phrases constitute an entity instantiation, it is not accurate enough to help with discourse relation classification.

We believe there are two main reasons why this feature does not improve performance. Firstly, our feature indicates that an entity instantiation exists between the arguments of the discourse relation if *any* of the possible pairs of noun phrases are judged to be an entity instantiation. In a pair of long sentences, there can be tens of noun phrase pairs, and a single error propagates, negating the correct classification of other pairs. The use of a numerical feature was intended to remedy this — the classifier could learn that just one entity instantiation between sentences is not a sufficiently reliable predictor and more are

Algorithm	**Accuracy**	**Precision**	**Recall**	**F-Score**
Majority	50.9%	0.0000	0.0000	0.0000
Unigram	64.2%	0.6371	0.6293	0.6332
Multifeature	70.8%	0.7095	0.6863	0.6977
ML_ent_inst-binary	59.6%	0.6028	0.5193	0.5580
ML_ent_inst-type	60.5%	0.6311	0.4705	0.5391
ML_ent_inst-count	58.6%	0.5951	0.4908	0.5379
ML_ent_inst-binary + ML_ent_inst-type + ML_ent_inst-count	59.9%	0.6257	0.4562	0.5277
Multifeature + ML_ent_inst-binary	68.7%	0.7572	0.5336	0.6260
Multifeature + ML_ent_inst-type	68.7%	0.7572	0.5336	0.6260
Multifeature + ML_ent_inst-count	68.7%	0.7572	0.5336	0.6260
Multifeature + ML_ent_inst-binary + ML_ent_inst-type + ML_ent_inst-count	69.6%	0.7710	0.5418	0.6364

Table 5.17: Classification of Expansion.Instantiation relations with machine-identified entity instantiation features.

needed to be sure that an entity instantiation is really present. However, it does not aid classification.

Secondly, our Multifeature discourse relation classifier captures some of the common indicators of the presence of an entity instantiation — such as the unigrams *some* and *many* — leading to redundancy.

5.7 Conclusion

5.7.1 Summary

In this Chapter, we have explored the connection between discourse relations and entity instantiations. We first reviewed relevant discourse relation literature, covering linguistic theories of discourse, the development of discourse relation corpora, and automatic approaches to discourse relation classification. Utilising the fact that our entity instantiation corpus, described in Chapter 3, overlaps with the Penn Discourse Treebank, currently largest corpus of hand-annotated explicit and implicit discourse relations, we were able to demonstrate a relationship between the occurrences of discourse relations and entity instantiations in text. In particular, we demonstrate a strong relationship between inter-sentential entity instantiations and the relation Expansion.Instantiation.

Subsequently, we annotated the arguments of 1,000 discourse relations, approximately half of which were Expansion.Instantiation relations, for the presence of entity instanti-

ations. An analysis of the annotation showed clearly that entity instantiations occurred significantly more often between the arguments of Expansion.Instantiation relations than other discourse relations. We then use this annotated data set to explore the impact of entity instantiation related features on discourse relation classification.

We develop a binary discourse relation classifier, which distinguishes between Expansion.Instantiation relations and other discourse relations. A strong Multifeature baseline scores an accuracy of 70.8% on the problem. Gold standard entity instantiation features alone match this, scoring 70.1%, which is not significantly different from the Multifeature classifier. The combination of the baseline and gold standard entity instantiation features leads to an accuracy of 73.9% — significantly higher than either in isolation.

Our attempts to automatically identify entity instantiations within discourse relations, and then incorporate these automatically identified instantiations as features for discourse relation learning proved less fruitful. We suggest that this is due to propagation of classification errors, and some redundancy in the feature set.

5.7.2 Future work

In future, we intend to apply improved entity instantiation classification techniques, with the aim of recreating the results of the gold standard entity instantiation feature. A more sophisticated mechanism for aggregating the entity instantiation results and incorporating them, so as to prevent errors propagating, would also be a useful addition. One option would be to sum the probability or confidence scores outputted by the learner, to get an idea as to the reliability of the predictions made.

Another possibility is that of *joint-learning*. Rather than learning entity instantiations separately prior to learning discourse relations, we wish to explore the possibility of simultaneously learning and classifying the phenomena. A similar paradigm has been adopted by Somasundaran et al. (2009) for the joint learning of discourse relations and sentiment, and by Choi et al. (2006) who jointly learn opinion holders and opinion expressions.

We also intend to experiment with entity instantiation features for other discourse relations. Contingency.cause, expansion.conjunction and expansion.restatement frequently have entity instantiations nested within their arguments, and seem likely candidates for this process.

There are also a number of other possible applications for entity instantiations, which are discussed in more detail in Chapter 6.

Chapter 6

Conclusion

In this Chapter, we summarise the work of this Thesis. We revisit our hypotheses and discuss the wider impact of our work. We also discuss potential future work, both in terms of specific extensions to our current work, and wide-ranging related ideas.

6.1 Summary

In this Section, we revisit the hypotheses we set out in Section 1.4.2, summarising how they have been proven.

The introduction of a novel, untackled research problem — entity instantiations. In Chapter 2 we comprehensively discuss related literature, broadly grouped into 3 categories; Information Extraction, Context-independent Relation Extraction and Bridging Anaphora. We confirm the novelty of our problem — no other research considers entity instantiations in full. We also reflect on the important differences between our problem and the related literature, and consider related work as inspiration for our own methods.

The problem is well formulated, and can be reliably annotated. In Chapter 3, we describe in detail our formulation of the problem, defining clearly what constitutes an entity instantiation, and enumerating specific annotation rules and special cases. We carried out separate agreement studies for inter- and intrasentential instantiations, achieving good

agreement for each. Subsequently, we annotate a total of 75 texts inter- and intrasententially, identifying a total of 4,521 instantiations.

Supervised machine learning can automatically identify positive and negative examples of the phenomenon. In Chapter 4, we comprehensively demonstrate that entity instantiations can be identified by supervised machine learning. Our best algorithms for intrasentential instantiations score accuracies of over 97% and F-Scores of over 0.71 on original data. The classification of intersentential instantiations proved substantially harder, but on a balanced data set the best algorithms achieved an accuracy of over 69% and an F-Score of 0.65.

We also addressed a number of hypotheses concerning the impact of particular types of features on instantiation classification. In the feature ablation studies we carry out, we find that the removal of our *surface* feature category leads to significantly poorer results, with the exception of on intersentential, balanced subset data. Similarly, we found that the *salience* of the two potential participants in an instantiation was an important indicator, along with features which use world knowledge to establish links between the two potential participants.

Knowledge of the syntactic relationship between the two participants in an intrasentential entity instantiation aids classification. Whilst our unstructured syntax-based features do not make a significant difference to classification in all settings, we find the use of tree kernels — methods which learn directly from structured constituency parse tree data — provide classifiers which perform comparably to our entire extensive unstructured feature set. When we combine tree kernels and unstructured features, we achieve significantly higher performance than by using either in isolation.

A strong link exists between entity instantiations and the Expansion.Instantiation discourse relation. In Section 5.3, we demonstrate a clear link between entity instantiations and discourse relations. We calculate the overlap between our corpus and the relations of the Penn Discourse Treebank, and see a strong relationship between the two, and a particularly clear link between entity instantiations and the Expansion.Instantiation discourse relation. We also annotate 1,000 discourse relations for the presence of entity instantiations, and find that they occur significantly more often in Expansion.Instantiation relations than other relations.

Entity instantiation knowledge can improve the classification of Expansion.Instantiation discourse relations. We find, in Section 5.5, that features based on Gold Standard entity instantiation knowledge perform similarly to a strong Multifeature baseline. Additionally, combining the Multifeature baseline with our Gold Standard entity instantiation features produces a classifier which is significantly better than either in isolation. However, we found that machine generated entity instantiation data, based on the classifier introduced in Chapter 4, was not sufficiently accurate to provide improvements over the Multifeature baseline.

6.2 Impact of Limitations

In Section 1.3, we outlined four of the ways in which we had delimited our research problem. Here, we discuss their impact on our study, especially with reference to our application of classifying the discourse relation *Expansion.Instantiation*. We also give further context to our application, by detailing some practical, real-world applications of discourse relations, to which our work could contribute.

Anaphoric and non-anaphoric entity instantiations. In this thesis we chose to annotate and classify both anaphoric and non-anaphoric relationships, rather than restricting the study to solely anaphoric instances. Although a dedicated study of anaphoric entity instantiations would have had merit, we included non-anaphoric cases because we felt that the knowledge that an entity instantiation exists between two NPs is useful, regardless of the realisation.

In the case of the application of entity instantiations we tackle in Chapter 5, we find that our non-anaphoric cases are often indicative of an Expansion.Instantiation relation. Examples 6.1, 6.2 and 6.3 are amongst those examples of Expansion.Instantiation discourse relations that co-occur with *non-anaphoric* entity instantiations.

(6.1) a. Jim Beam print ads, however, strike different chords in **different countries**.

b. In *Australia, land of the outback*, a snapshot of Jim Beam lies on a strip of hand-tooled leather.

(6.2) a. . . . and **the auto makers** fell sharply as well.

b. *Daimler-Benz* dropped 12.5 to 710.5, Bayerische Motoren Werke dropped 10.5 to 543.5, and Volkswagen lost 7.1.

(6.3) a. The collapse of the span has provoked surprise and anger among **state officials**.

b. *Gov. George Deukmejian* called for an immediate investigation.

Whilst an anaphora-focused study would have had the advantage of being more closely linked to a linguistic phenomenon, it seems clear that it would have adversely affected our ability to identify the discourse relation.

Focus on set membership and subsets. Our focus in this thesis has been entirely on two relationships; set members and subsets. One motivation for choosing these two particular relationships was their connection to discourse relations, and the relation Expansion.Instantiation in particular. Although this appears quite a narrow application, the classification of implicit discourse relations is a difficult task, and our method provides a useful insight into one relation that is especially important for automatic summarisation. We discuss the relationship between discourse relations and summarisation further below.

In the course of our examination of entity instantiations, we also found that discourse relations *Contingency.Cause, Expansion.Conjunction* and *Expansion.Restatement* have intrasentential entity instantiations nested within their arguments with some regularity, suggesting that entity instantiations could also be helpful in identifying other discourse relations. More generally, the connection between entity-level and discourse-level phenomena is an interesting research topic, and in the future we hope that exploring other entity relations could further contribute to discourse relation identification. For instance, co-set-membership — where two entities belong to the same set — seems likely to be helpful for identifying contrast discourse relations. Example 6.4 below shows an implicit contrast relation where the underlined entities are in a co-set-membership relation.

(6.4) a. The U.S. wants the removal of what it perceives as barriers to investment

b. Japan denies there are real barriers

Although the application to discourse relations was our initial motivation, the frequency of the phenomenon, as well as the challenge in tackling it, meant that we considered it as a stand-alone problem. We also were motivated by the fact that general relationships, such as set membership and subsets, are not considered by the current RE literature, which instead focus on real-world relations between concrete entities. In Section 6.3.3 we enumerate some other possible future applications for entity instantiations.

