Social Network Extraction from Text

Apoorv Agarwal

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2016

ABSTRACT

Social Network Extraction from Text

Apoorv Agarwal

In the pre-digital age, when electronically stored information was non-existent, the only ways of creating representations of social networks were by hand through surveys, interviews, and observations. In this digital age of the internet, numerous indications of social interactions and associations are available electronically in an easy to access manner as structured meta-data. This lessens our dependence on manual surveys and interviews for creating and studying social networks. However, there are sources of networks that remain untouched simply because they are not associated with any meta-data. Primary examples of such sources include the vast amounts of literary texts, news articles, content of emails, and other forms of unstructured and semi-structured texts.

The main contribution of this thesis is the introduction of natural language processing and applied machine learning techniques for uncovering social networks in such sources of unstructured and semi-structured texts. Specifically, we propose three novel techniques for mining social networks from three types of texts: unstructured texts (such as literary texts), emails, and movie screenplays. For each of these types of texts, we demonstrate the utility of the extracted networks on three applications (one for each type of text).

Table of Contents

List of Figures. viii

List of Tables. xiii

I Introduction **1**

1 Introduction **2**

1.1 Motivation 2

1.2 A High Level Organization of the Thesis 5

1.3 Contributions in Terms of Techniques 8

1.4 Contributions in Terms of Applications 11

1.5 Thesis Organization 13

2 A New Kind of a Social Network **14**

2.1 Definition of Entities 15

2.2 Definition of Social Events 15

2.3 Subcategorization of Social Events 19

2.3.1 Examples for the Subcategories of OBS 20

2.3.2 Examples for the Subcategories of INR 21

2.4 Social Events Considered in this Thesis 22

2.5 Comparison of Social Events with ACE Relations and Events 23

2.5.1 Social Events versus ACE Relations 23

2.5.2 Social Events versus ACE Events 24

2.6 Comparison of Social Events with Conversations 32

2.7 Conclusion 35

II Extracting Social Networks from Unstructured Text **36**

3 Introduction **38**

3.1 Task Definition . . . 38

3.2 Related Work on Relation Extraction . . . 39

3.2.1 Bootstrapping . . . 40

3.2.2 Unsupervised . . . 42

3.2.3 Distant Supervision . . . 43

3.2.4 Feature Based Supervision . . . 44

3.2.5 Convolution Kernels Based Supervision . . . 46

3.2.6 Rule Based . . . 51

3.3 Conclusion . . . 52

4 Machine Learning Approach **55**

4.1 Data . . . 56

4.1.1 Annotation Procedure . . . 57

4.1.2 Additional Annotation Instructions . . . 60

4.1.3 Inter-annotator Agreement . . . 61

4.2 Baselines . . . 65

4.2.1 Co-occurrence Based (CoOccurN) . . . 65

4.2.2 Syntactic Rule Based (SynRule) . . . 66

4.2.3 Feature Based Supervision (GuoDong05) . . . 69

4.2.4 Convolution Kernel Based Supervision (Nguyen09) . . . 71

4.2.5 Features derived from FrameNet (BOF and SemRules) . . . 74

4.3 Structures We Introduce . . . 76

4.3.1 Sequence on Grammatical Relation Dependency Tree (SqGRW) . . . 76

4.3.2 Semantic trees (FrameForest, FrameTree, FrameTreeProp) . . . 76

4.4 Experiments and Results: Intrinsic Evaluation . . . 80

4.4.1 Task Definitions . . . 80

4.4.2 Data Distribution . . . 80

4.4.3 Experimental Set-up . . . 81

4.4.4 Experiments and Results for Social Event Detection 81
4.4.5 Experiments and Results for Social Event Classification 85
4.4.6 Experiments and Results for Directionality Classification 86
4.4.7 Experiments and Results for Social Network Extraction (SNE) 87
4.5 Conclusion and Future Work 88

5 Application: Validating Literary Theories **93**
5.1 Introduction 93
5.2 Literary Theories 95
5.3 Conversational, Interaction, and Observation Networks 95
5.4 Evaluation of SINNET 97
5.4.1 Gold standards: Conv-Gold and SocEv-Gold 98
5.4.2 Evaluation and Results 99
5.4.3 Discussion of Results 100
5.5 Expanded Set of Literary Hypotheses 100
5.6 Methodology for Validating Hypotheses 102
5.7 Results for Testing Literary Hypotheses 104
5.8 Conclusion and Future Work 106

III Extracting Networks from Emails **108**

6 Introduction **111**
6.1 Terminology 111
6.2 Task Definition 114
6.3 Literature Survey 114
6.4 Conclusion 116

7 Machine Learning Approach **117**
7.1 Data 117
7.1.1 A Brief History of the Enron Corporation 117
7.1.2 The Enron Email Corpus 120

7.2 Name Disambiguation Approach . . . 121
7.2.1 Terminology . . . 121
7.2.2 Candidate Set Generation Algorithm . . . 122
7.2.3 Our Name Disambiguation Algorithm . . . 125
7.3 Experiments and Results . . . 125
7.3.1 Evaluation Set for Name Disambiguation . . . 125
7.3.2 Experiments and Results . . . 127
7.3.3 Examples of the Name Disambiguation Algorithm in Action . . . 128
7.3.4 Error Analysis . . . 131
7.4 Conclusion and Future Work . . . 132

8 Application: Predicting Organizational Dominance Relations 135
8.1 Introduction . . . 135
8.2 Related . . . 137
8.3 A New Gold Standard for Enron Hierarchy Prediction . . . 140
8.4 Dominance Prediction Technique . . . 141
8.5 Baseline Approaches . . . 142
8.5.1 Unsupervised SNA-Based Approach . . . 142
8.5.2 Supervised NLP-Based Approach . . . 142
8.5.3 Experiments and Results . . . 143
8.6 Dominance Prediction Using the Mention Network . . . 144
8.6.1 Set of Experiments . . . 144
8.6.2 Weighted versus Unweighted Networks . . . 146
8.6.3 Type of Network . . . 147
8.6.4 Linking to the Mentioned Person . . . 149
8.6.5 Type of Degree Centrality . . . 150
8.6.6 Summary: What Matters in Mentions . . . 150
8.7 Conclusion and Future Work . . . 151

IV Extracting Networks from Screenplays **153**

9 Introduction **156**

9.1 Terminology . . . 156
9.2 The Structure of Screenplays . . . 158
9.3 Task Definition . . . 159
9.4 Literature Survey . . . 160
9.5 Conclusion . . . 163

10 Machine Learning Approach **164**

10.1 Data . . . 165
10.1.1 Distant Supervision . . . 165
10.1.2 Heuristics for Preparing Training Data . . . 166
10.1.3 Data Distribution . . . 169
10.2 Baseline Approach . . . 169
10.3 Machine Learning Approach . . . 170
10.3.1 Terminology . . . 171
10.3.2 Overall Machine Learning Approach . . . 171
10.4 Features . . . 174
10.5 Experiments and Results . . . 177
10.5.1 Training Experts . . . 178
10.5.2 Finding the Right Ensemble . . . 179
10.5.3 Feature Analysis . . . 181
10.5.4 Performance on the Test Set . . . 182
10.6 Conclusion and Future Work . . . 184

11 Application: Automating the Bechdel Test **188**

11.1 Introduction . . . 188
11.2 Related Work . . . 191
11.3 Data . . . 193
11.4 Test 1: are there at least two named women in the movie? . . . 194

11.4.1 Resources for Determining Gender . . . 194
11.4.2 Results and Discussion . . . 195
11.5 Test 2: Do these women talk to each other? . . . 198
11.6 Test 3: Do these women talk to each other about something besides a man? . 200
11.6.1 Feature Set . . . 200
11.6.2 Baseline . . . 202
11.6.3 Evaluation and Results . . . 203
11.6.4 Discussion . . . 205
11.7 Evaluation on the End Task . . . 206
11.8 Conclusion and Future Work . . . 207

V Conclusions **209**
11.9 Summary of Contributions . . . 210
11.10New Datasets . . . 213
11.11Limitations and Future Work . . . 213
11.12Other Future Work . . . 217

VI Bibliography **218**

Bibliography **219**

VII Appendices **243**

A Support Vector Machines and Convolution Kernels **244**
A.1 Support Vector Machines . . . 244

B Frame Semantic Rules **246**
B.1 Semantic Rules . . . 246

C List of Features for Bechdel Test **255**
C.1 Frame Features and their Counts for the Bechdel Test . . . 255

C.2 Bag of Terminology used for the Bechdel Test 260

List of Figures

1.1 Association between number of substantive areas specified and year of publication of social network research. This figure and caption is taken from Freeman 2004. 3

1.2 A scene from the movie *Hannah and Her Sisters*. The scene shows one conversation between two characters, **Mickey** and **Gail**. 8

2.1 Diagrammatic illustration of the definition of a social event 16

2.2 Social networks as a result of extracting social entities and social events from example sentences in Table 2.1 18

2.3 Subcategories of the two social events OBS and INR. 19

3.1 Categorization of related work on relation extraction. 40

4.1 Snippet of an offset XML file distributed by LDC. 58

4.2 Snapshot of Callisto. Top screen has the text from a document. Bottom screen has tabs for Entities, Entity Mentions etc. An annotator selected text *said*, highlighted in dark blue, as an event of type OBS.FAR between entities with entity ID E1 and E9. 59

4.3 Set of decisions that an annotator makes for selecting one of seven social event categories. 61

4.4 A full dependency parse for the sentence *Military officials say a missile hit his warthog and he was forced to eject.* There is an OBS social event between the entities **Military officials** and **his**. The SYNRULE baseline makes the correct prediction because path $P12$ exists but path $P21$ does not exist. . . . 67

4.5 A full dependency parse for the sentence *He had to say to her*. There is an INR social event between the entities **he** and **her**. The SynRule baseline makes an incorrect prediction; it predicts $\overrightarrow{OBS}$. This is because the path $P12$ exists but path $P21$ does not exist. 68

4.6 A full dependency parse for the sentence *On behalf of republican candidates and I tend to do a lot of campaigning in the next year for the president.* There is an $\overrightarrow{OBS}$ social event from **I** to **the president**. The SynRule baseline makes an incorrect prediction; it predicts NoEvent. This is because neither of the paths $P12$ and $P21$ exists. 69