Distance restriction. In this thesis, we limited our annotation and identification of entity instantiations to within sentences and between adjacent sentences. Our motivations for this were twofold.

On a practical level, it made sense to restrict the annotation in this first treatment of entity instantiations by localising the problem. Indeed, the restrictions that we made allowed for reliable annotation and acceptable agreement. Our other motivation for this restriction was again related to implicit discourse relations. Implicit discourse relations in the PDTB are also annotated between adjacent sentences, and mirroring their extent was sensible in view of our application.

In Section 3.7 we experimented with removing these restrictions, with the intention of both discovering how many entity instantiations were missed by our restrictions, and understanding the feasibility of restriction-free annotation. In terms of feasibility, we found the annotation challenging, and it is difficult to see how agreement could be reached without considerable training. We also found that a significant proportion of entity instantiations existed outwith our adjacent sentence restriction, and though coreference data could be used to identify those that were simply repeats of entity instantiations expressed locally, there were still a number that would need manual annotation to identify.

Limitations on participants. The main restriction placed on NPs in our annotation concerned their plurality. Only singular NPs were considered as possible set members, and only plural NPs were considered as possible sets and subsets. This restriction was enforced in order to avoid the chance of marking relationships other than set membership and subsethood, and also to avoid the drawing of entity instantiations from NPs where the set was in some way more than a simple grouping of entities.

This restriction excluded potentially valid sets that were expressed in a singular form, such as *family, set* and *group*. Whilst in future it may be sensible to rectify this by including some valid singular sets, this restriction did not prevent entity instantiations being useful for the identification of Expansion.Instantiation discourse relations.

Applications of discourse relations. One of the motivations for the limitations outlined above relates to an application of entity instantiations — the identification of discourse relations — which we have explored in Chapter 5 of this thesis. Our interest in discourse relations is motivated by their utility for a range of further applications.

Chief amongst these applications is the *automatic summarisation* of texts, in which a short summary of a single text or several documents is automatically generated, either by *extracting* the most relevant sentences or by creating an entirely new *abstract* (Das and

Martins, 2007). Several pieces of research have employed discourse relations or hierarchical discourse structures for automatic summarisation (Ono et al., 1994; Marcu, 1998; Wolf and Gibson, 2004; Uzêda et al., 2008; Louis et al., 2010a), based on the intuition that a knowledge of the way elements of a text relate to each other is helpful for evaluating which parts of the text are important. Of particular interest is Louis et al. (2010a), in which the authors find that sentences which are the first argument of an implicit Expansion.Instantiation strongly correlate with human-created extractive summaries, but sentences that are the second argument do not.

Other useful applications for discourse relations include relation extraction (Maslennikov and Chua, 2007), machine translation (Marcu et al., 2000) and judging the readability of texts (Pitler and Nenkova, 2008).

6.3 Future Work

In this Section, we discuss a variety of possible extensions to our work.

6.3.1 Corpus extensions

A number of extensions are possible to our corpus.

Increasing the size of the corpus. The corpus we introduce in this thesis is substantial; it spans 75 texts, and includes annotation of over 4,000 instantiations. However, research in the field of NLP has shown that larger amounts of data are highly beneficial, and aid in the development of better automatic identification methods and more sound statistical analysis (Banko and Brill, 2001).

Producing a corpus of similar dimensions to the PDTB would be likely to give us over 100,000 instantiations, allowing for supervised machine learning to better capture outlier cases and create better rules for automatic identification. Creating annotated data on that scale takes a great deal of time, effort and money; even a much more modest corpus of 150 texts may well give us a better understanding of the phenomenon.

Another option is to automatically retrieve extra instantiations using a semi-supervised learning approach, as we discuss in Section 6.3.2.

Experimenting with the annotation of different genres. Our decision to annotate Wall Street Journal newswire texts was well justified — the texts were also annotated as part of several other corpora, including the Penn Treebank, Penn Discourse Treebank,

OntoNotes, PropBank and NomBank. This overlap with existing annotated resources allows easy study of the interaction between entity instantiations and other phenomena, and gives us the opportunity to develop features for machine learning which use these annotations.

Whilst the texts annotated include both essays and summaries as well as news articles (see Section 3.8.1 for a discussion of the distribution of genres in the corpus), their origin means they share some common characteristics.

Firstly, the texts in the corpus tend to focus on real-world objects and things, making the identification of sets, members and subsets more straightforward than might be the case in other genres. Secondly, the texts are written in formal English, suitable for publication in a newspaper. They are well formatted and spelled, and written in grammatically correct English.

In future, experimenting with texts that do not share these properties could be interesting. Identifying instantiations within a philosophy text, or a novel, could raise additional challenges, and it is uncertain how the current annotation scheme would perform in these circumstances. Dealing with entity instantiations in less formal settings, such as web pages, blogs, tweets or even dialogue could be harder, but also may offer more future applications for our work.

In our annotation of newswire texts, we restricted the relationships annotated to those which occured either intrasententially or between adjacent sentences. Relaxing this restriction seems possible, albeit challenging, on our newswire texts. However, were we to consider texts of different genres, such as novels, which can span hundreds of pages, it is difficult to see how annotation could be feasible without distance restrictions similar to those employed in the current annotation.

Removing annotation restrictions. We imposed a number of restrictions on our annotations to make the process of identifying entity instantiations easier. Firstly, we limited the set NPs to plural NPs using the process specified in Section 3.5.3. This reduces the chances significantly of relationships such as meronymy, employment or location being mistakenly marked as instantiations, and avoids some of the vagaries of deciding whether an NP such as *'the parliament'* or *'the team'* is just a grouping of entities or is somehow more than the sum of its parts. However, the drawback to this decision is the potential omission of some valid entity instantiations. In the future, we are keen to see if reasonable agreement can be achieved without this restriction, and whether the annotation drifts away from the phenomenon we wish to identify. It may be that additional annotator training is needed to maintain the quality of the annotation.

Secondly, our annotation addressed intrasentential and sentence-adjacent intersentential annotation. In Section 3.7, we carried out a small pilot study, and found that totally unrestricted annotation was very difficult. An obvious middle-ground would be to restrict annotations to within a paragraph, but the paragraphs of the Penn Treebank WSJ corpus contain between 2–3 sentences on average (Webber, 2009), which would not add much to the current intersentential annotation. If we were to experiment with different genres of text, this level of restriction might be more useful. Another option would be to re-attempt the unrestricted annotation study, but develop a better annotation tool to aid the annotator in tracking the possible instantiations throughout a long text.

We also simplify our annotation by including generics as possible sets. This approach does not cause much difficultly, because of the tendency of the source materials to focus on real-world objects. Were we to attempt to annotate different genres of text with less of a focus on real-world objects, we could employ the methods of Reiter and Frank (2010), who use supervised machine learning to identify generic noun phrases, as a useful preprocessing step for our annotation.

Experimenting with other languages. In this Thesis, we have concentrated exclusively on the English language, considering only how entity instantiations occur in English texts. It would be interesting to see how the phenomenon occurs in different languages. Would manual annotation be more or less difficult? Is syntax as important for distinguishing intrasentential entity instantiations, or is this finding English-dependent? In languages that have more complex morphologies, such as Arabic, or lack sentence boundaries, such as Chinese, how would one go about identifying and learning entity instantiations?

6.3.2 Machine learning improvements

There are a number of possible avenues for future work that could lead to improvements in the performance of our entity instantiation classifier, which was introduced in Chapter 4.

World knowledge features. In Section 2.2, we discuss several methods for extracting context-independent relations, some of which use many corpus-based patterns to identify relations, and include algorithms which automatically identify new patterns and evaluate their utility. Our feature set uses a single one of Hearst (1992)'s patterns, which gives good results. Employing at least one of these more complex methods to better identify those entity instantiations that are based upon context-independent, well-known relationships (e.g. $France \in EU\ Countries$) is likely to improve results.

Another useful resource for discovering context-independent relations that could be employed as a feature is WikiNet (Nastase and Strube, 2013). WikiNet uses Wikipedia's category structure and infoboxes — article summaries containing structured data located in the top-right corner of Wikipedia pages — to construct a large scale concept network. This network could be searched for relations that overlap with a given entity instantiation.

Tree kernels. Due to the strong syntactic relationship between the participants of intrasentential entity instantiations, we employ tree kernels to learn directly from constituency parse trees. The tree representation we used is based on that of Zhang et al. (2006) and Swampillai and Stevenson (2011), coupled with the SubSet Tree (SST) kernel learner that is part of SVM-LIGHT-TK (Moschitti, 2006b), and it performs well. However, there are a number of other options that are worth exploring for tree kernel based classification.

Firstly, one might try using a different tree kernel. The SST kernel (Collins and Duffy, 2002) allows learning from more generalised, leaf-less internal sub-trees, when compared to the SubTree (ST) kernel, which learns from sub-trees containing all the descendants of the target root node until the leaves. There is however, an even more generalised option which may be useful, the Partial Tree (PT) kernel (Moschitti, 2006a), which allows learning from tree fragments which do not necessarily conform to production rules. These even more general substructures may lead to better tree kernel classification results, and this generality may also open up the possibility of using tree kernels for intersentential learning.

Secondly, one might try learning from a structure other than a constituency parse tree. Dependency trees are commonly used in relation extraction (Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; Zhao and Grishman, 2005), and could prove an interesting alternative, or addition, to our constituency tree kernel classifier.

Thirdly, one could apply tree kernels to intersentential entity instantiations. We have not done so in this thesis because our intuition was that syntactic relationships were much more important for intrasentential instantiations. However, tree kernels could also be useful to some degree for intersentential instantiations. At least two options are feasible; following Swampillai and Stevenson (2011) and joining together the trees of two sentences under a new node, and creating two individual kernels, one for the set-containing sentence and one for the set member/subset-containing sentence, before summing the results.

Graph-based and joint learning. In this thesis we consider the classification of entity instantiations as a *local* problem — each instance is treated separately, and although we

include some contextual features, the classification of other nearby entity instantiations does not affect the process. However, our intuition suggests that a global classification approach might produce better results. In an example, such as Example 6.5, knowing that *'the UK'* is an instantiation of *'Several countries'* means that the entities it appears in conjunction with, *'France'* and *'Spain'* are more likely to also be instantiations of the same set. At an even simpler level, knowing that a set has had one instantiation drawn from it may make it more likely to be used again.

(6.5) a. **Several countries** attended the conference, including *the UK*, France and Spain.

b. Iceland arrived late.

Given these considerations, we might find representing instantiations as a network or *graph* useful. Figure 6.1 shows a possible representation, with nodes representing possible instantiations, and edges representing members participating in conjunctions and sharing sets. In a model such as this, the nodes would be classified by taking into account the local probability that each NP pair is an instantiation, as well as the global links between NP pairs.

This sort of learning also provides the potential to learn inter- and intrasentential and set member and subset relations simultaneously, which could certainly be advantageous Potential tools for carrying out such learning experiments include NetKit-SRL (Macskassy and Provost, 2007) and Alchemy (Domingos et al., 2006), a Markov-Logic based statistical relational learner.

Semi-supervised learning. In Section 6.3.1 we discussed the potential advantages of a bigger corpus. Rather than manually annotating new examples, it may be possible to employ a semi-supervised learning algorithm to automatically leverage unlabelled data for training. At least two options are possible: self-training and graph-based semi-supervised learning.

In self-training, a classifier is trained on a small amount of labelled data, which then classifies a larger, unlabelled data set. The most confident predictions from the unlabelled set are included in the training set, the classifier is retrained and the procedure is repeated for further unlabelled data (Zhu, 2005). One could apply this method to the rest of the texts in the PDTB, or other newswire texts, to identify further examples of entity instantiations. However, with this method errors can propagate, and a very precise classifier may be required.

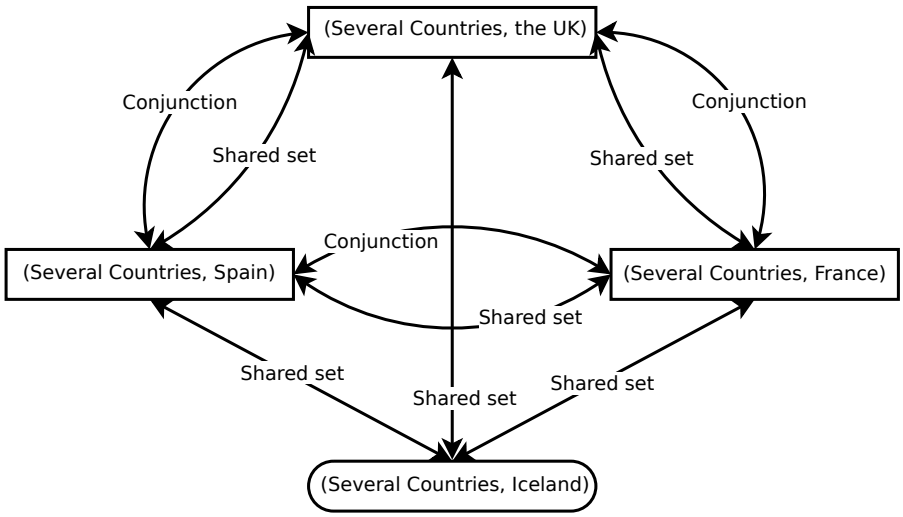


Figure 6.1: A graphical representation of the instantiations in Example 6.5.

Previously in this Section, we have described the notion of graph-based learning, in which instances are represented as nodes on a graph, with edges representing relationships between them. Using this paradigm, it is possible to have unlabelled nodes, that are labelled based on their incoming edges. The edges we previously suggested (coreferring sets and members and conjunctions) were based on relationships within a document, but because we annotated full texts, there are no document-internal unlabelled intrasentential or sentence-adjacent intersentential instances. We could, however, use this method to identify instantiations between non-adjacent sentences. Alternatively, we could use edges that represent cross-document relationships, such as coreference or synonymy between sets and members/subsets, to classify instantiations in unannotated documents.

Unsupervised learning. In Section 2.1.4, we discussed unsupervised learning for RE, in which the taxonomy of relations are not pre-specified, but are instead discovered from the data, usually in some sort of clustering process. In the case of RE, the relations were restricted between specific entity types, such as PERSON-GPE, and relations discovered included *President, Governor* and *Senator* (Hasegawa et al., 2004).

In this thesis, we did not explore these techniques, as we wished to focus solely on entity instantiations, rather than having to deal with the other sorts of relations an unsupervised method could discover. However, we too have restrictions between entity types, in terms of restricting ourselves to plural-plural or plural-singular NP pairs. An unsupervised experiment on this basis could lead to the automatic discovery of entity instantiations, and could also help us identify other important non-instantiation relationships which occur be-

tween NP pairs. This knowledge could also then feed into our feature creation process: if we know what other non-instantiation relationships are common, we can develop features which attempt to preclude them from being classified as instantiations.

6.3.3 Applications

Discourse relations. In Chapter 5, our main focus was on the discourse relation *Expansion.Instantiation* and its relationship to entity instantiations. However, there are other discourse relations which may similarly be linked to entity instantiations, such as *Contingency.Cause, Expansion.Conjunction* and *Expansion.Restatement*, which frequently have entity instantiations nested within their arguments

Due to the link between discourse relations and entity instantiations, one may also consider learning the two phenomena jointly, in a similar way to the joint learning of differing types of entity instantiations describe in Section 6.3.2.

Connection to anaphora resolution. One important avenue of future work is exploring the connection between entity instantiations and other entity-level phenomena, both in terms of the impact entity instantiation data may have on their classification, and also in terms of the impact these phenomena may have on entity instantiation classification. We discussed in detail in Section 2.3 the connection between bridging anaphora and entity instantiations; some entity instantiations are bridged, and entity instantiation knowledge could be useful in their resolution.

Sentiment analysis. Entity instantiations could aid the interpretation of sentiment in text. As stated in Chapter 3, our primary principle for identifying instantiations is that we require all statements that apply to the set — excluding cases where the statements could not apply to an individual member of the set, but instead describe the nature of the set — to also hold true for the member/subset. This means that any sentiment applied to the set is also true for the member/subset. For phrase-based sentiment analysis, the presence of an entity instantiation could therefore give important contextual information.

Summarisation. A common form of intersentential entity instantiation, especially prevalent in the WSJ corpus, is that shown in Example 6.6. The general pattern is that a statement is made in the sentence-containing set, and elaborated on by means of an example in the member-containing set.

(6.6) a. But **other analysts** said that having Mr. Phillips succeed Mr. Roman would make for a smooth transition.

b. "Graham Phillips has been there a long time, knows the culture well, is aggressive, and apparently gets along well with Mr. Sorrell", said *Andrew Wallach, an analyst with Drexel Burnham Lambert.*

In examples such as these, the second sentence is unlikely to contain information that would be required in a summary. Therefore, we suggest that knowledge of entity instantiations would be useful for automatic summarisation.

Knowledge extraction. Knowledge extraction is the process of automatically extracting structured knowledge from data, often with the goal of creating an ontology. Our definition of entity instantiations includes complex noun phrases and context-dependent relationships, but a portion of our entity instantiations represent concrete, context independent facts. Entity instantiation identification could therefore serve as an important pre-processing step in ontology construction.

Bibliography

ACE. Automatic Content Extraction. `http://www.ldc.upenn.edu/Projects/ACE/`, 2000-2005.

E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM Conference on Digital libraries*, pages 85–94. Association for Computing Machinery, 2000.

D. Ahn. The stages of event extraction. In *Proceedings of the 2006 COLING/ACL Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, 2006.

A. Al-Saif and K. Markert. Modelling discourse relations for arabic. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 736–747. Association for Computational Linguistics, 2011.

C. Aone, L. Halverson, T. Hampton, and M. Ramos-Santacruz. Sra: Description of the IE2 system used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

D. E. Appelt, J. R. Hobbs, J. Bear, D. Israel, and M. Tyson. FASTUS: a finite-state processor for information extraction from real-world text. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1172–1172, 1993.

N. Asher. *Reference to abstract objects in discourse*. Kluwer Academic (Dordrecht and Boston), 1993.

N. Asher and A. Lascarides. *Logics of conversation*. Cambridge University Press, 2003.

N. Asher and A. Lascarides. Bridging. *Journal of Semantics*, 15(1):83–113, 1998.

D. Ayuso, S. Boisen, H. Fox, H. Gish, R. Ingria, and R. Weischedel. BBN: Description of the PLUM system as used for MUC-4. In *Proceedings of the 4th Message Understanding Conference (MUC-7)*, pages 169–176. Association for Computational Linguistics, 1992.

J. Baldridge and A. Lascarides. Probabilistic head-driven parsing for discourse structure. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 96–103. Association for Computational Linguistics, 2005.

J. Baldridge, N. Asher, and J. Hunter. Annotation for and robust parsing of discourse structure on unrestricted texts. *Zeitschrift fur Sprachwissenschaft*, 26(213-239), 2007.

M. Banko and E. Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33. Association for Computational Linguistics, 2001.

M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *Proceedings of the 2007 International Joint Conference on Artificial Intelligence*, pages 2670–2676, 2007.

R. Barzilay and M. Lapata. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34, 2008.

S. Baumann and A. Riester. Referential and lexical givenness: Semantic, prosodic and cognitive aspects. In G. Elordieta and P. Prieto, editors, *Prosody and Meaning*, number 25 in Interface Explorations. Mouton de Gruyter, Berlin, 2011.

M. Berland and E. Charniak. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 57–64. Association for Computational Linguistics, 1999.

S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly Media, 2009.

K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, 2008.

T. Brants and A. Franz. *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia, 2006.

S. Brennan, M. Friedman, and C. Pollard. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 155–162. Association for Computational Linguistics, 1987.

S. Brin. Extracting patterns and relations from the world wide web. *The World Wide Web and Databases*, pages 172–183, 1999.

M. Buch-Kromann and I. Korzen. The unified annotation of syntax and discourse in the Copenhagen Dependency Treebanks. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 127–131. Association for Computational Linguistics, 2010.

R. Bunescu and R. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics, 2005.

R. Bunescu and R. Mooney. Subsequence kernels for relation extraction. *Advances in Neural Information Processing Systems*, 18:171, 2006.

C. Cardie. A case-based approach to knowledge acquisition for domain-specific sentence analysis. In *Proceedings of the National Conference on Artificial Intelligence*, pages 798–798. John Wiley & Sons LTD, 1993.

C. Cardie and K. Wagstaff. Noun phrase coreference as clustering. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 82–89, 1999.

A. Carlson, C. Cumby, J. Rosen, and D. Roth. The SNoW learning architecture. Technical report, University of Illinois at Urbana-Champaign, Department of Computer Science, 1999.

L. Carlson, B. Onyshkevych, and M. Okurowski. Corpora and data preparation. In *Proceedings of the 5th Message Understanding Conference*, pages 1–5. Association for Computational Linguistics, 1993.

L. Carlson, D. Marcu, and M. Okurowski. RST discourse treebank. *Linguistic Data Consortium, Philadelphia*, 2002.

S. Cederberg and D. Widdows. Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 111–118. Association for Computational Linguistics, 2003.

Y. S. Chan and D. Roth. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 152–160. Coling 2010 Organizing Committee, 2010.

N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. SMOTE: synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357, 2002.

J. Chen, D. Ji, C. Tan, and Z. Niu. Unsupervised feature selection for relation extraction. In *Proceedings of the International Joint Conference on Natural Language Processing.*, 2005.

H. Chieu, H. Ng, and Y. Lee. Closing the gap: Learning-based information extraction rivaling knowledge-engineering methods. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 216–223. Association for Computational Linguistics, 2003.

N. Chinchor. Overview of muc-7/met-2. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Association for Computational Linguistics, 1998.

T. Chklovski and P. Pantel. VerbOcean: mining the web for fine-grained semantic verb relations. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 33–40. Association for Computational Linguistics, 2004.