4.7 Tree kernel data representations proposed by Nguyen *et al.* 2009. This figure is taken from their paper. The dotted subtree in (a) is referred to as PET (path enclosed tree), the tree in (c) is referred to as DW (dependency word), the tree in (d) is referred to as GR (grammatical role), and the tree in (e) is referred to as GRW (grammatical role word). 73

4.8 Two overlapping scenarios for frame annotations of a sentence, where $F1$ and $F2$ are frames. 76

4.9 Semantic trees for the sentence "Coleman claimed [he]$_{T1-Ind}$ bought drugs from the [defendants]$_{T2-Grp}$, but offered little or no supporting evidence.". This example is annotated as INR. Clearly, if two entities are in a commercial transaction, they are mutually aware of each other and of the transaction taking place. The tree on the left is FrameForest and the tree on the right is FrameTree. $\triangle$ in FrameForest refers to the boxed subtree. **Ind** refers to individual and **Grp** refers to group. 77

4.10 Precision, Recall, and F1 measure for the CoOccurN baseline. X-axis denotes the number of co-occurrences of entities in a document. 82

4.11 Semantic parse for the sentence *Toujan Faisal said [she] {was informed} of the refusal by an [Interior Ministry committee].* 85

6.1 A sample email from the Enron email corpus. The email is from **Sara Shackleton** to **Mark Taylor** regarding "attorney workload". The email contains first name references of five entities (all highlighted): *Mary*, *Frank*, *Brent*, and *Cheryl*. 112

6.2 Email network (thick arrow) and mention network (thin arrows) for the sample email in Figure 6.1. 113

7.1 A graph showing the shortest paths of the top three candidates for the mention *Chris* from the sender, **Jeff Gobbell**, and the recipients, **Tom Martin** and **Cindy Knapp**. See Table 7.5 for the legend. The three candidates are **Chris Barbe** (at a joint distance of 6 from the sender and the recipients), **Chris Stokley** (at a joint distance of 8), and **Chris Gaskill** (at a joint distance of 9). **Chris Barbe** is the correct prediction. 129

7.2 A graph showing the shortest paths of the top five candidates for the mention *Gary* from the sender, **Clem Cernosek**. The five candidates are **Gary Hanks**, **Gary E. Anderson**, **Gary Lamphier**, **Gary Hickerson**, and **Gary Smith**. The mention *Gary* refers to the entity **Gary E. Anderson**, who is at a joint distance of 2 from the sender. Our name disambiguation algorithm makes an incorrect prediction. It predicts **Gary Hanks** to be the referent who is at a shorter joint distance of 1. 130

7.3 A graph showing the shortest paths of the top six candidates for the mention *Philip* from the sender, **Jason Williams**, and the recipient, **Spiro Spirakis**. The six candidates are **Philip Rouse**, **Philip Polsky**, **Philip Warden**, **Willis Philip**, **Philip Sutterby**, and **Philip Bacon**. The mention *Philip* refers to the entity **Philip Bacon**, who is at a distance 9 from the sender and the recipient. Our name disambiguation algorithm is unable to make a prediction because there are two candidates, **Philip Rouse** and **Philip Polsky**, at the same joint distance of 5. 133

7.4 Name disambiguation error caused to do entity normalization error. There are three candidates: **joe.parks@enron.com** at a joint distance of 5 from the sender and the recipients, **joe.parks@bridgeline.net** also at a joint distance of 5, and **joe parks** at a joint distance of 7. The ground truth is **joe parks**. The sender is **knipe3**, the red node, and the recipients are **brian constantine**, **cmccomb**, **erik wollam**, and **keith mccomb**, the blue nodes. Clearly, if the three different ways of referring to **Joe Parks** is normalized to one entity, name disambiguation will make the correct prediction. 134

9.1 A scene from the movie *Hannah and Her Sisters*. The scene shows one conversation between two characters, **Mickey** and **Gail**. The line tagged with the tag S is a scene boundary, lines tagged with the tag N belong to a scene description, lines tagged with the tag C contain the names of speaking characters, lines tagged with the tag D contain the dialogue spoken by these characters, and lines containing all the remaining information are tagged using the tag M. 158

10.1 Screenplay snippet from the movie *Sleepy Hollow*. Scene boundaries and scene descriptions are at five levels of indentation. Character names are at 29 levels of indentation and the dialogues they speak are at 15 levels of indentation. . . 167

10.2 Overall machine learning approach for parsing screenplays. 173

10.3 Example screenplay: first column shows the tags we assign to each line in the screenplay. M stands for "Meta-data", S stands for "Scene boundary", N stands for "Scene description", C stands for "Character name", and D stands for "Dialogue." We also show the lines that are at context -2 and +3 for the line "CRAWFORD." 177

10.4 Learning curves for training on TRAIN_000 and testing on DEV1_000. X-axis is the % of training data, in steps of 10%. Y-axis is the macro-F1 measure for the five classes. Each learning curve belongs to a particular value of CONTEXT. 179

10.5 Network created from the screenplay parsed using the rule based baseline for the movie *Silver Linings Playbook*. 185
10.6 Network created from the screenplay parsed using our machine learning model for the movie *Silver Linings Playbook*. 186
10.7 Network created from the screenplay that was manually annotated by a human for the movie *Silver Linings Playbook*. 187

11.1 A scene from the movie *Hannah and Her Sisters*. The scene shows *one* conversation between two *named women* **Mickey** and **Gail**. Tag S denotes scene boundary, C denotes character mention, D denotes dialogue, N denotes scene description, and M denotes other information. 192
11.2 The network of the main characters from the movie *Up In The Air*. The set of women = {Julie, kara, Alex, Natalie}. The set of men = {Ryan, Craig Gregory}. 203
11.3 Distribution of three **SNA** features (top to bottom): mean degree centrality, mean closeness centrality, and mean betweenness centrality of named women. Red histogram is for movies that fail and the Blue histogram is for movies that pass the third Bechdel Test. The histograms show that the average centralities of women are higher for movies that pass the Bechdel test. 206

List of Tables

2.1 Examples of social event mentions in different types of text. Entity mentions (that participate in a social event) are enclosed in square brackets [...] and words that trigger a social event are enclosed in set brackets {...}. 17

2.2 This table maps the type and subtype of ACE events to our types and sub-types of social events. The columns have ACE event types and sub-types. The rows represent our social event types and sub-types. The last column is the number of our events that are not annotated as ACE events. The last row has the number of social events that our annotator missed but are ACE events. 30

3.1 Citations for work on relation extraction. 41

3.2 Feature set introduced by Kambhatla 2004 for the ACE relation extraction task. This table is taken as-is from their paper. 45

3.3 Types of string kernels and their implicit feature space for the string "cat". Both these kernels were introduced by Lodhi *et al.* 2002. 47

3.4 Data representations for string kernels. 48

3.5 Types of tree kernels and their implicit feature space for the tree [.A [.B E] [.C F] [.D G]] (framed tree in the second row of the table). The plus sign (+) indicates "in addition to above structures." So the implicit feature space of SST is smaller than that of CT and the feature space for CT is smaller than that of ST. 50

3.6 Data representations for tree kernels. 54

4.1 This table presents two inter-annotator agreement measures (Kappa in Column 7 and F1 measure in the last column). Columns 3-6 show the flattened confusion matrix for each decision point. Y, Y refers to Yes, Yes i.e. both annotators say Yes to a question in Column 2. Y, N refers to Yes, No i.e. the first annotator says Yes but the second annotator says No to a question, and so on. 62
4.2 Data distribution of our gold standard. 81
4.3 A comparison of performance of all the baselines with our best performing system for the task of Social Event Detection. † refers to a novel kernel combination. The basic structures in this combination have been proposed by Nguyen et. al 2009. ⋆ refers to the new structures and combinations we propose in this work. 89
4.4 A comparison of performance of all the baselines with our best performing system for the task of Social Event Classification. † refers to a novel kernel combination. The basic structures in this combination have been proposed by Nguyen et. al 2009. ⋆ refers to the new structures and combinations we propose in this work. 90
4.5 A comparison of performance of all the baselines with our best performing system for the task of Directionality Classification. † refers to a novel kernel combination. The basic structures in this combination have been proposed by Nguyen et. al 2009. ⋆ refers to the new structures and combinations we propose in this work. 91
4.6 A comparison of performance of all the baselines with our best performing system for the overall task of Social Network Extraction. † refers to a novel kernel combination. The basic structures in this combination have been proposed by Nguyen et. al 2009. ⋆ refers to the new structures and combinations we propose in this work. 92

5.1 A comparison of the number of links in the two gold standards. 98
5.2 Performance of the two systems on the two gold standards. 99

5.3 Hypotheses and results. All correlations are statistically significant. $\sim$ is to be read as *does not change significantly*. As an example, hypothesis H0 is to be read as: *As settings go from rural to urban ... the number of characters does not change significantly*. Grayed out boxes are not valid hypotheses. For example, *As # of characters* $\uparrow$ *..... # of characters* $\sim$ is not a valid hypothesis. 105

7.1 A table showing the full names of four entities and their name variants. All four entities have the same first name, *Chris*. Each entity is assigned an identifier ranging from E1 through E4. 122

7.2 The name variant map Map_{nv} for the entities in Table 7.1. 123

7.3 An email from **jgobbel@flash.net** to **Cindy Knapp**, **Tom Martin**, and **Chris Barbe**. The content of the email mentions *Chris*, whose true referent is one of the recipients, **Chris Barbe**. 127

7.4 A comparison of performance of name disambiguation techniques. 127

7.5 Legend for the graphs in Figure 7.1, 7.2, and 7.3. 129

8.1 Results of four experiments comparing the performance of purely NLP-based systems with simple SNA-based systems on two gold standards G and $T \in G$. 143

8.2 Some examples of terminology used in this paper to refer to different types of systems. 145

8.3 Two examples in which the degree centrality measure in an unweighted network makes the correct prediction compared with its weighted counter-part (Section 8.6.2). Asterisk (*) denotes higher up in the hierarchy. 147

8.4 Results for the best performing systems based on three different network types and evaluation groups. 148

8.5 Two examples in which the degree centrality measure in an mention-only network makes the correct prediction compared with the email-only network (Section 8.6.3). Asterisk (*) denotes higher up in the hierarchy. 149

8.6 Three examples . showing the importance of linking recipient and the mentioned (Section 8.6.6) Asterisk (*) denotes higher up in the hierarchy. 151