Y. Choi, E. Breck, and C. Cardie. Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439. Association for Computational Linguistics, 2006.

H. H. Clark. Bridging. In *Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing*, pages 169–174. Association for Computational Linguistics, 1975.

J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

R. Cohen. A computational theory of the function of clue words in argument understanding. In *Proceedings of the 10th International Conference on Computational Linguistics and 22nd Annual Meeting of the Association for Computational Linguistics*, pages 251–258. Association for Computational Linguistics, 1984.

M. Collins. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637, 2003.

M. Collins and N. Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics, 2002.

M. Collins and S. Miller. Semantic tagging using a probabilistic context free grammar. In *Proceedings of the Sixth Workshop on Very Large Corpora*. Association for Computational Linguistics, 1998.

S. Corston-Oliver. Beyond string matching and cue phrases: Improving efficiency and coverage in discourse analysis. In *The AAAI Spring Symposium on Intelligent Text Summarization*, pages 9–15, 1998.

J. Cowie. Automatic analysis of descriptive texts. In *Proceedings of the First Conference on Applied Natural Language Processing*, pages 117–123. Association for Computational Linguistics, 1983.

E. Crothers. *Paragraph structure inference*. Ablex Publishing Corporation, 1979.

S. Cuendet, D. Z. Hakkani-Tür, E. Shriberg, J. G. Fung, and B. Favre. Cross-genre feature comparisons for spoken sentence segmentation. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007)*, pages 265–274. IEEE Computer Society, 2007.

A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, page 423. Association for Computational Linguistics, 2004.

I. Dagan, O. Glickman, and B. Magnini. The PASCAL Recognising Textual Entailment challenge. *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, 2006.

D. Das and A. F. Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4:192–195, 2007.

H. Daumé III and D. Marcu. A large-scale exploration of effective global features for a joint entity detection and tracking model. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 97–104. Association for Computational Linguistics, 2005.

G. DeJong. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3(3):251–273, 1979.

Q. Do and D. Roth. Constraints based taxonomic relation classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1099–1109. Association for Computational Linguistics, 2010.

P. Domingos, S. Kok, H. Poon, M. Richardson, and P. Singla. Unifying logical and statistical AI. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 2–7, 2006.

R. Elwell and J. Baldridge. Discourse connective argument identification with connective specific rankers. In *2008 IEEE International Conference on Semantic Computing*, pages 198–205. IEEE, 2008.

O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91 – 134, 2005.

B. Favre, D. Hakkani-Tür, and S. Cuendet. ICSIboost. `http://code.google.come/p/icsiboost`, 2007.

B. Favre, D. Hakkani-Tur, S. Petrov, and D. Klein. Efficient sentence segmentation using syntactic features. In *Proceedings of the Spoken Language Technology Workshop, 2008*, pages 77 –80, 2008.

T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

C. Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

V. Feng and G. Hirst. Text-level discourse parsing with rich linguistic features. In *Proceedings of the The 50th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2012.

C. Fillmore. Pragmatics and the description of discourse. *Radical pragmatics*, pages 143–166, 1981. New York: Academic Press.

J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual*

Meeting of the Association for Computational Linguistics (ACL'05), pages 363–370. Association for Computational Linguistics, 2005.

B. Fraser. What are discourse markers? *Journal of pragmatics*, 31(7):931–952, 1999. Elsevier.

K. Fraurud. Definiteness and the processing of noun phrases in natural discourse. *Journal of Semantics*, 7(4):395, 1990. Oxford University Press.

D. Freitag. Toward general-purpose learning for information extraction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 404–408. Association for Computational Linguistics, 1998.

D. Freitag and A. McCallum. Information extraction with HMM structures learned by stochastic optimization. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 584–589, 2000.

R. Gaizauskas and Y. Wilks. Information extraction: Beyond document retrieval. *Journal of documentation*, 54(1):70–105, 1998.

C. Gardent, H. Manuelian, and E. Kow. Which bridges for bridging definite descriptions. In *Proceedings of 4th International Workshop on Linguistically Interpreted Corpora (LINC-03) at EACL 2003*, pages 69–76. Association for Computational Linguistics, 2003.

S. Ghosh, G. Riccardi, and R. Johansson. Global features for shallow discourse parsing. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 150–159. Association for Computational Linguistics, 2012.

J. Grimes. *The thread of discourse*. The Hague: Mouton, 1975.

R. Grishman. Information extraction: Capabilities and challenges. *Notes prepared for the 2012 International Winter School in Language and Speech Technologies*, 2012.

R. Grishman and B. Sundheim. Message Understanding Conference 6: A brief history. In *The 16th International Conference on Computational Linguistics (COLING 1996)*, pages 466–471, 1996.

R. Grishman, J. Sterling, and C. Macleod. New York University: description of the PROTEUS system as used for MUC-3. In *Proceedings of the 3rd Message Understanding Conference*, pages 183–190. Association for Computational Linguistics, 1991.

Grolier. Academic American Encyclopedia. Grolier Electronic, 1990.

B. Grosz, S. Weinstein, and A. Joshi. Centering: A framework for modeling the local coherence of discourse. *Computational linguistics*, 21(2):203–225, 1995.

M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

M. A. K. Halliday and R. Hasan. *Cohesion in English*. Longman Publishing Group, 1976.

S. Harabagiu, R. Bunescu, and S. Maiorano. Text and knowledge mining for coreference resolution. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL-2001)*, pages 55–62. Association for Computational Linguistics, 2001.

D. Harman and M. Liberman. TIPSTER complete. *Corpus number LDC93T3A, Linguistic Data Consortium, Philadelphia*, 1993.

T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 415–422. Association for Computational Linguistics, 2004.

J. Hawkins. *Definiteness and Indefiniteness: A Study in Reference and Grammaticality Prediction*. Croom Helm Linguistics Series. Croom Helm, 1978.

M. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of 14th International Conference on Computational Linguistics (COLING 1992)*, pages 539–545, 1992.

M. Hearst. Automated discovery of WordNet relations. In C. Fellbaum, editor, *WordNet: an electronic lexical database*, pages 131–151. MIT Press, Cambridge, MA, 1998.

I. Hendrickx, G. Bouma, F. Coppens, W. Daelemans, V. Hoste, G. Kloosterman, A. Mineur, J. Van Der Vloet, and J. Verschelde. A coreference corpus and resolution system for dutch. *Proceedings of the Sixth International Language Resources and Evaluation (LREC08), Marrakech*, pages 28–30, 2008.

I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, 2010.

H. Hernault, H. Prendinger, M. Ishizuka, et al. HILDA: A discourse parser using Support Vector Machine classification. *Dialogue & Discourse*, 1(3), 2010.

J. Hirschberg and D. Litman. Empirical studies on the disambiguation of cue phrases. *Computational linguistics*, 19(3):501–530, 1994.

L. Hirschman. Comparing MUCK-II and MUC-3: Assessing the difficulty of different tasks. In *Proceedings of the 3rd Message Understanding Conference*, pages 25–30. Association for Computational Linguistics, 1991.

J. R. Hobbs. Resolving pronoun references. *Lingua*, 44(4):311–338, 1978.

J. R. Hobbs. *On the coherence and structure of discourse*. Center for the Study of Language and Information, Stanford, California, 1985.

J. R. Hobbs. The generic information extraction system. In *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pages 87–91. Association for Computational Linguistics, 1993.

J. R. Hobbs. Coherence and coreference. *Cognitive Science*, 3(1):67–90, 1979.

Y. Hong, X. Zhou, T. Che, J. Yao, Q. Zhu, and G. Zhou. Cross-argument inference for implicit discourse relation recognition. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 295–304. Association for Computing Machinery, 2012.

Y. Hou, K. Markert, and M. Strube. Global inference for bridging anaphora resolution. In *Proceedings of NAACL-HLT*, pages 907–917, 2013.

S. Huffman. Learning information extraction patterns from examples. In S. Wermter, E. Riloff, and G. Scheller, editors, *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing*, pages 246–260. Springer: Berlin, 1996.

B. Hutchinson. Mining the web for discourse markers. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 407–410, 2004a.

B. Hutchinson. Acquiring the meaning of discourse markers. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics*, pages 684–691. Association for Computational Linguistics, July 2004b.

B. Hutchinson. Modelling the substitutability of discourse connectives. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 149–156. Association for Computational Linguistics, 2005a.

B. Hutchinson. Modelling the similarity of discourse connectives. In *Proceedings of the the 27th Annual Meeting of the Cognitive Science Society (CogSci2005)*, 2005b.

M. Iris. Problems of the part-whole relation. In *Relational models of the lexicon*, pages 261–288. Cambridge University Press, 1989.

H. Ji and R. Grishman. Refining event extraction through cross-document inference. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pages 254–262. Association for Computational Linguistics, 2008.

H. Ji and D. Lin. Gender and animacy knowledge discovery from web-scale N-grams for unsupervised person mention detection. In *23rd Pacic Asia Conference on Language, Information and Computation*, 2009.

J. Jiang and C. Zhai. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics, 2007.

T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. J. Burges, and A. J. Smola, editors, *Advances in Kernel Methods Support Vector Learning*, pages 169–184. MIT Press, 1999.

R. Johansson and P. Nugues. Extended constituent-to-dependency conversion for English. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA 2007)*, pages 105–112, 2007.

I. Jolliffe. *Principal Component Analysis*. Springer, New York, 2 edition, 2002.

D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics.* Prentice-Hall, 2nd edition, 2009.

N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *The Companion Volume to the Proceedings*

of 42st Annual Meeting of the Association for Computational Linguistics, pages 178–181. Association for Computational Linguistics, 2004.

A. Kehler. Probabilistic coreference in information extraction. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, page 163. Association for Computational Linguistics, 1997.

J.-T. Kim and I. Moldovan. Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):713, 1995.

A. Knott and R. Dale. Using linguistic phenomena to motivate a set of coherence relations. *Discourse Processes*, 18:35–35, 1994.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(12):273 – 324, 1997. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(97)00043-X. URL `http://www.sciencedirect.com/science/article/pii/S000437029700043X`. ¡ce:title¿Relevance¡/ce:title¿.

S. Kok and P. Domingos. Extracting semantic networks from text via relational clustering. *Machine Learning and Knowledge Discovery in Databases*, pages 624–639, 2008.

J. Kolář. A comparison of language models for dialog act segmentation of meeting transcripts. In *Proceedings of Text, Speech and Dialogue: 11th International Conference*, volume 5246, pages 117–124. Springer, 2008.

Z. Kozareva, E. Riloff, and E. Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1048–1056. Association for Computational Linguistics, 2008.

G. Krupka, P. Jacobs, L. Rau, and L. Iwańska. GE: Description of the NLToolset System as Used for MUC-3. In *Proceedings of the 3rd Message Understanding Conference*, pages 144–149. Association for Computational Linguistics, 1991.

M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of the 14th International Conference on Machine Learning (ICML 1997)*, pages 179–186, 1997.