10.1 Data distribution . . . 169
10.2 Common types of anomalies found in screenplays and our encoding scheme. . 172
10.3 The complete set of features used for parsing screenplays. . . . 175
10.4 Data distribution . . . 178
10.5 Comparison of performance (macro-F1 measure) of our rule based baseline with our machine learning based models on development sets DEV1_000, DEV1_001, ..., DEV1_111. All models are trained on 50% of the training set, with the feature space including CONTEXT equal to 1. . . . 180
10.6 Macro-F1 measure for the five classes for testing on DEV2 set. 000 refers to the model trained on data TRAIN_000, 001 refers to the model trained on data TRAIN_001, and so on. MAJ, MAX, and MAJ-MAX are the three ensembles. The first column is the movie name. LTC refers to the movie "The Last Temptation of Christ." . . . 180
10.7 Performance of MAJ-MAX classifier with feature removal. Statistically significant differences are in bold. . . . 181
10.8 A comparison of performance of our rule based baseline with our best machine learning model on the five classes. . . . 183
10.9 A comparison of network statistics for the three networks extracted from the movie *Silver Linings Playbook*. . . . 184
10.10A comparison of Pearson's correlation coefficients of various centrality measures for $\mathcal{N}_B$ and $\mathcal{N}_{\text{MAJ-MAX}}$ with $\mathcal{N}_G$. . . . 184

11.1 Distribution of movies for the three tests over the training/development and test sets. . . . 193
11.2 Results for **Test 1**: "are there at least two named women in the movie". . . . 196
11.3 Results for **Test 2**: "do these women talk to each other?" . . . 198
11.4 An example of a screenplay (movie *Wanted*, 2008) in which a scene description divides a long scene into two sub-scenes with a different set of conversing characters. Characters **Nicole** and **Janice** never converse with each other in the movie. . . . 199
11.5 Feature values for **SNA** feature vectors (4) and (5). . . . 203

11.6 Feature values for **SNA** feature vectors (6) and (7). 204
11.7 Results for **Test 3**: "do these women talk to each other about something besides a man?" Column two specifies the kernel used with the SVM classifier. 204
11.8 Results on the unseen test set on the end task: does a movie passes the Bechdel Test? 207

Acknowledgments

First and foremost, I am greatly thankful and deeply grateful to my advisor, Dr. Owen Rambow. He provided me with the support to grow and the space to explore. I would like to thank Owen not only for everything he has taught me about science, analysis, and critical thinking, but also for teaching me several softer skills that are necessary for succeeding in the real world.

I am also greatly thankful to Professor Kathy McKeown. She has seen me from the first day of my school at Columbia. She has played an instrumental role in coaching me on how to become a sound researcher. She is a visionary. Under her guidance, I have learned how to look at the bigger picture, both in terms of long term and short term impact. I would like to thank Kathy for her constant support and advice.

Professor Adam Cannon has also seen me from the first day of my school at Columbia. He has been very kind to make time to meet me at least twice a semester throughout my stay at Columbia. He taught me a lot about Mathematics and helped me maneuver through the courses at the Math department. He also gave me an opportunity to teach along with advice on how to be an effective teacher.

I would like to congratulate Professor Rocco Servedio for being such an amazing teacher. His course on Computational Learning Theory was absolutely life changing. I am not sure how much machine learning I really understood before taking this course. I would like to thank him not only for the knowledge he provided during the course but also all the help and advice on machine learning for several semesters after I first took his course.

I would like to thank Professor Clement Hongler and Professor Patrick Gallagher from the Math department for helping me navigate the infinite dimensional spaces in the world of Mathematics. They have played a crucial role in my understanding of the material and gave me a path to get from zero to one.

My experience at Columbia would not have been complete without all the wonderful

professors who introduced me to the world of movement: Caitlin Trainor, Tessa Chandler, Jodi Melnick, and Allegra Kent. It was during this time when I was learning about movement and abstract algebra, that I really fully understood creativity and abstraction.

I would like to thank all the high school, undergraduate, and graduate students who I had the pleasure to work with during these years (in no particular order): Prashant Jayannavar, Jingwei Zhang, Jiehan Zheng, Wisodm Omuya, Anup Kotalwar, Shruti Kamath, Evelyn Rajan, Jing Zhang, Shirin Dey, Jacob Jensen, Augusto Corvalan, Melody Ju, Sarthak Dash, Sriram Balasubramanian, Ari Fay, Linan Qiu, Ashlesha Shirbhate, Renee Mania, Anahita Bhiwandiwalla, and Tejaswini Pedapati. There is no way I could have done research on several topics without their help. I am very grateful for all their contributions.

My stay at Columbia and CCLS would not have been the same without all these amazing people who played the role of a mentor and a friend. I would like to thank Smaranda Muresan, Manoj Pooleery, Axinia Radev, Kathy Hickey, and Idrija Ibrahimagic for helping me in more than one and meaningful ways.

I would also like to thank my dear friends, Vinod Prabhakaran, Mohamed Altantawy, Yassine Benajiba, Daniel Bauer, Or Biran, Heba Elfardy, Ahmed El Kholy, Ramy Eskander, Noura Farra, Weiwei Guo, Yves Petinot, Sara Rosenthal, Wael Salloum, David Elson, Karl Stratos, Kapil Thadani, Ilia Vovsha, and Boyi Xie for their constant support and friendship.

Special thanks to Kapil Thadani and Daniel Bauer. They are pretty fortunate and are probably thanking their stars that I have, at least temporarily, left academia. I used to go to them at the last minute for providing comments on my papers. They have been very kind to accommodate my last minute requests. They have also contributed significantly to my research profile through deep and lengthy discussions.

I would also like to thank my close friends, Amal Masri and Caronae Howell for helping me improve the scientific content and readability of my papers. They are both professional writers and taught me a lot about writing. They have also helped me go through several ups and downs of a demanding doctoral program.

Finally, I would like to thank my family – my mother, father, and sister for their constant support and belief in me. My mother made several trips to New York just to help me out in intense periods of time. My sister, especially when she was living in New York, was

extremely helpful in numerous ways. My father, even though he is yet to come to New York, has been a source of inspiration and constant support.

For My Family

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation

Creating and maintaining social networks are central to human existence. Over the years, researchers have shown the power and utility of social network analysis techniques on a wide range of academic disciplines: crime prevention and intelligence [Sparrow, 1991], psychology [Seidman, 1985; Koehly and Shivy, 1998], management science [Tichy *et al.*, 1979; Cross *et al.*, 2001; Borgatti and Cross, 2003], anthropology [Sanjek, 1974; Johnson, 1994; Hage and Harary, 1983], political science [Knoke, 1990; Brandes *et al.*, 2001], and literary theory [Moretti, 2005]. In fact, as Figure 1.1 shows, the number of substantive areas that utilize social network analysis techniques has been growing linearly [Otte and Rousseau, 2002].

But how are these social networks created for analysis? Freeman [2004] provides a comprehensive historical account of the evolution of techniques used for creating social networks. We summarize Freeman's historical account here. One of the earliest network structures was created by Hobson [1884]. Hobson created a table by hand that showed a two mode network of how five major South African companies were linked by six board members. Almack [1922] used interviews to collect network data about who invites whom to a party. Wellman [1926] collected network data by observing who played with whom among pre-school children. Elizabeth Hagman brought these two approaches (interview and observation) together in 1933.

In the pre-digital, pre-internet era, there was really no other way of creating social net-

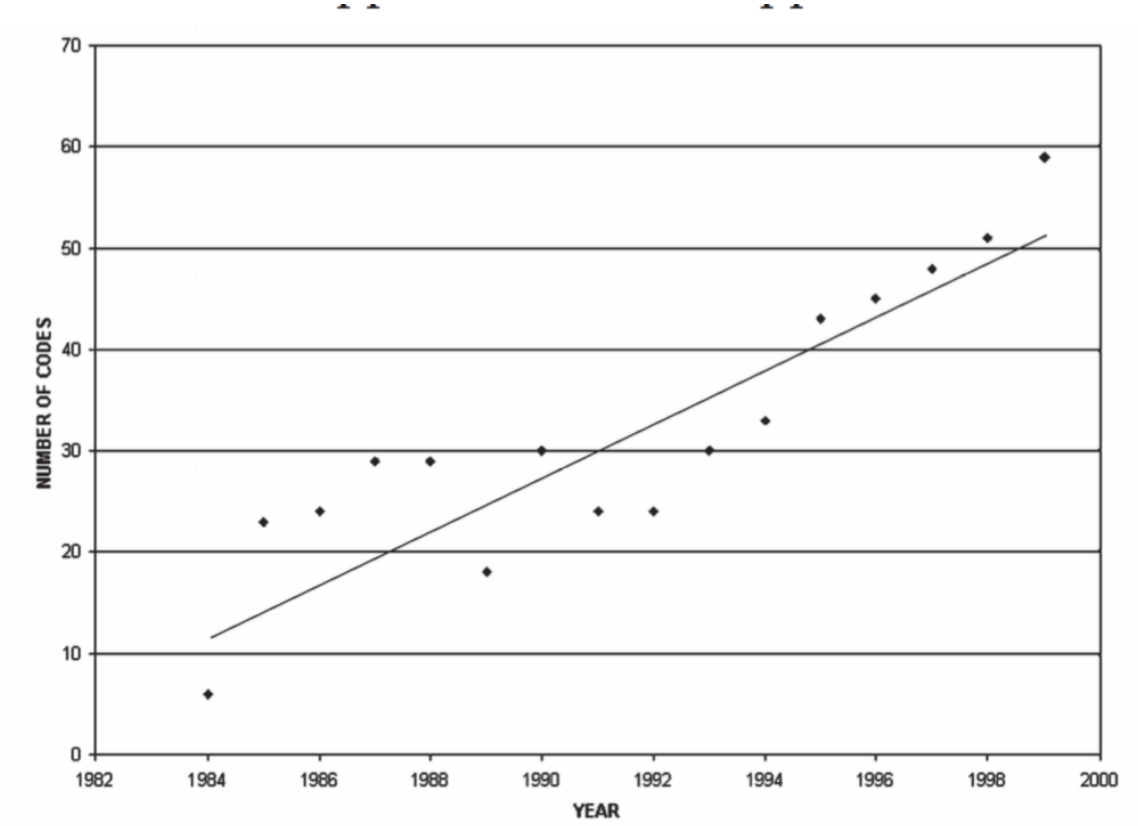


Figure 1.1: Association between number of substantive areas specified and year of publication of social network research. This figure and caption is taken from Freeman 2004.

works other than surveys, interviews, and observations. However, in this digital age, one can collect many kinds of social network data without explicitly interviewing, observing or surveying people. The Stanford Large Network Dataset Collection [Leskovec and Krevl, 2014] presents a wide variety of datasets for social network analysis. For example, the *ego-Facebook* network is an undirected network consisting of 4039 nodes and 88234 edges representing social circles from Facebook. The *soc-Epinions1* network is a directed network representing who-trusts-whom on Epinions.com. The *com-Friendster* network is an undirected network with communities containing over 65 million nodes and 1.8 billion edges representing the social network on Friendster. There are a total of 79 datasets listed on the website. All of these networks are created using *meta-data* information. For example, people become friends on the Friendster website by adding each other as friends. This information about

self-declared friendship is recorded as a link in the meta-data fields on the website. Similarly, if one person emails another person, this information is recorded in the meta-data fields of the email that is easy to extract. This information can be used to create links between the sender and the recipients (sender-recipient links).

Typically researchers construct a social network from various forms of electronic interaction records like self-declared friendship links, sender-recipient email links, knowledge about membership to the same community, etc. However, a vastly rich network that is present in the content of some of these sources is missed. As an example, if **Mary** emails **Cheryl** and *talks about* her telephonic interactions with **John**, **Mary's** interactions with **John** may not expressed through email meta-data, and these interactions are thus missed using meta-data based techniques of creating social networks. Furthermore, several rich sources of social networks remain untouched simply because there is no meta-data associated with them (literary texts, movie screenplays, among several others).

A scientific work that highlights the absence of techniques to mine social networks from unstructured texts and that highlights the importance of doing so is by Franco Moretti. In this work, [Moretti, 2005] [Moretti, 2011] [Moretti, 2013], Franco Moretti constructs networks by hand and proposes a radical transformation in the study of literature. The following quote by Moretti [2011] establishes the fact that there was no technique (at least until 2011) that could have been used to mine interaction networks from unstructured texts such as literary texts.

> First, the edges are not "weighted": when Claudius tells Horatio in the graveyard scene, "I pray thee, good Horatio, wait upon him", these eight words have in this Figure exactly the same value as the four thousand words exchanged between Hamlet and Horatio. This can't be right. And then, the edges have no "direction": when Horatio addresses the Ghost in the opening scene, his words place an edge between them, but of course that the Ghost wouldn't reply and would only speak to Hamlet is important, and should be made visible. But, I just couldn't find a non-clumsy way to visualize weight and direction; and as a consequence, the networks in this study were all made by hand, with the very simple aim of maximizing visibility by minimizing overlap. This is not a long term solution, of

> course, but these are small networks, where intuition can still play a role; they're like the childhood of network theory for literature; a brief happiness, before the stern adulthood of statistics.

Moretti discusses several consequences of enabling *distant reading* versus the traditional *close reading* of literary texts. We quote one of the consequences here [Moretti, 2011]:

> Third consequence of this approach: once you make a network of a play, you stop working on the play proper, and work on a model instead: you reduce the text to characters and interactions, abstract them from everything else, and this process of reduction and abstraction makes the model obviously much less than the original object – just think of this: I am discussing Hamlet, and saying nothing about Shakespeare's words – but also, in another sense, much more than it, because a model allows you to see the underlying structures of a complex object.

Distant reading, as Moretti suggests, allows the study of literature at a new level – a level that uncovers the socio-structural aspects of societies built by authors in their stories and settings.

1.2 A High Level Organization of the Thesis

The goal of this thesis is to introduce techniques for extracting social networks from unstructured and semi-structured texts which encode rich social networks that are inaccessible through traditional techniques of creating social networks. We propose three novel techniques for mining social networks from three types of texts: unstructured texts (such as literary texts), emails, and movie screenplays. For each of these types of texts, we demonstrate the utility of the extracted networks on three applications (one for each type of text).

This thesis is divided into three parts. The theme that ties these three parts together is the overall goal – to develop techniques for automatically extracting social networks from unstructured and semi-structured texts. We introduce a new kind of social network – a network in which nodes are people and links are what we call *social events*. Two entities

(of type person) are said to participate in a social event if at least one of the entities is cognitively aware of the other. For example, in the sentence, *John said Mary has a beautiful bag*, **John** is cognitively aware of **Mary** (or has **Mary** in his mind because he is talking about **Mary**). We say there is a social event directed from **John** to **Mary**. In the sentence, *John and Mary are having dinner together*, both entities are mutually aware of one another and of each others' mutual awarenesses. We say there is a bidirectional social event between the two entities. Our definition of social networks is grounded in the most basic building blocks of relationships – cognition. We claim that social events are the smallest possible, the most rudimentary building blocks for more complex social relationships such as friendships. People have to be cognitively aware of each other for building and maintaining complex social relations. Our notion of social events grounds the definition of social networks in the most basic building blocks of relationships – cognition. We claim that social events are the smallest possible, the most rudimentary building blocks for more complex social relationships such as friendships. People have to be cognitively aware of each other for building and maintaining complex social relations. We hope that our nomenclature serves as a unifying definitional platform for other types social networks.

The first part of this thesis introduces a novel machine learning approach for automatically extracting these social networks from unstructured texts. Unstructured texts such as newspaper articles or novels often tell stories about people (real or fictional). These stories revolve around people and social events between these people. Social events aggregate over the course of the story to form a social network. In order to automatically extract social networks – the aggregation of social events – we build models to detect and classify social events that are expressed using language. For example, given the sentence, *John and Mary are having dinner together*, we want our models to detect a social event between the two entities and classify the social event as an interaction event. We use these models for extracting social networks from nineteenth century British literature and study some of the long standing literary theories that comment on the structure of social networks in novels. We refer to this system as SINNET[1] for the rest of this thesis.

[1]SINNET stands for Social Interaction Network Extraction Tool. Sinnet is a type of rope that is made by plaiting strands of grass. It is used for tying things together.

The second part of this thesis introduces a novel technique for extracting social networks from electronic mails (emails). Emails, unlike raw text, have a structure; they contain meta-data information (that is well structured with fields such as *to*, *from*, *cc*, *subject*) and content (that is largely unstructured). By utilizing the well structured meta-information, specifically the fields *to*, *from*, *cc*, and *bcc*, one can easily create a social network of "who sends emails to whom." However, there is a rich social network in the unstructured content of emails; people *talk about* other people in the content of emails. By virtue of talking about other people, there is a social event directed from the sender to the mentioned person (and from the recipients to the mentioned person once the email is read or replied to). To extract these "who talks about whom" links, we must first resolve the people being *talked about* to real people. For example, in an email from **Marie Heard** to **Sara Shackleton** that *mentions* a person named *Jeff*, we must first determine the referent of this mention. After all, there may be hundreds of people with *Jeff* as their first name (as is the case in the Enron email corpus). The problem of extracting social networks from emails thus poses a new challenge – we need a mechanism to disambiguate entities mentioned in the content of emails to real people in the network. In monolithic, coherent bodies of text, such as novels, it is unlikely that two different characters are referred using the same name. In organizational emails, however, this phenomenon is common. An organization may have hundreds of people with *Jeff* as their first name who are referred as *Jeff* in several emails. To this end, we introduce a novel technique for disambiguating named mentions to real people in an email network. We use this technique for extracting what we call the *mention* network (a mention link is a type of social event, specifically an observation social event). We demonstrate the utility of the mention network on an extrinsic task that is about predicting organizational dominance relations between employees of the Enron corporation.

The third and final part of this thesis introduces a novel technique for extracting social networks from movie screenplays. Screenplays are text documents written by screenwriters for the purposes of storytelling. But unlike novels, which tell a story using free flow text, screenplays tell a story in a text format that is highly structured. For example, screenplays are segmented into scenes and each scene starts with an indicator INT. or EXT. Scenes contain dialogues between characters that are clearly marked using other textual and for-

matting indicators (see Figure 1.2). Given a well-structured screenplay, creating a network of interactions of characters is trivial – we know the position of scene boundaries, characters, and their dialogues – connecting all conversing characters in a scene with interaction links gives the social network. However, screenplays found on the web are ill-structured. We show that identifying scene boundaries, characters, and their dialogues using regular expressions is not sufficient for creating an interaction network. We propose a novel machine learning approach for automatically recovering the structure of screenplays. This allows us to extract social networks, where nodes are characters and links are a type of social events (interaction social event). We utilize these networks for a novel NLP application of automating the Bechdel Test.

```
INT. MICKEY'S OFFICE - NIGHT

Gail, wearing her glasses, stands behind a crowded but well-
ordered desk.  Two assistants, a man and a woman, stand around
her.

                    MICKEY
          (turning to Gail,
          gesturing nervously)
Sssss, if I have a brain tumor, I
don't know what I'm gonna do.
          (sighing)

                    GAIL
You don't have a brain tumor.  He
didn't say you had a brain tumor.

                    MICKEY
          (sighing)
No, naturally
```

Figure 1.2: A scene from the movie *Hannah and Her Sisters*. The scene shows one conversation between two characters, **Mickey** and **Gail**.

1.3 Contributions in Terms of Techniques

- **Extracting social networks from unstructured texts**:
  - One of the main contributions of this thesis is the development of a technique

for extracting social networks from unstructured texts such as literary novels. We take motivation from the relation extraction community and use convolution kernels (subsequence and tree) for developing this technique.

– We show that convolution kernels are task independent. This is a nice property to have because the same kernel representations may be used for different tasks (relation extraction and social network extraction). In fact, SINNET is now being used in the DEFT project at Columbia University for an entirely new task of source-and-target belief and sentiment detection. In contrast, we show that fine grained feature engineering based approaches do not adapt well to a new task. They tend to be task dependent.

– We experiment with a wide variety of data representations already introduced for relation extraction and propose four new structures: one subsequence structure that is a sequence of nodes on a special dependency tree (details deferred to later) and three tree kernel representations that attempt to combine the feature spaces from all levels of language abstractions (lexical, syntactic, and semantic). By semantics we mean frame semantics, specifically the ones derived from the FrameNet annotations. We further introduce a set of linguistically motivated hand-crafted frame semantic features and compare their performance with other baselines. Our results show that hand-crafted frame semantic features add less value to the overall performance in comparison with the frame-semantic tree kernels. We believe this is due to the fact that hand-crafted features require frame parses to be highly accurate and complete. In contrast, tree kernels are able to find and leverage less strict patterns without requiring the semantic parse to be entirely accurate or complete.