S. Kurohashi and M. Nagao. Automatic detection of discourse structure by checking surface information in sentences. In *Proceedings of the 15th Conference on Computational Linguistics*, pages 1123–1127. Association for Computational Linguistics, 1994.

M. Lapata and A. Lascarides. Inferring sentence-internal temporal relations. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 153–160. Association for Computational Linguistics, 2004.

S. Lappin and H. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.

A. Lascarides and N. Asher. Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and philosophy*, 16(5):437–493, 1993.

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

LDC. *ACE (Automatic Content Extraction) English Annotation Guidelines for Events, version 5.4.3 2005.07.01*, 2005a.

LDC. *ACE (Automatic Content Extraction) English Annotation Guidelines for Relations, version 5.8.3 2005.07.01*, 2005b.

H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proceedings of the CoNLL-2011 Shared Task*, pages 28–34, 2011.

W. Lehnert. Symbolic/subsymbolic sentence analysis: Exploiting the best of two worlds. *High-level connectionist models*, 1:135, 1991. Ablex Publishing Corporation.

W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. University of Massachusetts: Description of the CIRCUS System as Used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 282–288. Association for Computational Linguistics, 1992.

W. Lehnert, J. McCarthy, S. Soderland, E. Riloff, C. Cardie, J. Peterson, F. Feng, C. Dolan, and S. Goldman. UMass/Hughes: description of the CIRCUS system used for MUC-5. In *Proceedings of the 5th Message Understanding Conference*, pages 277–291. Association for Computational Linguistics, 1993.

V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

B. Levin. *English verb classes and alternations: A preliminary investigation*. University of Chicago press, 1993.

S. Liao and R. Grishman. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797. Association for Computational Linguistics, 2010.

S. Liao and R. Grishman. Acquiring topic features to improve event extraction: in preselected and balanced collections. In *Proceedings of the Conference on Recent Advances in Natural Language Processing*, pages 9–16. RANLP 2011 Organising Committee, 2011.

Z. Lin, M. Kan, and H. Ng. Recognizing implicit discourse relations in the Penn Discourse Treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 343–351. Association for Computational Linguistics, 2009.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002.

R. Longacre. *An anatomy of speech notions*. Peter de Ridder Press, 1976.

A. Louis, A. Joshi, and A. Nenkova. Discourse indicators for content selection in summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 147–156. Association for Computational Linguistics, 2010a.

A. Louis, A. Joshi, R. Prasad, and A. Nenkova. Using entity features to classify implicit discourse relations. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 59–62. Association for Computational Linguistics, 2010b.

S. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *The Journal of Machine Learning Research*, 8:935–983, 2007.

I. Mani, M. Verhagen, B. Wellner, C. Lee, and J. Pustejovsky. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 753–760. Association for Computational Linguistics, 2006.

W. Mann and S. Thompson. Relational Propositions in Discourse. *Discourse Processes*, 9(1):57–90, 1986.

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

D. Marcu. The rhetorical parsing of natural language texts. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–103. Association for Computational Linguistics, 1997.

D. Marcu. To build text summaries of high quality, nuclearity is not sufficient. In *Working Notes of the AAAI-98 Spring Symposium on Intelligent Text Summarization*, pages 1–8, 1998.

D. Marcu. A decision-based approach to rhetorical parsing. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 365–372. Association for Computational Linguistics, 1999.

D. Marcu and A. Echihabi. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 368–375. Association for Computational Linguistics, 2001.

D. Marcu, L. Carlson, and M. Watanabe. The automatic translation of discourse structures. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, NAACL 2000, pages 9–17, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=974305.974307`.

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

K. Markert and M. Nissim. Comparing knowledge sources for nominal anaphora resolution. *Computational Linguistics*, 31(3):367–402, 2005.

K. Markert, M. Strube, and U. Hahn. Inferential realization constraints on functional anaphora in the centering model. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society (CogSci 1996)*, pages 609–614, 1996.

K. Markert, N. Modjeska, and M. Nissim. Using the web for nominal anaphora resolution. In *Proceedings of the EACL 2003 Workshop on the Computational Treatment of Anaphora*, pages 39–46, 2003.

K. Markert, Y. Hou, and M. Strube. Collective classification for fine-grained information status. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 8–14. Association for Computational Linguistics, 2012.

M. Maslennikov and T.-S. Chua. A multi-resolution framework for information extraction from free text. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 592–599, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

J. McCarthy and W. Lehnert. Using decision trees for conference resolution. In *Proceedings of the 14th International Joint Conference on Artificial intelligence*, pages 1050–1055. Morgan Kaufmann Publishers Inc., 1995.

A. McKinlay and K. Markert. Modelling entity instantiations. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 268–274. RANLP 2011 Organising Committee, 2011.

A. McKinlay and K. Markert. Recognising sets and their elements: Tree kernels for entity instantiation identification. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, 2013.

Q. McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. The NomBank Project: An Interim Report. In *Proceedings of HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31. Association for Computational Linguistics, 2004.

D. Miller, R. Schwartz, R. Weischedel, and R. Stone. Named entity extraction from broadcast news. In *Proceedings of the DARPA Broadcast News Workshop (HUB-4)*, pages 37–40, 1999.

S. Miller, M. Crystal, H. Fox, L. Ramshaw, R. Schwartz, R. Stone, and R. Weischedel. Algorithms that learn to extract information BBN: Description of the SIFT system as used for MUC-7. In *Proceedings of the 7th Message Understanding Conference*. Association for Computational Linguistics, 1998.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual*

Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing, pages 1003–1011. Association for Computational Linguistics, 2009.

T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

R. Mitkov. Anaphora resolution: The state of the art. *Unpublished Manuscript*, 1999.

N. Modjeska. Towards a resolution of comparative anaphora: A corpus study of "other". In *Proceedings of PAPACOL*, 2000.

N. Modjeska. *Resolving other-anaphora*. PhD thesis, University of Edinburgh, 2004.

C. Montgomery, B. Stalls, R. Stumberger, N. Li, R. Belvin, A. Arnaiz, and S. Hirsh. Language Systems, Inc.: description of the DBG system as used for MUC-4. In *Proceedings of the 4th Message Understanding Conference*, pages 197–206. Association for Computational Linguistics, 1992.

A. Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In *Proceedings of the 17th European Conference on Machine Learning*, pages 318–329, 2006a.

A. Moschitti. Making tree kernels practical for natural language learning. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics, 2006b.

MUC. The NIST MUC website, 1987-1998. `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/`.

V. Nastase and M. Strube. Transforming Wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, 194(0):62–85, 2013.

A. Nedoluzhko, J. Mírovskỳ, and P. Pajas. The coding scheme for annotating extended nominal coreference and bridging anaphora in the Prague Dependency Treebank. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 108–111. Association for Computational Linguistics, 2009.

V. Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411. Association for Computational Linguistics, 2010.

M. Nissim. Learning information status of discourse entities. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 94–102. Association for Computational Linguistics, 2006.

M. Nissim, S. Dingare, J. Carletta, and M. Steedman. An annotation scheme for information status in dialogue. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2004)*, 2004.

OED Online. "op-ed, n. and adj.". Oxford University Press, URL http://www.oed.com/view/Entry/131693?redirectedFrom=op-ed&, 2013.

K. Ono, K. Sumita, and S. Miike. Abstract generation based on rhetorical structure extraction. In *Proceedings of the 15th conference on Computational linguistics-Volume 1*, pages 344–348. Association for Computational Linguistics, 1994.

U. Oza, R. Prasad, S. Kolachina, D. Sharma, and A. Joshi. The Hindi discourse relation bank. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 158–161. Association for Computational Linguistics, 2009.

M. Paşca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Names and similarities on the web: Fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 809–816. Association for Computational Linguistics, 2006.

M. Palmer, D. Gildea, and P. Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106, 2005.

P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113–120. Association for Computational Linguistics, 2006.

J. Park and C. Cardie. Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 108–112. Association for Computational Linguistics, 2012.

S. Patwardhan and E. Riloff. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 151–160. Association for Computational Linguistics, 2009.

E. Pitler and A. Nenkova. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 186–195. Association for Computational Linguistics, 2008.

E. Pitler, M. Raghupathy, H. Mehta, A. Nenkova, A. Lee, and A. Joshi. Easily identifiable discourse relations. In *Coling 2008: Companion volume: Posters*, pages 87–90. Coling 2008 Organizing Committee, 2008.

E. Pitler, A. Louis, and A. Nenkova. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 683–691. Association for Computational Linguistics, 2009.

M. Poesio. Associative descriptions and salience: A preliminary investigation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003) Workshop on The Computational Treatment of Anaphora*, pages 31–38, 2003.

M. Poesio and R. Vieira. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216, 1998.

M. Poesio, R. Vieira, and S. Teufel. Resolving bridging references in unrestricted text. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 1–6. Association for Computational Linguistics, 1997.

M. Poesio, R. Mehta, A. Maroudas, and J. Hitzeman. Learning to resolve bridging references. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04)*, pages 143–150. Association for Computational Linguistics, 2004.

S. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 192–199. Association for Computational Linguistics, 2006.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. European Language Resources Association (ELRA), 2008.

R. Prasad, A. Joshi, and B. Webber. Exploiting scope for shallow discourse parsing. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), 2010.

E. Prince. Toward a Taxonomy of Given-New Information. In P. Cole, editor, *Radical Pragmatics*, pages 223–255. New York: Academic Press, 1981.

E. Prince. The ZPG letter: subjects, definiteness, and information-status. In W. Mann and S. Thompson, editors, *Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text*, pages 295–326. John Benjamins Publishing Company, 1992.

J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. The TIMEBANK corpus. In *Proceedings of the Corpus Linguistics 2003 conference*, pages 647–656, 2003.

K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501. Association for Computational Linguistics, 2010.

A. Rahman and V. Ng. Learning the information status of noun phrases in spoken dialogues. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1069–1080. Association for Computational Linguistics, 2011.

A. Rahman and V. Ng. Learning the fine-grained information status of discourse entities. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 798–807. Association for Computational Linguistics, 2012.

L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics, 2009.

M. Recasens, E. Hovy, and M. A. Martı. A typology of near-identity relations for coreference (NIDENT). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), 2010.

N. Reiter and A. Frank. Identifying generic noun phrases. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 40–49. Association for Computational Linguistics, 2010.

A. Riester, D. Lorenz, and N. Seemann. A recursive annotation scheme for referential information status. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC '10)*, pages 717–722, 2010.

E. Riloff. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the National Conference on Artificial Intelligence*, pages 811–811. John Wiley & Sons LTD, 1993.

E. Riloff and J. Shepherd. A corpus-based approach for building semantic lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124, 1997.

B. Roark and E. Charniak. Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. In *Proceedings of the 17th International Conference on Computational linguistics*, pages 1110–1116. Association for Computational Linguistics, 1998.

B. Rosenfeld and R. Feldman. Clustering for unsupervised relation identification. In *Proceedings of the sixteenth ACM conference on Conference on Information and Knowledge Management*, pages 411–418. Association for Computing Machinery, 2007.

D. Roth and W. Yih. Probabilistic reasoning for entity & relation recognition. In *Proceedings of the 19th International Conference on Computational linguistics*, pages 1–7. Association for Computational Linguistics, 2002.

T. Sanders, W. Spooren, and L. Noordman. Toward a taxonomy of coherence relations. *Discourse Processes*, 15(1):1–35, 1992.

S. Sarawagi. Information extraction. *FnT Databases*, 1(3):261–377, 2007.

L. Sarmento, V. Jijkuon, M. de Rijke, and E. Oliveira. More like these: growing entity classes from seeds. In *Proceedings of the sixteenth ACM conference on Conference on Information and Knowledge Management*, pages 959–962. Association for Computing Machinery, 2007.

R. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine learning*, 39(2):135–168, 2000.

S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. CRYSTAL: inducing a conceptual dictionary. In *Proceedings of the 14th International Joint Conference on Artificial intelligence*, pages 1314–1319. Morgan Kaufmann Publishers Inc., 1995.

S. Somasundaran, G. Namata, L. Getoor, and J. Wiebe. Opinion graphs for polarity and discourse classification. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-4)*, pages 66–74. Association for Computational Linguistics, 2009.

W. Soon, H. Ng, and D. Lim. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544, 2001.

R. Soricut and D. Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (NAACL '03)*, pages 149–156. Association for Computational Linguistics, 2003.

C. Sporleder. Manually vs. automatically labelled data in discourse relation classification: Effects of example and feature selection. *LDV Forum*, 22(1):1–20, 2007.

C. Sporleder and M. Lapata. Discourse chunking and its application to sentence compression. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 257–264. Association for Computational Linguistics, 2005.

C. Sporleder and M. Lapata. Broad coverage paragraph segmentation across languages and domains. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(2): 1–35, 2006.

C. Sporleder and A. Lascarides. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14(3):369–416, 2008.

A. Sun, R. Grishman, and S. Sekine. Semi-supervised relation extraction with large-scale word clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 521–529. Association for Computational Linguistics, 2011.

K. Swampillai and M. Stevenson. Extracting relations within and across sentences. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 25–32. RANLP 2011 Organising Committee, 2011.

K. Swampillai and M. Stevenson. Inter-sentential relations in information extraction corpora. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA), 2010.

The PDTB Research Group. The PDTB 2.0. Annotation Manual. Technical Report IRCS-08-01, 2008.

V. R. Uzêda, T. Pardo, and M. Nunes. Evaluation of automatic text summarization methods based on rhetorical structure theory. In *Intelligent Systems Design and Applications, 2008. ISDA'08. Eighth International Conference on*, volume 2, pages 389–394. IEEE, 2008.

D. Vadas and J. Curran. Adding noun phrase structure to the penn treebank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 240–247. Association for Computational Linguistics, 2007.

K. van Deemter. Towards a generalization of anaphora. *Journal of Semantics*, 9(1):27–51, 1992.

K. van Deemter and R. Kibble. On coreferring: Coreference in MUC and related annotation schemes. *Computational linguistics*, 26(4):629–637, 2000.

T. Van Dijk. Pragmatic connectives. *Journal of Pragmatics*, 3(5):447–57, 1979.

Y. Versley. Multilabel tagging of discourse relations in ambiguous temporal connectives. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 154–161. RANLP 2011 Organising Committee, 2011.

R. Vieira and M. Poesio. An empirically based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593, 2000.

S. Vishwanathan and A. J. Smola. Fast kernels for string and tree matching. *Advances in neural information processing systems*, 15:569–576, 2002.

R. C. Wang and W. W. Cohen. Language-independent set expansion of named entities using the web. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 342–350. IEEE Computer Society, 2007.

W. Wang, J. Su, and C. L. Tan. Kernel based discourse relation recognition with temporal ordering information. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 710–719. Association for Computational Linguistics, 2010.

B. Webber. Genre distinctions for discourse in the Penn TreeBank. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint*

Conference on Natural Language Processing of the AFNLP, pages 674–682. Association for Computational Linguistics, 2009.

B. Webber, A. Knott, M. Stone, and A. Joshi. Discourse relations: A structural and presuppositional account using lexicalised TAG. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 41–48. Association for Computational Linguistics, 1999.

B. Webber, M. Stone, A. Joshi, and A. Knott. Anaphora and discourse structure. *Computational Linguistics*, 29(4):545–587, 2003.

R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, M. Franchini, M. El-Bachouti, R. Belvin, and A. Houston. Ontonotes release 4.0. Linguistic Data Consortium, Philadelphia, 2011. LDC Catalog Number LDC2011T03.

B. Wellner and J. Pustejovsky. Automatically identifying the arguments of discourse connectives. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 92–101. Association for Computational Linguistics, 2007.

T. Wilson, J. Wiebe, and R. Hwa. Just how mad are you? Finding strong and weak opinion clauses. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-2004)*, pages 761–766, 2004.

T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics*, 35(3):399–433, 2009.

M. Winston, R. Chaffin, and D. Herrmann. A taxonomy of part-whole relations. *Cognitive science*, 11(4):417–444, 1987.

F. Wolf and E. Gibson. Paragraph-, word-, and coherence-based approaches to sentence ranking: A comparison of algorithm and human performance. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 383. Association for Computational Linguistics, 2004.

F. Wolf and E. Gibson. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2):249–287, 2005.

F. Wolf, E. Gibson, A. Fisher, and M. Knight. A procedure for collecting a database of texts annotated with coherence relations. *Documentation accompanying the Discourse GraphBank, LDC2005T08*, 2003.

D. Yang and D. M. Powers. Verb similarity on the taxonomy of wordnet. In *Proceedings of The Third International WordNet Conference: GWC-06*, pages 121–128, 2006.

J. Yang and V. Honavar. Feature subset selection using a genetic algorithm. In *Feature extraction, construction and selection*, pages 117–136. Springer, 1998.

X. Yang and J. Su. Coreference resolution using semantic relatedness information from automatically discovered patterns. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 528–535. Association for Computational Linguistics, 2007.

X. Yang, G. Zhou, J. Su, and C. Tan. Coreference resolution using competition learning approach. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 176–183. Association for Computational Linguistics, 2003.

Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.

L. Yao, A. Haghighi, S. Riedel, and A. McCallum. Structured relation discovery using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1456–1466. Association for Computational Linguistics, 2011.

A. Yates and O. Etzioni. Unsupervised resolution of objects and relations on the web. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 121–130. Association for Computational Linguistics, 2007.

G. P. Zarri. Automatic representation of the semantic relationships corresponding to a french surface expression. In *Proceedings of the First Conference on Applied Natural Language Processing*, pages 143–147. Association for Computational Linguistics, 1983.

D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106, 2003.

D. Zeyrek and B. Webber. A discourse resource for Turkish: Annotating discourse connectives in the METU corpus. In *Proceedings of The 6th Workshop on Asian Language Resources (ALR 6)*, pages 65–72. Association for Computational Linguistics, 2008.

M. Zhang, J. Su, D. Wang, G. Zhou, and C. Tan. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, pages 378–389. Association for Computational Linguistics, 2005.

M. Zhang, J. Zhang, J. Su, and G. Zhou. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 825–832. Association for Computational Linguistics, 2006.

S. Zhao and R. Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 419–426. Association for Computational Linguistics, 2005.

G. Zhou, J. Su, J. Zhang, and M. Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 427–434. Association for Computational Linguistics, 2005.

G. Zhou, M. Zhang, D. Ji, and Q. Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 728–736. Association for Computational Linguistics, 2007.

Z.-M. Zhou, Y. Xu, Z.-Y. Niu, M. Lan, J. Su, and C. L. Tan. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10): Posters*, pages 1507–1514. Association for Computational Linguistics, 2010.

X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

Appendix A

Inter-sentential Annotation Scheme

We annotate full Wall Street Journal texts for the presence of *Entity Instantiations*. *Entity Instantiations* (EIs) are set membership and subset relationships between noun phrases.

Example A.1 shows an example of a set membership instantiation. The plural noun phrase (NP) *'Some European funds'* describes a set of investment funds, of which the singular NP *'Spain Fund'* is a member.

Example A.2 shows an example of a subset instantiation. In this case, the plural NP *'Bids totalling $515 million'* describes a set of bids for a company, and the plural NP *'Accepted offers'* refers to a subset of the bids which have been accepted.

(A.1) a. **Some European funds** recently have skyrocketed.

b. *Spain Fund* has surged to a startling 120% premium.

(A.2) a. **Bids totalling $515 million** were submitted.

b. *Accepted offers* ranged from 8.38% to 8.395%

Instantiations can be marked based on knowledge present in the sentence being annotated, knowledge present elsewhere in the document, from widely known facts, or knowledge gained by researching the entities involved. Example A.3 shows an instantiation which may be marked on the basis of knowledge in the sentence. Example A.4 shows an instantiation which requires common world knowledge. Example A.5 shows an instantiation which may require one to research the members of the organisations OPEC and EU to establish if an instantiation is present.

(A.3) a. Gandhi won **eight Oscars** in 1983.

b. This included *the Oscar for best picture*

(A.4) a. **Drinks companies** and chocolate producers have struggled this quarter.

b. *Pepsi-Co* issued a profit warning last week.

(A.5) a. **OPEC members and EU members** have come together at this week's summit.

b. *Venezuela* has served as host.

Additionally, correct interpretation of an Entity Instantiation often needs contextual knowledge. In Examples A.6 and A.7, the contextual information about the attitudes of the workers is necessary to determine whether an Entity Instantiation exists.

(A.6) a. **Some workers** are opposed to strike action.

b. *John Smith* fears that a strike could damage the industry's public perception.

(A.7) a. **Some workers** are opposed to strike action.

b. *David Jones*, however, is willing to put his job on the line for the cause. *(Not an instantiation.)*

A test to establish the presence of an Entity Instantiation

A simple 'rule-of-thumb' can be used to to establish whether an instantiation is present:

> All statements that were made about the set must also apply to the set member or subset.

This rule excludes cases where the the statements could not apply to an individual member of the set, but instead describe the nature of the set, such as *'is large'* or *'contains five members'*. An easy way to check whether the rule applies is to rephrase the instantiation in the following format:

> {Set Member/Subset} is a {Set} that {statements made about the set}

Following this rule, in Example A.1 we can say

> *'Spain Fund is a European fund that has recently skyrocketed'*.

Similarly in Example A.4 one may say

> *'Pepsi-Co is a drinks company that has struggled this quarter.'*

In the case of Example A.7, which is not an instantiation, we cannot say that

'David Jones is one of the workers who is opposed to strike action'.

An exception to this rule occurs when the set is comprised of a conjunction. For example, one cannot say in Example A.5 that

'Venezuela is a OPEC and EU member that has attended this week's summit.'

This is because we view a set comprising a conjunction as a union of its members. In these circumstances, we can replace the *and* with *or* and the rule now makes sense:

'Venezuela is a OPEC or EU member that has attended this week's summit.'