– For training and testing our methods, we provide social event annotations on a well-known and widely used corpus distributed by the Linguistic Data Consortium (LDC) called the Automatic Content Extraction (ACE) 2005 Multilingual Training Corpus.[2] We refer to this corpus as the **ACE-2005 corpus** throughout

[2] `https://catalog.ldc.upenn.edu/LDC2006T06`. LDC Catalog number: LDC2006T06

this document. The ACE-2005 corpus contains annotations for entities, entity mentions, ACE relations, and ACE events. The data sources in the corpus come from weblogs, broadcast news, newsgroups, broadcast conversation. We overlay our social event annotations onto the dataset and make it available for download in LDC's standard offset annotation format.

- **Extracting social networks from Emails**:
  - We introduce a novel unsupervised technique for resolving named mentions in emails to real people in the organization. We use this technique for extracting the mention network – a new kind of network that has not been explored for applications in the past.
- **Extracting social networks from movie screenplays**:
  - We introduce the first NLP and ML based system for extracting social networks from movie screenplays. Our system outperforms the previously proposed regular expression and grammar based systems by large and significant margins. The models we propose may also be applied for extracting networks from other types of screenplays such as drama and theatrical play screenplays.
  - One of the main challenges in building a system for automatically parsing screenplays (which is required for extracting a social network) is the absence of training data. We propose a novel methodology for automatically obtaining a large and varied sample of annotated screenplays. This methodology is inspired by the distant learning paradigm. For different types of anomalies, we *perturb* the training data and train separate classifiers that are experts in handling certain combinations of possible anomalies. We combine these experts into one classifier using ensemble learning techniques. We believe that our general technique may be applied for automatically parsing other types of documents that are supposed to be well-structured but are not, for example, emails that are converted to text using optical character recognition techniques.

1.4 Contributions in Terms of Applications

- **Validating literary theories**:
  - Elson *et al.* [2010] previously introduced the task of computationally validating literary theories that assume a structural difference between the social worlds of rural and urban novels using *conversational* networks extracted from nineteenth-century British novels. We revisit these theories and employ SINNET for extracting social networks from these literary texts. SINNET extracts interactional links (a conceptual generalization of conversational links) and a new class of links called observational links (details deferred to later). This allows us to examine a wider set of hypotheses and thus provide deeper insights into literary theories.
  - We present an evaluation of the system on the task of automatic social network extraction from literary texts. Our results show that SINNET is effective in extracting interaction networks from a genre that is quite different from the genre it was trained on, namely news articles.
  - For evaluating SINNET, we introduce a dataset that consists of social event annotations on the four excerpts introduced by Elson *et al.* [2010] for the evaluation of their system.
- **Predicting organizational dominance relations**:
  - The task of predicting dominance relation between pairs of employees in the Enron email corpus is well-studied [Rowe *et al.*, 2007; Diehl *et al.*, 2007; Creamer *et al.*, 2009; Bramsen *et al.*, 2011; Gilbert, 2012; Wang *et al.*, 2013; Prabhakaran and Rambow, 2014]. We propose a social network analysis based technique that outperforms previously proposed techniques by a large and significant margin. We highlight one of the major limitations of using a natural language processing based system for the task of dominance prediction. The limitation is related to the fact that we seldom have access to entire email collections and it is thus impractical to assume the presence of communications between all possible pairs of employees.

- We utilize the mention network for predicting dominance relations between employees and show that it performs better than the more commonly used email network. Through a comprehensive set of experiments, we provide evidence for a new finding about the Enron corpus – *you're the boss if people get mentioned to you*. We find that people who receive emails that contain a lot of mentions to other people are the boss. We believe this finding may be attributed to the corporate reporting culture in which managers report to their superiors about the performance of their team (thus mentioning a high volume of people in the emails to their superiors).
- Through this work, we introduce the largest known gold standard for both dominance and hierarchy prediction of Enron employees. Previously used gold standards contain dominance relations of only 158 Enron employees. The gold standard we introduce contains dominance relations and hierarchy relations of 1518 Enron employees.[3]

- **Automating the Bechdel Test**:
  - The Bechdel Test is a sequence of three questions designed to assess the presence of women in movies. Many believe that because women are seldom represented in film as strong leaders and thinkers, viewers associate weaker stereotypes with women. We present the first computational approach to automating the task of finding whether or not a movie passes the Bechdel test. This automation allows us to study the key differences in the importance of roles of women in movies that pass the test versus the movies that fail the test. Our experiments confirm that in movies that fail the test, women are in fact portrayed as less-central or less-important characters.

[3]The corpus may be downloaded from `http://www1.ccls.columbia.edu/~rambow/enron/`.

1.5 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 of this first part of the thesis introduces our working definition of social networks for the rest of the thesis. We introduce the notion of social events and differentiate this notion from other notions of events and types of links that may be used for creating a social network. Part II introduces a technique for– and an application of– extracting social networks from unstructured texts. This part of the thesis is organized as follows: Chapter 3 introduces the task definition along with literature survey on relation extraction, Chapter 4 provides details about the data, our annotation effort, our machine learning approach, and experiments, Chapter 5 presents an application of automatic social network extraction for validating literary theories.

Part III introduces a technique for– and an application of– extracting social networks from electronic mails. This part of the thesis is organized as follows: Chapter 6 introduces the terminology regarding emails, their structure, and the problem definition, Chapter 7 presents our unsupervised approach to resolving named mentions to real people, and Chapter 8 uses these extracted networks for predicting the organizational dominance relations between employees of the Enron corporation.

Part IV introduces a technique for– and an application of– extracting social networks from movie screenplays. This part is organized as follows: Chapter 9 introduces the terminology regarding screenplays, their structure, and the problem definition, Chapter 10 presents our machine learning approach for recovering the structure of screenplays for extracting interaction networks, Chapter 11 uses these extracted networks for automating the *Bechdel Test*. We conclude and present directions for future work of the thesis in Part V.

Chapter 2

A New Kind of a Social Network

The Oxford dictionary defines a social network as follows:

> [1] A network of social interactions and personal relationships.
>
> [2] A dedicated website or other application that enables users to communicate with each other by posting information, comments, messages, images, etc.

This thesis is concerned with the first definition of a social network – a network of social interactions and personal relationships. This definition is in harmony with the definition of a social network that Wasserman and Faust [1994] provide: a social network is a network of *social entities* (such as people and organizations) and their *relationships*. Wasserman and Faust [1994] also provide a list of kinds of relationships that a social network analysis study might include. This list includes relationships such as individual evaluations (friendship, liking, respect), transactions or transfer of material resources (buying, selling), transfer of non-material resources (communications, sending receiving information), interactions, kinship. In this thesis, we introduce a novel kind of link called *social event* which aggregates to form more complex social relations. This section provides a formal definition of social events and differentiates this notion from related kinds of relationships. These definitions were first introduced in Agarwal *et al.* [2010].

2.1 Definition of Entities

We borrow the definition of entity and entity mention from the Automatic Content Extraction (ACE) guidelines. According to the ACE Entity annotation guidelines[1]:

> An entity is an object or set of objects in the world. A mention is a reference to an entity. Entities may be referenced in a text by their name, indicated by a common noun or noun phrase, or represented by a pronoun. For example, the following are several mentions of a single entity:
>
> Name Mention: Joe Smith
>
> Nominal Mention: the guy wearing a blue shirt
>
> Pronoun Mentions: he, him

ACE defines seven broad categories of entities: PERSON, ORGANIZATION, GEO-POLITICAL, LOCATION, FACILITY, VEHICLE, and WEAPON. ACE further defines subtypes for the entity type PERSON: Individual (PER.INDIVIDUAL) and Group (PER.GROUP). Since we are only concerned with networks between people and groups of people, throughout this document, we take an entity to mean an entity of type PERSON (PER) with subtypes Individual (PER.INDIVIDUAL) and Group (PER.GROUP).

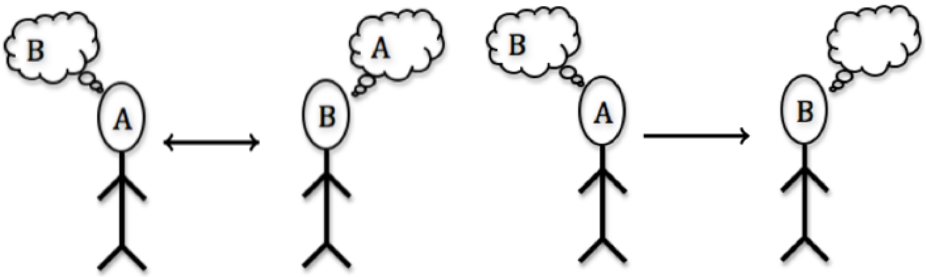
2.2 Definition of Social Events

Two entities are said to participate in a social event if at least one entity is cognitively aware of the other. We define two broad categories of social events: (1) Observation (OBS) and (2) Interaction (INR) . Observation is a unidirectional social event in which only one entity is cognitively aware of the other. Interaction is a bidirectional social event in which both entities are cognitively aware of each other *and* of their mutual awarenesses. For example, in the sentence, *John is talking to Mary about Sara*, there is an OBS social event directed from **John** to **Sara** (because **John** is talking about **Sara** and is thus cognitively aware of her and there is no evidence that **Sara** is mutually aware of **John**), another OBS

[1] http://nlp.cs.rpi.edu/kbp/2014/aceentity.pdf

social event directed from **Mary** to **Sara** (because **Mary** is hearing about **Sara** and is thus cognitively aware of her and there is no evidence that **Sara** is mutually aware of **Mary**), and an INR social event between entities **John** and **Mary** (because **John** and **Mary** are having a conversation in which both are aware of each other and each others' awarenesses).

Figure 2.1 diagrammatically illustrates the definition of a social event. There are two entities, A and B. Thought bubbles represent cognitive states of these entities. In the interaction social event (Figure 2.1a), entity A is aware of entity B, entity B is aware of entity A, and the two entities are mutually aware of their awarenesses. In the observation social event (Figure 2.1b), only entity A is aware of entity B.