Simplifications made

In order to make the annotation easier, a number of simplifications of the problem have been made. They are detailed below.

What constitutes a set

The annotation tool automatically extracts NPs and divides them into singular NPs and plural NPs, based on the Penn Treebank part-of-speech tags associated with the words in the NP. Only plural NPs can function as sets and subsets. Only singular NPs can function as set members. These restrictions mean some potential instantiations, such as that in Example A.8 will not be marked.

(A.8) a. **The group** had recently been formed.

b. *John Smith* was head of it.

Adjacent intersentential annotation

Our largest simplification, which considerably restricts the possible set members and subsets that can be marked as participating in a relationship with a set, is limiting our annotation to *between*[1] adjacent sentences.

Without this restriction, any plural NP in a document could participate in an instantiation with any other NP in the entire document, which would be difficult to annotate in documents longer than 3 or 4 paragraphs.

We allow for a pair of sentences to be annotated in both directions; potential sets in the first sentence and the potential set members or subsets in the second, and potential sets in the second sentence and potential set members and subsets in the first sentence.

[1] Our annotation is strictly between sentences. No intrasentential annotation is carried out.

When not to mark an instantiation

Generic mentions

In Example A.9, *'the planner'* mentioned in the second sentence refers to a notional planner, rather than any actual member of the set of **Planners**, and therefore should not be marked as an instantiation. Examples A.10 and A.11 show similar examples where the instantiation should not be marked.

(A.9) a. **Planners** often have to make difficult decisions.

b. The issue: does *the planner* have the required qualifications to make them.

(A.10) a. **John Smith and John Doe** are competing for the contract.

b. *Either* could clinch it.

(A.11) a. **Both companies** are struggling.

b. *Either* might go bust before the year is out.

Members implicitly excluded from sets

Occasionally the context of the candidate instantiation can implicitly exclude a member or subset from participating. In Example A.12, the set of **Democrats** excludes *Senator Smith*, as he is certainly going to vote for the measure.

(A.12) a. **Democrats** are reluctant to break ranks and vote against the measure.

b. *Senator Smith*, their leader in the senate has staked his reputation on the bill, and those voting against would be betraying his confidence.

Metonymic mentions

If a candidate set member or subset is a metonymic reference, then an instantiation should only be marked if the set is of the concept the metonymy represents rather than the literal reading of the word. This often occurs in the WSJ corpus with regards to shares in a company being referred to by the name of the company itself.

Example A.13 shows a situation in which an instantiation should not be marked — *'Hollywood'* is referring to the industry rather than the district of L.A. Conversely, an instantiation should be marked in Example A.14, as *'Westminster'* refers to the UK Parliament, rather than the area in this context.

(A.13) a. *Hollywood* has made countless films about L.A.

b. **All of the districts in the city** have starred at some point.

(A.14) a. **Parliaments around the EU** were ratifying the treaty this week.

b. *Westminster* passed it on Tuesday.

Negated Mentions

If a set, set member or subset is negated, such as those in Examples A.15 and A.16, it cannot participate in an instantiation.

(A.15) a. **Neither the US nor the UK** have managed to keep their debt under control.

b. *The UK*'s debt has risen by 10% this year alone.

(A.16) a. *John*, Mary and James just sat and watched.

b. **Not one of them** dared intervene.

'Not A Mention' Definition

The annotation tool provides an option to mark potential sets, members and subsets as 'not a mention'. Non-mentions *cannot* participate in an instantiation.

There are two main reasons why a noun phrase may be marked as 'not a mention'; an error in the pre-processing/annotation which is used to identify the NPs and classify them as singular or plural, and cases where it is impossible for the tool to tell whether the NP is a mention or not.

Idiomatic Mentions

Idiomatic NPs that have no literal meaning should be marked as 'not a mention'. Examples A.17, A.18 and A.19 show examples of idiomatic NPs which are not mentions. Example A.20 is a mention — the MP's eyes exist.

(A.17) How many senators does it take to change *a light bulb*?

(A.18) The chairman has *an axe* to grind with the regulators.

(A.19) On *the ropes*.

(A.20) The MP, known for his *eagle eyes*, spotted the error immediately.

Occasionally, metaphoric mentions can occur as part of a recurring theme, with the potential for instantiations to occur, such as in Example A.21. These should not be marked 'not a mention'.

(A.21) a. Bob Dylan asked '**How many roads** must a man walk down?'

b. Well, *one road* is particularly well walked.

Generic Pronouns

Generic uses of *'we'* and *'you'* should be marked as 'not a mention', as should references to the reader or audience of a text.

(A.22) *You* know, it's really tricky to figure out where to begin with this mess.

(A.23) Maybe he didn't start it, but Mohandas Gandhi certainly provided a recognizable beginning to non-violent civil disobedience as *we* know it today.

(A.24) So *dear reader*, we advise that you don't rush into your investments.

Non-referential 'it'

Non-referential uses of *'it'*, such as those in Examples A.25 and A.26 should also be marked not a mention.

(A.25) *It* seems that this weather is here to stay for the week.

(A.26) *It* is said that only fools rush in.

Pre-processing errors

As previously mentioned, it is necessary to use the 'not a mention' label to mark NPs which are either misclassified in terms of singular/plural. This occasionally occurs with conjunctive NPs, as it can sometimes be difficult for the algorithm to establish the difference between a company name with an 'and' in it and a genuine conjunction (e.g. Example A.27).

As the pronoun 'you' can be either plural or singular, it is included as both. Use 'not a mention' to mark whichever it is **not** — i.e. mark the plural as a non-mention if it is singular, mark the singular as a non-mention if it is plural. If it is generic, as explained in Section A, mark both mentions as 'not a mention'.

(A.27) Marks and Spencer.

The other type of processing error which we encounter is down to (rare) parsing errors in the Penn Treebank, or (rare) errors in the NP extraction process. These can include: parts of proper noun phrases being included as potential sets or members (Example A.28), discourse connectives being included (Example A.29) and odd parsing of phrases (Example A.30). All of these types of errors should be marked as non-mentions.

(A.28) The House of **Lords**.

(A.29) In *fact*, this has happened.

(A.30) Senator Moore *R., Iowa*.

(A.31) Hardly a day passes without news photos of *the police dragging limp protesters from some building or thoroughfare in one of our cities*.

Appendix B

Intra-sentential Annotation Scheme

Problem Definition

We annotate full Wall Street Journal texts for the presence of *Entity Instantiations*. *Entity Instantiations* (EIs) are set membership and subset relationships between noun phrases.

Example B.1 shows an example of a set membership instantiation. The plural noun phrase (NP) *'Some European funds'* describes a set of investment funds, of which the singular NP *'Spain Fund'* is a member.

Example B.2 shows an example of a subset instantiation. In this case, the plural NP *'Bids totalling $515 million'* describes a set of bids for a company, and the plural NP *'Accepted offers'* refers to a subset of the bids which have been accepted.

(B.1) **Some European funds** have recently skyrocketed, such as *Spain Fund* which has surged to a startling 120% premium.

(B.2) **Bids totalling $515 million** were submitted, with *accepted offers* ranging from 8.38% to 8.395%

Often, intra-sentential instantiations can be part of a nested NP. Example B.3 shows a set member nested within the set from which it is drawn, and Example B.4 shows a set nested within its subset.

(B.3) So if anything happened to me, I'd want to leave behind enough so that my 33-year-old husband would be able to pay off ***the mortgage*** **and some other debts**.

(B.4) *Two of* **the men** disagreed with the judgement.

Instantiations can be marked based on knowledge present in the sentence being annotated, knowledge present elsewhere in the document, from widely known facts, or knowledge gained by researching the entities involved. Example B.5 shows an instantiation which may be marked on the basis of knowledge in the sentence. Example B.6 shows an instantiation which requires common world knowledge. Example B.7 shows an instantiation which may require one to research the members of the organisations OPEC and EU to establish if an instantiation is present.

(B.5) Gandhi won **eight Oscars** in 1983, including *the Oscar for best picture*

(B.6) **Drinks companies** and confectionery producers, such as *Pepsi-Co*, have struggled this quarter.

(B.7) **OPEC members and EU members** have come together at this week's summit, hosted by *Venezuela*.

Additionally, correct interpretation of an Entity Instantiation often needs contextual knowledge. In Examples B.8 and B.9, the contextual information about the attitudes of the workers is necessary to determine whether an Entity Instantiation exists.

(B.8) **Some of the workers** are opposed to strike action, including *John Smith* who fears that a strike could damage the industry's public perception.

(B.9) **Some workers** are opposed to strike action, but *David Jones* is willing to put his job on the line for the cause. *(Not an instantiation.)*

A test to establish the presence of an Entity Instantiation

A simple 'rule-of-thumb' can be used to to establish whether an instantiation is present:

> All statements that were made about the set must also apply to the set member or subset.

This rule excludes cases where the the statements could not apply to an individual member of the set, but instead describe the nature of the set, such as *'is large'* or *'contains five members'*. An easy way to check whether the rule applies is to rephrase the instantiation in the following format:

> {Set Member/Subset} is a {Set} that {statements made about the set}

Following this rule, in Example B.1 we can say

'Spain Fund is a European fund that has recently skyrocketed'.

Similarly in Example B.6 one may say

'Pepsi-Co is a drinks company that has struggled this quarter.'

In the case of Example B.9, which is not an instantiation, we cannot say that

'David Jones is one of the workers who is opposed to strike action'.

An exception to this rule occurs when the set is comprised of a conjunction. For example, one cannot say in Example B.7 that

'Venezuela is a OPEC and EU member that has attended this week's summit.'

This is because we view a set comprising a conjunction as a union of its members. In these circumstances, we can replace the *and* with *or* and the rule now makes sense:

'Venezuela is a OPEC or EU member that has attended this week's summit.'

Simplifications made

In order to make the annotation easier, a number of simplifications of the problem have been made. They are detailed below.

What constitutes a set

The annotation tool automatically extracts NPs and divides them into singular NPs and plural NPs, based on the Penn Treebank part-of-speech tags associated with the words in the NP. Only plural NPs can function as sets and subsets. Only singular NPs can function as set members. These restrictions mean some potential instantiations, such as that in Example B.10 will not be marked.

(B.10) **The group** had recently been formed, with *John Smith* as head of it.

Intra-sentential annotation

Our largest simplification, which considerably restricts the possible set members and subsets that can be marked as participating in a relationship with a set, is limiting our annotation to *within* single sentences.

Without this restriction, any plural NP in a document could participate in an instantiation with any other NP in the entire document, which would be difficult to annotate in documents longer than 3 or 4 paragraphs.

Dealing with nested instantiations

An additional complication of intrasentential instantiations is the possibility of nesting. We follow a number of rules for marking these instantiations.

Conjunctive Phrases

The annotation of conjunctions and list phrases is the most straightforward nesting situation to deal with — any direct child NP is an instantiation of the conjunctive NP. In Example B.12, the NP *'Leo Messi'*, part of an apposition, would not be included.