(a) Interaction (INR) (b) Observation (OBS)

Figure 2.1: Diagrammatic illustration of the definition of a social event

In the definition of INR, the point about the entities being aware of each others' mutual awarenesses is crucial. Consider a hypothetical situation in which one entity, say **John**, is spying on another entity, say **Mary**. **John** thinks **Mary** is unaware of the *spying* event. As it turns out, **Mary** is in fact aware of being spied upon by **John**. This hypothetical situation gives rise to two OBS events, one from **John** to **Mary** and the other from **Mary** to **John**, instead of one INR event between the two entities.

Table 2.1 presents examples of social events in different types of text. Figure 2.2 shows the corresponding social networks that result from extracting entities and social events from example sentences in Table 2.1. In these networks, nodes are entities and links are social events that appear between these entities.

In the first example in Table 2.1 (news article), **Faisal** is *talking about* the **committee** (triggered by the word *said*). While there is evidence that **Faisal** has the **committee**

Source	Text	Social Event
News article	[Toujan Faisal], 54, {said}$_{OBS}$ [she] was {informed}$_{INR}$ of the refusal by an [Interior Ministry committee] overseeing election preparations.	OBS from Faisal to committee. INR between Faisal and committee.
Novel	"[Emma] never thinks of herself, if she can do good to others," {rejoined} [Mr. Woodhouse]	OBS from Mr. Woodhouse to Emma
Email	An email from [Kate] to [Sam]: [Jacob], the City attorney had a couple of questions ...	INR between Kate and Sam. OBS from Kate to Jacob. OBS from Sam to Jacob.
Film screenplay	[BOURNE] imploding, [the kids] {staring} at him... .....[BOURNE] .....Who do you think sent me? .....[WOMBOSI] .....I know who sent you. I don't .....know why.	OBS from the kids to Bourne. INR between Bourne and Wombosi.

Table 2.1: Examples of social event mentions in different types of text. Entity mentions (that participate in a social event) are enclosed in square brackets [...] and words that trigger a social event are enclosed in set brackets {...}.

in her cognitive state, there is no evidence that the **committee** also has **Faisal** is their cognitive state. Therefore, there is a unidirectional OBS link from **Faisal** to the **committee**. However, in what **Faisal** is saying, there is an INR social event, triggered by the word *informed*. Figure 2.2 (a) shows the two nodes in the network (**Faisal** and **committee**) and the two links, one OBS from **Faisal** to the **committee** and one INR between the two entities. The second example in Table 2.1 is an excerpt from the novel *Emma* by Jane Austin. In this example, **Mr. Woodhouse** is talking about **Emma**, triggered by the word *rejoined*. By virtue of talking about **Emma**, **Mr. Woodhouse** is cognitively aware of **Emma** but there

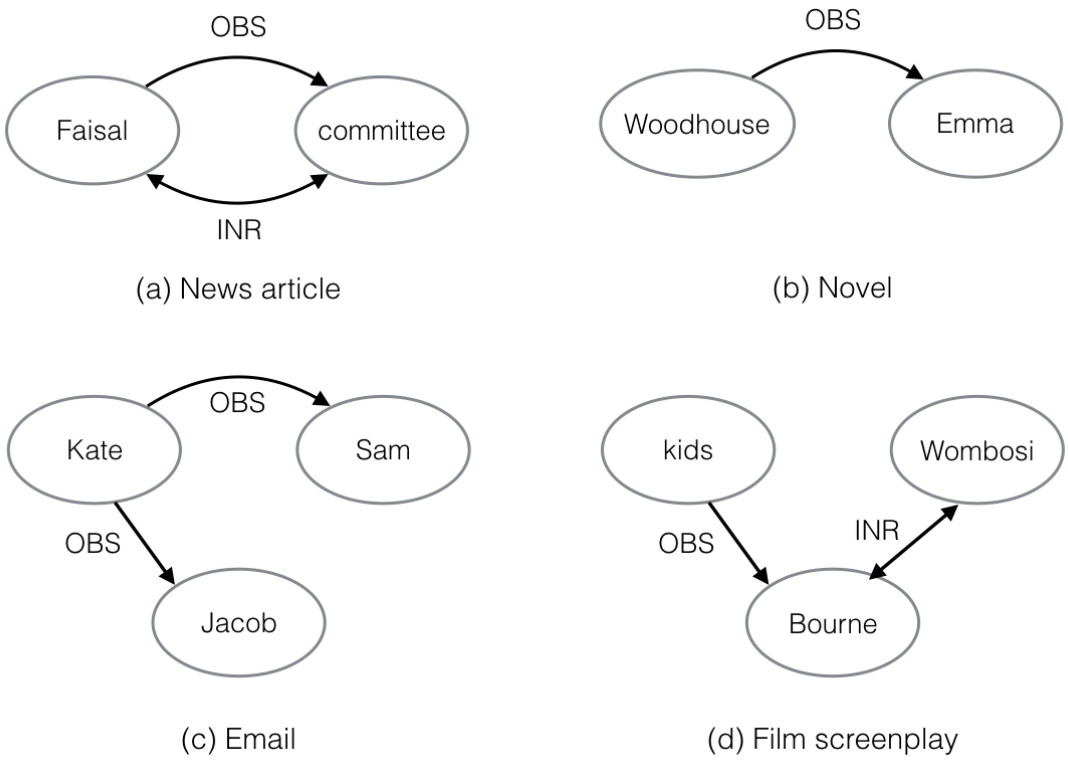


Figure 2.2: Social networks as a result of extracting social entities and social events from example sentences in Table 2.1

is no evidence that **Emma** is also aware of **Mr. Woodhouse**. Therefore, there is an OBS link directed from **Mr. Woodhouse** to **Emma** (see Figure 2.2 (b)). The third example in Table 2.1 is an email excerpt from the Enron email corpus. **Kate** sends an email to **Sam**. At the time of composing and sending this email, only **Kate** is cognitively aware of **Sam**. This information is recorded in the meta-data of the email and triggers an OBS social event directed from **Kate** to **Sam**. In the content of the email, **Kate** mentions **Jacob**. By virtue of *writing about* **Jacob**, **Kate** has **Jacob** in her cognitive state. Therefore, there is an OBS event directed from **Kate** to **Jacob** (see Figure 2.2 (c)). This email alone does not provide evidence that **Sam** has read the email and is cognitively aware of **Jacob**. Therefore, there is no OBS event directed from **Sam** to **Jacob**.

The last example in Table 2.1 is an excerpt from the screenplay of the film *The Bourne Identity*. **Bourne** is being *stared at* by **the kids** and therefore **the kids** have **Bourne** in

their cognitive state. There is no evidence that **Bourne** is cognitively aware of **the kids**. Therefore, there is an OBS event from **the kids** to **Bourne**. Furthermore, **Bourne** is having a *conversation* with **Wombosi**. Both **Bourne** and **Wombosi** are mutually aware of each other and their mutual awarenesses. Therefore, there is an INR event between **Bourne** and **Wombosi** (see Figure 2.2 (d)).

Important Note: Note that an attempt to make the cognitive states of entities apparent by the author of the text is necessary. For instance, in Example 1, the author simply states a matter of fact – the **White Rabbit** *ran by* **Alice**. The author does not make the cognitive states of the entities explicit. Therefore, as per our definition, there is no social event between the **Rabbit** and **Alice**.

(1) The [White Rabbit] ran by [Alice]. NoEvent

2.3 Subcategorization of Social Events

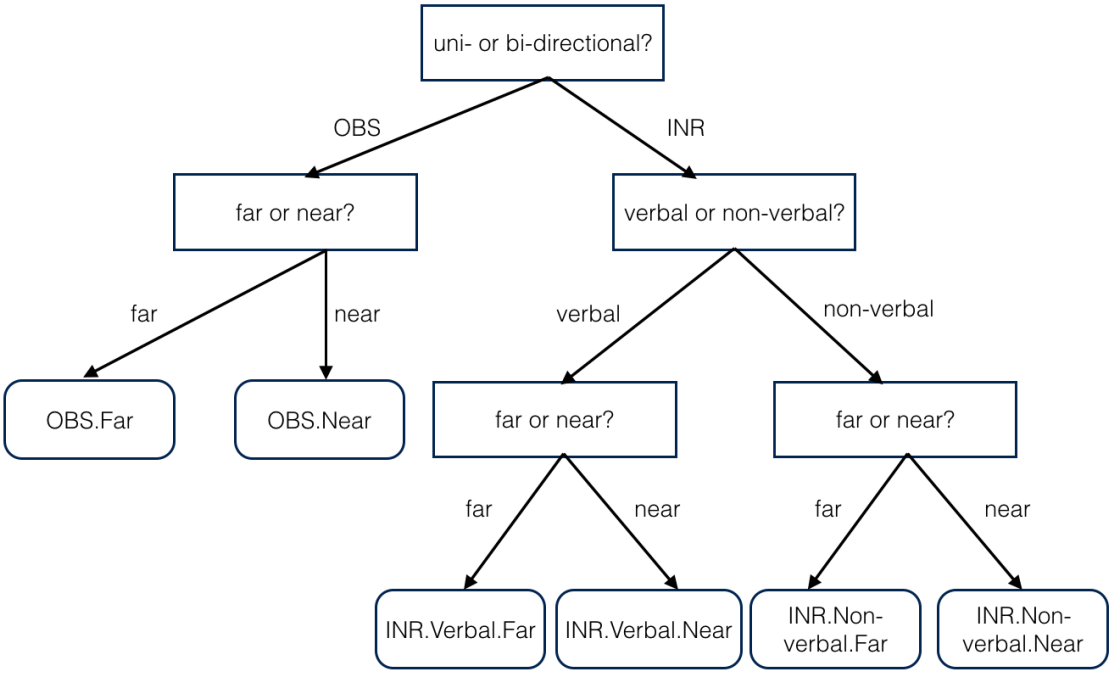


Figure 2.3: Subcategories of the two social events OBS and INR.

The two broad categories of social events (OBS and INR) have subcategories (see Figure 2.3). OBS social events may either take place in physical proximity or not. When an entity observes the other entity through a non-electronic medium, such as bare eyes, binoculars, or long range rifle, the events are OBS.NEAR. In all other cases, such as watching some on television or thinking about someone, the events are OBS.FAR.

INR social events between two entities may either be verbal or non-verbal, denoted by INR.VERBAL and INR.NON-VERBAL respectively. In a verbal interaction, entities interact primarily through words (monologue or dialogue, through email, phone or direct conversation). All other interactions, for example, waving at one another, gazing into each others eyes, are non-verbal interactions. INR.VERBAL and INR.NON-VERBAL interactions have further subcategorization. Each of these interactions may happen either in physical proximity or not. When entities interact through a medium such as electronic mails, telephone, their interaction falls in the subcategory FAR. Otherwise the interaction falls in the subcategory NEAR. The following subsections provide examples of each of these subcategories.

2.3.1 Examples for the Subcategories of OBS

(2) [Alice] {saw} the [White Rabbit] run by her. OBS.NEAR

In this example, **Alice** sees the **White Rabbit** run by her. There is no evidence that the **White Rabbit** notices **Alice**. So while there is evidence that **Alice** has the **White Rabbit** in her cognitive state, there is no evidence that the **White Rabbit** has **Alice** in its cognitive state. Therefore, this is an OBS social event directed from **Alice** to the **White Rabbit**. Furthermore, **Alice** observes the **White Rabbit** from physical proximity. Therefore the applicable subcategory of OBS is NEAR.

(3) The woman's parents; [William] and [Nancy Scott]; {found} the decomposing body of the [first baby] in her closet... OBS.NEAR

(4) Television footage showed [medical teams] {carting away} [dozens of wounded victims] with fully armed troops on guard. OBS.NEAR

In both examples 3 and 4, the observers (and only the observers) are cognitively aware of the entities being observed. In Example 3, the entity being observed, namely **first baby**,

is not alive and is therefore not capable of observing the other entity. In Example 4, there is no evidence that the entity being observed, namely **dozens of wounded victims**, are aware of the observer. After all, the victims may be unconscious. Furthermore, the entities being observed are in close physical proximity of the observers. Therefore, the social event is OBS.NEAR directed from the observers to the entities being observed.

(5) So; [we] {know} [she]'s in good spirits. OBS.FAR

In this example, a group of people (**we**) are thinking about another person (**she**). Since only the group of people have the other person in their cognitive state (there is no evidence that **she** has **we** in her cognitive state), this is an OBS social event. Furthermore, since there is no evidence that **she** is being thought about in physical proximity, the applicable subcategory of OBS is FAR.

(6) "To be sure," {said} [Harriet], in a mortified voice, "[he] is not so genteel as real gentlemen." OBS.FAR

Similar to the previous example, one entity, **Harriet**, is talking about another entity, **he**, and therefore the social event is OBS.FAR.

2.3.2 Examples for the Subcategories of INR

(7) And [one of the guys] {looked at me and said}: [Duke]; what's it like to kill. INR.VERBAL.NEAR

In this example, the two entities (**one of the guys** and **Duke**) are having a conversation or a verbal interaction. The phrase *looked at me* makes clear that the interaction is happening in physical proximity, and thus the social event is INR.VERBAL.NEAR. Note that whenever the physical proximity relation is unclear from the context, the default subcategory is FAR.

(8) [Jones] {met} with [Defense Minister Paulo Portas] on Tuesday. INR.VERBAL.NEAR

We assume that the primary mode of interaction in a meeting, unless otherwise explicitly specified, is verbal. We also assume that meetings, unless otherwise explicitly specified, happen in physical proximity. Under these assumptions, the social event in the above example is INR.VERBAL.NEAR.

(9) [The Russian Prime Minister] {had a conversation} with the [Turkish Prime Minister] on phone last evening. INR.VERBAL.FAR

In this example, the two entities (**The Russian Prime Minister** and **Turkish Prime Minister**) have a verbal interaction over an electronic medium and not in physical proximity. Therefore, the social event is INR.VERBAL.FAR.

(10) [The Army's 3rd Infantry] has punched through Karbala; {meeting only light resistance} from the [Medina Republican Guard]; INR.NON-VERBAL.NEAR

In this example, the two entities (**The Army's 3rd Infantry** and **Medina Republican Guard**) are mutually aware of each other and of the interaction, which is primarily non-verbal. The context makes clear that interaction happens in physical proximity and therefore the social event is INR.NON-VERBAL.NEAR. The following example has a similar reasoning for being a INR.NON-VERBAL.NEAR social event.

(11) [The Marines from the 1st Division] have secured a key Tigris River crossing near Al Kut and reported to have essentially {destroyed} the combat fighting ability of that [light infantry Baghdad division that was supposed to be providing defense down there]. INR.NON-VERBAL.NEAR

(12) [John] and [Mary] endlessly {gazed into each others' eyes} over Skype. INR.NON-VERBAL.FAR

In this example, the two entities (**John** and **Mary**) are mutually aware of one another through a non-verbal interaction. Since the two entities are engaged in an interaction over Skype, a popular video conferencing platform, their interaction is not in physical proximity. Therefore, the social event is INR.NON-VERBAL.FAR.

2.4 Social Events Considered in this Thesis

In this thesis, we automate the extraction of only the two broad categories of social events, namely OBS and INR. The primary reason for this choice was the unavailability of training

data for many of the fine grained categories. For example, we found only two instances of the OBS.NEAR social event compared to 110 instances of the OBS.FAR social event and only 17 instances of the INR.NON-VERBAL social event compared to 83 instances of the INR.VERBAL social event in 62 news articles (see Table 2.2).

The two most notable and related notions of social networks in the computational linguistics literature are due to Doddington *et al.* [2004] and Elson *et al.* [2010]. In the following sections, we discuss these notions in turn and differentiate them from the notion of social events.

2.5 Comparison of Social Events with ACE Relations and Events

The objective of the Automatic Content Extraction (ACE) program, as described by Doddington *et al.* [2004], has been to develop technology to automatically identify *entity* mentions, the *relations* between these entities, and the *events* in which these entities participate. A technology that is able to identify entities, their mentions, relations between entities, and the events in which entities participate can be used to create a network of entities and their connections. The definition of ACE entity and entity mention is presented in Section 2.1. In this section, we present the definition of relations and events as defined in the ACE guidelines. We then study the differences between ACE relations, ACE events, and social events to present empirical evidence that our notion of social events is substantially different from the notion of ACE relations and events.

2.5.1 Social Events versus ACE Relations

ACE Relation annotation guidelines[2] define six types of relations: PHYSICAL, PART-WHOLE, PERSONAL-SOCIAL, ORGANIZATION-AFFILIATION, AGENT-ARTIFACT, and GEN-AFFILIATION. Out of these, only the PERSONAL-SOCIAL relation describes a relation between people. All other relations describe a relation between entities of type other than PERSON and are

[2]https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-relations-guidelines-v6.2.pdf

hence irrelevant to the discussion. ACE guidelines define three types of PERSONAL-SOCIAL relations: BUSINESS, FAMILY, and LASTING-PERSONAL. According to the guidelines,

> The BUSINESS Relation captures the connection between two entities in any professional relationship. This includes boss-employee, lawyer-client, student-teacher, co-workers, political relations on a personal level, etc. This does not include relationships implied from interaction between two entities (e.g. "President Clinton met with Yasser Arafat last week").

Examples of the BUSINESS relation include *their colleagues*, *his lawyer*, and *a spokesperson for the senator*. None of these are social events. In fact, the definition explicitly mentions that BUSINESS relations do not include the interactions between entities. This condition of non-inclusion of interactions also holds for the other two types of ACE PERSONAL-SOCIAL relations, namely FAMILY (*his wife*, *his ailing father*) and LASTING-PERSONAL (*your priest*, *her neighbor*). Because of this condition, ACE relations are fundamentally different from social events, which are mainly about interactions between people. We now turn to ACE Events.

2.5.2 Social Events versus ACE Events

ACE Event annotation guidelines[3] define an event as follows:

> An Event is a specific occurrence involving participants. An Event is something that happens. An Event can frequently be described as a change of state.

ACE guidelines define eight types of events: LIFE, MOVEMENT, TRANSACTION, BUSINESS, CONFLICT, CONTACT, PERSONNEL, and JUSTICE. The following paragraphs provide a definition for each of these events and differentiate them from our definition of social events.

[3]https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

2.5.2.1 ACE Event Life

The LIFE event has five sub-types: BE-BORN, MARRY, DIVORCE, INJURE, DIE. "A BE-BORN Event occurs whenever a Person Entity is given birth to. BE-BORN Events have one participant slot (PERSON-ARG)." This event does not involve an interaction between two entities. For example, the sentence *Jane Doe was born in Casper, Wyoming on March 18, 1964* contains a LIFE.BE-BORN event that is triggered by the word *born* with **Jane Doe** as the only entity participating in the event. The BE-BORN event is therefore unrelated to our definition of social events; social events require at least two participants.

MARRY and DIVORCE events are official events, where two people are married and divorced, respectively, under the legal definition. Both MARRY and DIVORCE events have one participant slot (PERSON-ARG). This participant slot may contain one or more entities of type PERSON. For example, in the sentence, *He'd been married before and had a child*, there is only one entity in the participant slot, namely **He**. In the sentence, *Jane Doe and John Smith were married on June 9, 1998*, there are two participant entities, namely **Jane Doe** and **John Smith**. Whenever the MARRY and DIVORCE events have two or more participant entities, these events are social events of type INR. This is because when two entities marry or divorce one another, they are mutually aware of each other and of each others' awarenesses (an interaction). Because getting married and getting divorced are only two specific types of interactions, these events are a proper subset of the INR social event; social events consist of a much larger class of interactions.

Lastly, INJURE and DIE events occur whenever an entity of type PERSON experiences physical harm and death respectively. Both Injure and Die events have three participant slots: AGENT-ARG (can be a person, an organization, or a geo-political entity), VICTIM-ARG (can only be a person), and INSTRUMENT-ARG (the device used to inflict harm or kill). For an agent to cause harm to a victim, at least the agent needs to be cognitively aware of the victim. Those situations in which the agent is a person and only the agent is aware of the victim are situations that meet the criteria of an OBS social event. Situations in which the agent is a person and both the agent and the victim are mutually aware of one another meet the criteria of an INR social event. While some of the ACE INJURE and DIE events may be social events, not all social events are INJURE and DIE events.

2.5.2.2 ACE Event Movement

The second ACE event, MOVEMENT, has only one subtype: TRANSPORT. The ACE guidelines define a TRANSPORT event as an event that "occurs whenever an ARTIFACT (WEAPON or VEHICLE) or a PERSON is moved from one PLACE (GPE, FACILITY, LOCATION) to another." Since this event does not involve the participation of two entities of type PERSON, the MOVEMENT event is unrelated to social events.