(B.11) **both** ***the Washington Post*** **and** ***the New York Times***

(B.12) ***Cristiano Ronaldo*** **and** ***Gonzalo Higuain, a compatriot of Leo Messi***

'Of' Phrases

Many nested phrases take the form *'X of Y'*. We follow a number of guidelines for these constructions.

Ensure that there is an instantiation. Firstly, we must ensure that there is a true set membership or subset relationship, by applying the test described above. For instance, in Example B.13, *West Berlin, Hesse, and North-Rhine Westphalia* is not a subset of *the states of West Berlin, Hesse, and North-Rhine Westphalia*, nor vice versa, and so there is no instantiation. Example B.14, however would be a subset instantiation.

(B.13) (the states of (West Berlin, Hesse, and North-Rhine Westphalia))[1]

(B.14) (states including (West Berlin, Hesse, and North-Rhine Westphalia))

Subsets should not be marked as set members. We only mark set membership instantiations between a singular NP and a plural NP, and only mark subset instantiations between a pair of plural NPs. In situations such as Examples B.15 and B.16, where due to the nature of the NP classification *a series of management proposals* is classified as *singular*, but in fact the NP represents a plural, a set membership instantiation should not be marked.

(B.15) (a series of (management proposals))

[1] Brackets are used to make clear the boundaries of the NPs in question.

(B.16) (a third of (Hong Kong consumers))

However, if the same sort of pattern occurs between two plural NPs, such as in Examples B.17, B.18 and B.19, it should be marked.

(B.17) (dozens of (US states))

(B.18) (the 23 pairs of (chromosomes) in the cells that contain the genes)

(B.19) (many of (the counties of England))

Do not mark instantiations which describe general proportions rather than sets. Examples B.20 and B.21, and similar constructions should not be marked as entity instantiations.

(B.20) (one of (every five women))

(B.21) (three of (every twelve men))

Dealing with vague and generic sets. With concrete, enumerable sets, such as those mentioned in Examples B.16 and B.17, the interpretation of the instantiation is reasonably straightforward. It can be harder to interpret with more vague sets, such as Examples B.22, B.23 and B.24. However, we still annotate instantiations from these sets, and all three examples are positive examples of instantiations.

(B.22) The red granite mausoleum draws (thousands of (visitors)) daily.

(B.23) He earns (millions of (dollars)) each year.

(B.24) Mr. Auvil, razor sharp at 83, has picked and packed (a zillion pecks of (apples)) over the past 65 years.

When not to mark an instantiation

Generic mentions

In Example B.25, *'the planner'* mentioned in the second sentence refers to a notional planner, rather than any actual member of the set of **Planners**, and therefore should not be marked as an instantiation. Examples B.26 and B.27 show similar examples where the instantiation should not be marked.

(B.25) a. **Planners** often have to make difficult decisions.

b. The issue: does *the planner* have the required qualifications to make them.

(B.26) a. **John Smith and John Doe** are competing for the contract.

b. *Either* could clinch it.

(B.27) a. **Both companies** are struggling.

b. *Either* might go bust before the year is out.

Members implicitly excluded from sets

Occasionally the context of the candidate instantiation can implicitly exclude a member or subset from participating. In Example B.28, the set of **Democrats** excludes *Senator Smith*, as he is certainly going to vote for the measure.

(B.28) a. **Democrats** are reluctant to break ranks and vote against the measure.

b. *Senator Smith*, their leader in the senate has staked his reputation on the bill, and those voting against would be betraying his confidence.

Metonymic mentions

If a candidate set member or subset is a metonymic reference, then an instantiation should only be marked if the set is of the concept the metonymy represents rather than the literal reading of the word. This often occurs in the WSJ corpus with regards to shares in a company being referred to by the name of the company itself.

Example B.29 shows a situation in which an instantiation should not be marked — *'Hollywood'* is referring to the industry rather than the district of L.A. Conversely, an instantiation should be marked in Example B.30, as *'Westminster'* refers to the UK Parliament, rather than the area in this context.

(B.29) a. *Hollywood* has made countless films about L.A.

b. **All of the districts in the city** have starred at some point.

(B.30) a. **Parliaments around the EU** were ratifying the treaty this week.

b. *Westminster* passed it on Tuesday.

Negated Mentions

If a set, set member or subset is negated, such as those in Examples B.31 and B.32, it cannot participate in an instantiation.

(B.31) **Neither the US nor the UK** have managed to keep their debt under control — *The UK*'s debt has risen by 10% this year alone.

(B.32)

(B.33) *John*, Mary and James just sat and watched, **not one of them** would dare intervene.

'Not A Mention' Definition

The annotation tool provides an option to mark potential sets, members and subsets as 'not a mention'. Non-mentions *cannot* participate in an instantiation.

There are two main reasons why a noun phrase may be marked as 'not a mention'; an error in the pre-processing/annotation which is used to identify the NPs and classify them as singular or plural, and cases where it is impossible for the tool to tell whether the NP is a mention or not.

Idiomatic Mentions

Idiomatic NPs that have no literal meaning should be marked as 'not a mention'. Examples B.34, B.35 and B.36 show examples of idiomatic NPs which are not mentions. Example B.37 is a mention — the MP's eyes exist.

(B.34) How many senators does it take to change *a light bulb*?

(B.35) The chairman has *an axe* to grind with the regulators.

(B.36) On *the ropes*.

(B.37) The MP, known for his *eagle eyes*, spotted the error immediately.

Occasionally, metaphoric mentions can occur as part of a recurring theme, with the potential for instantiations to occur, such as in Example B.38. These should *not* be marked 'not a mention'.

(B.38) a. Bob Dylan asked 'How many **roads** must a man walk down?'

b. Well, *one road* is particularly well walked.

Generic Pronouns

Generic uses of *'we'* and *'you'* should be marked as 'not a mention', as should references to the reader or audience of a text.

(B.39) *You* know, it's really tricky to figure out where to begin with this mess.

(B.40) Maybe he didn't start it, but Mohandas Gandhi certainly provided a recognizable beginning to non-violent civil disobedience as *we* know it today.

(B.41) So *dear reader*, we advise that you don't rush into your investments.

Non-referential 'it'

Non-referential uses of *'it'*, such as those in Examples B.42 and B.43 should also be marked not a mention.

(B.42) *It* seems that this weather is here to stay for the week.

(B.43) *It* is said that only fools rush in.

Pre-processing errors

As previously mentioned, it is necessary to use the 'not a mention' label to mark NPs which are either misclassified in terms of singular/plural. This occasionally occurs with conjunctive NPs, as it can sometimes be difficult for the algorithm to establish the difference between a company name with an 'and' in it and a genuine conjunction (e.g. Example B.44).

As the pronoun 'you' can be either plural or singular, it is included as both. Use 'not a mention' to mark whichever it is **not** — i.e. mark the plural as a non-mention if it is singular, mark the singular as a non-mention if it is plural. If it is generic, as explained in Section B, mark both mentions as 'not a mention'.

(B.44) Marks and Spencer.

The other type of processing error which we encounter is down to (rare) parsing errors in the Penn Treebank, or (rare) errors in the NP extraction process. These can include: parts of proper noun phrases being included as potential sets or members (Example B.45), discourse connectives being included (Example B.46) and odd parsing of phrases (Example B.47). All of these types of errors should be marked as non-mentions.

(B.45) The House of **Lords**.

(B.46) In *fact*, this has happened.

(B.47) Senator Moore *R., Iowa*.

(B.48) Hardly a day passes without news photos of *the police dragging limp protesters from some building or thoroughfare in one of our cities*.

Appendix C

Psuedocode of Singular/Plural NP Classifier

```
PROCEDURE singular_plural_classifier ( NP )

  plural pronouns = {'ourselves', "'em", 'ours', 'you', 'we',"'s",
                     'ya', 'them', 'they','themselves', 'theirs',
                     'us', 's',"y'all"}
  singular pronouns = {'his', 'it', 'yourself', 'itself', 'thyself',
                       'her', 'him', 'you', 'himself',  "'s", 'ya',
                       'mine', 'one','oneself','herself', 'he', 'me',
                       'myself', 'i','s', 'she', "'t",  "'t-","t'"}

  if NP is a named entity:
    return singular

  if NP is a conjunctive phrase:
    return plural

  headword = retrieve_head_word(NP)

  tag = part_of_speech_tag(headword)

  # Plural noun
  if tag == "NNS":
    return "plural"
```

```
# Singular noun or proper noun
else if tag == "NN" or tag == "NNP":
  return "singular"

# Determiner
else if tag == "DT":
  if headword in ["a","an","one","another","this","that"]:
    return "singular"
  else:
    return "plural"

# Plural proper noun
else if tag == "NNPS":
  if headword is a named entity:
    return "singular"
  else:
    return "plural"

# Cardinal number
else if tag == "CD":
  if len(headword) == 4 and headword is all digits:
    if int(headword) in range(1600,2050): # Likely to be a year
      return "singular"
    else:
      return "plural"
  else:
    if headword in ["1","one"]:
      return "singular"
    else if "." in headword: # Decimal number
      return "singular"
    else:
      return "plural"

# Personal Pronoun
else if tag.startswith("PRP"):
  if headword in singularpn and in pluralpn:
    return "both"
  else if headword in singularpn:
```

```
    return "singular"
  else if headword in pluralpn:
    return "plural"

# Dollar or pound value
else if tag == "$" or tag == "#":
  return "singular"

# Verbal gerund
else if tag == "VBG":
  return "singular"

# Superlative adjective, such as best, worst or oldest.
else if tag == "JJS":

  # e.g. the best, the oldest
  if headword has "the" as a dependency:
    return "singular"
  else:
    return "plural"

# Comparative adjective; better, worse or older.
else if tag == "JJR":
  for word in dependencynodelist:

  # e.g. the better, the older
  if headword has "the" as a dependency:
    return "singular"
  else:
    return "plural"

else if tag == "JJ":
  return "singular"

# Foreign word, hard to classify.
else if tag == "FW":
  return "singular"
```

```
# Pre-determiner
else if tag == "PDT":
  if headword in ["both","many","all","half"]:
    return "plural"
  else:
    return "exclude"

else if tag == "RB" or tag == "RBS":
  if headword in ["some","many","all","most"]:
    return "plural"
  else:
    return "singular"

# Mathematical letters (a,b,c,x,y,z) and so on
else if tag == "SYM":
  return "both"

# Comparative adverb
else if tag == "RBR":
  return "both"

# Interjection
else if tag == "UH":
  return "singular"

# "Neither" is the only CC that should appear as a headword
else if tag == "CC":
  return "plural"

# Possesive e.g. 's
else if tag == "POS":
  possesive_head = get_head_word(headword)
  return singular_plural_classifier(possesive_head)

# If these are heads, we don't require the NP.
else if tag in ["TO","MD","WRB","WP","RP","WDT","IN"]:
  return "exclude"
```

```
# Verb
else if tag.startswith("VB"):
  return "both"

# Possesive pronoun, can't tell.
else if tag == "PRP$":
  return "both" #
```

Appendix D

Hierarchy of relations in the PDTB