2.5.2.3 ACE Event Transaction

The third ACE event, TRANSACTION, has two subtypes: TRANSFER-OWNERSHIP and TRANSFER-MONEY. Each of these events refer to the giving or receiving of artifacts and money respectively. The TRANSFER-OWNERSHIP events have five participant slots (BUYER-ARG, SELLER-ARG, BENEFICIARY-ARG, ARTIFACT-ARG, and PRICE-ARG). The buyer, the seller, and the beneficiary can be people. The TRANSFER-MONEY events have four participant slots (GIVER-ARG, RECIPIENT-ARG, BENEFICIARY-ARG, and MONEY-ARG). The giver, the recipient, and the beneficiary can be people. Since giving or receiving artifacts or money entails cognitive awareness of participants towards one another, the TRANSACTION ACE events meet the criteria for being social events. Of course, not all social events are ACE TRANSACTION events.

2.5.2.4 ACE Event Business

The fourth ACE event, BUSINESS, has four subtypes: START-ORG, MERGE-ORG, DECLARE-BANKRUPTCY, and END-ORG. None of these events are between two people. For example, START-ORG events have two participant slots (AGENT-ARG and ORG-ARG). The agent can be a person but the second participant can only be an organization. Since this event does not involve the participation of two entities of type PERSON, the BUSINESS event is unrelated to social events.

2.5.2.5 ACE Event Conflict

The fifth ACE event, CONFLICT, has two subtypes: ATTACK and DEMONSTRATE. According to the ACE guidelines, "An ATTACK Event is defined as a violent physical act causing

harm or damage. ATTACK events have three participant slots (ATTACKER-ARG, TARGET-ARG and INSTRUMENT-ARG)." Both the attacker and the target can be of type person. When one entity attacks the other, at least the attacker needs to be cognitively aware of the target. Therefore, situations in which both the attacker and the target are people, meet the criteria of being social events. In contrast, the DEMONSTRATE event has only one participant slot (ENTITY-ARG). Since this event does not involve the participation of two entities of type PERSON, the DEMONSTRATE event is unrelated to social events.

2.5.2.6 ACE Event Contact

The sixth ACE event, CONTACT, has two subtypes: MEET and PHONE-WRITE. According to the ACE guidelines,

> A MEET Event occurs whenever two or more Entities come together at a single location and interact with one another face-to-face. MEET Events include talks, summits, conferences, meetings, visits, and any other Event where two or more parties get together at some location. A PHONE-WRITE Event occurs when two or more people directly engage in discussion which does not take place face-to-face. To make this Event less open-ended, we limit it to written or telephone communication where at least two parties are specified. Communication that takes place in person should be considered a MEET Event. The very common PERSON *told reporters* is not a taggable Event, nor is *issued a statement.* A PHONE-WRITE Event must be explicit phone or written communication between two or more parties.

As the definition suggests, all MEET and PHONE-WRITE events are INR social events. However, the category of INR social events is larger; there are other types of interactions that fall in the INR category. Following are two example sentences that have INR social events but not ACE CONTACT events.

(13) Yesterday a silent [Dee Ana Laney] waited as the [judge] {read the charges} against [her] INR.VERBAL.NEAR

(14) [The Army's 3rd Infantry] has punched through Karbala; {meeting only light resistance} from the [Medina Republican Guard]; INR.NON-VERBAL.NEAR

2.5.2.7 ACE Event Personell

The seventh ACE event, PERSONNEL, has four subtypes: START-POSITION, END-POSITION, NOMINATE, and ELECT. "A START-POSITION Event occurs whenever a PERSON Entity begins working for (or changes offices within) an ORGANIZATION or GPE." For example, in the sentence, *Mary Smith joined Foo Corp. as CEO in June 1998*, the entity, **Mary Smith**, begins working at an organization, **Foo Corp.** A similar definition and example applies for the END-POSITION events. Since neither START-POSITION nor END-POSITION involve an interaction between people, these events are unrelated to social events.

"A NOMINATE Event occurs whenever a PERSON is proposed for a START-POSITION Event by the appropriate PERSON, through official channels. NOMINATE Events have two participant slots (PERSON-ARG and AGENT-ARG)." The AGENT-ARG can be an entity of type PERSON. A similar definition and participant slots also hold for the ACE event ELECT. For one person to nominate or elect another person, at least that one person needs to be cognitively aware of the other person. Therefore, both NOMINATE and ELECT events can be social events. Of course, not all social events are of type NOMINATE and ELECT.

2.5.2.8 ACE Event Justice

The eighth and last ACE event is JUSTICE. The JUSTICE event has 13 subtypes: ARREST-JAIL, RELEASE-PAROLE, TRIAL-HEARING, CHARGE-INDICT, SUE, CONVICT, SENTENCE, FINE, EXECUTE, EXTRADITE, ACQUIT, APPEAL, and PARDON. All of these social events can be between two or more people where the parties can be cognitively aware of one another. Therefore, all these events can be social events. However, not all social events are of type JUSTICE.

2.5.2.9 Quantitative Evaluation of the Differences between ACE Events and Social Events

As discussed above, several ACE events can be social events. However, not all social events are covered by the ACE event categories. In this section, we provide a quantitative evaluation to show that a large majority of social events are not covered by any of the AEC events.

We perform the evaluation on 62 news articles taken from the ACE-2005 corpus.[4] We refer to this collection as **ACE-62**. These news articles contain ACE entity, entity mention, and event annotations. We annotate these news articles with social events and report the degree of overlap between ACE events and social events. We say that an ACE event *matches* a social event if both the following conditions hold:

1. The span of text that triggers an ACE event overlaps with the span of text that triggers a social event.

2. The entities that participate in an ACE event are the same as the entities that participate in a social event.

Table 2.2 presents the intersection between ACE events and social events. The rows represent social events and the columns represent ACE events. Each event has an integer in parentheses. For example INR.VERBAL.NEAR has the integer 66. This integer represents the number of times a particular event occurs in the ACE-62 corpus. As another example, the table shows that the CONTACT ACE event appears 32 times in the ACE-62 corpus.

Each cell in the table shows the number of social events that are covered by a particular ACE event. For example, the cell INR.VERBAL.NEAR and CONTACT.MEET contains the value 26. This means that 26 out of 66 social events of type INR.VERBAL.NEAR are annotated as CONTACT.MEET in the ACE-62 corpus.

The last column contains the count of social events that are not covered by any of the ACE events. Continuing with the INR.VERBAL.NEAR example, the last column for this row contains a value 31. This means that 31 out of 66 INR.VERBAL.NEAR events are not covered by any of the ACE events.

[4] https://catalog.ldc.upenn.edu/LDC2006T06

62 News Articles			Life (7)		Trans action (2)	Conflict (5)	Contact (32)		Justice (13)	Not Found
			Marry/ Divorce	Injure/ Die			Meet	Phone -Write		
INR	Verbal	Near (66)	0	0	0	0	26	0	9	31
		Far (17)	1	0	0	0	0	3	3	10
	Non-Verbal	Near (14)	0	2	1	3	0	0	0	8
		Far (3)	0	0	1	0	0	0	0	2
OBS	Near (2)		0	2	0	0	0	0	0	0
	Far (110)		0	1	0	2	0	0	0	107
# of incorrect annotations			0	1	0	0	3	0	1	

Table 2.2: This table maps the type and subtype of ACE events to our types and subtypes of social events. The columns have ACE event types and sub-types. The rows represent our social event types and sub-types. The last column is the number of our events that are not annotated as ACE events. The last row has the number of social events that our annotator missed but are ACE events.

Overall, the table shows that the ACE-62 corpus contains 212 (66 + 17 + 14 + 3 + 2 + 110) social events. Out of these only 54 (212 - (31 + 10 + 8 + 2 + 0 + 107)) match an ACE event. This means that only 25.5% of the social events are covered by the already annotated ACE events. Furthermore, neither INR nor OBS social events are subsumed by any of the ACE events. This necessitates the annotation of all social events. We conclude that the notion of social events is significantly different from the notion of ACE events and that a separate annotation effort for annotating social events is required.

We now present examples for each of the social event categories that are not covered by the ACE events. The goal of these examples is to give the reader an intuition behind the conceptual differences between social events and ACE events. None of the following sentences are annotated with ACE events. However, these sentences contain social events.

(15) Amid a chill in relations over the war in Iraq, which Canada opposed, [Bush] indefinitely postponed a visit to Canada, instead choosing to {host} [Australian Prime Minster John Howard], who endorsed that military campaign. INR.VERBAL.NEAR

In the above sentence, since **Bush** is hosting **Australian Prime Minster John Howard**, there is evidence that both entities are mutually aware of each other. It is reasonable to assume that the primary mode of interaction is verbal. Furthermore, hosting someone implies physical proximity. Therefore, according to our guidelines, there is a social event of type INR.VERBAL.NEAR.

(16) [The judges] also {rejected an application by} [Anwar] to be released on bail. INR.VERBAL.FAR

In the above sentence, there is evidence that both the entities, **The judges** and **Anwar** are mutually aware of one another, the primary mode of interaction is verbal, and there is no evidence of physical proximity. Therefore, according to our guidelines, there is a social event of type INR.VERBAL.FAR.

(17) [A team of specialists] here {have been conducting tests} on [the female twins, Laleh and Ladan Bijani], since last year to determine if the operation can be successful. INR.NON-VERBAL.NEAR

In the above sentence, an entity, **A team of specialists**, is performing tests on another entity, **the female twins**. There is evidence that both entities are mutually aware of one another, the interaction is primarily non-verbal (*conducting tests*), and is in physical proximity. Therefore, according to our guidelines, there is a social event of type INR.NON-VERBAL.NEAR.

(18) [The writer] will retain the rights to his books and films, although he has {agreed to split a raft of other possessions with} [Anne Marie], his wife of 13 years, according to documents filed in Los Angeles Superior Court. INR.NON-VERBAL.FAR

In the above sentence, an entity, **The writer**, is performing tests on another entity, **Anne Marie**. There is evidence that both entities are mutually aware of one another, the interaction is primarily non-verbal (*split a raft of possessions*), and there is no evidence that the interaction is in physical proximity. Therefore, according to our guidelines, there is a social event of type INR.NON-VERBAL.FAR.

(19) [We]'ve been {waiting all day word from} [the doctors]. OBS.FAR

In the above sentence, an entity, **We**, has been waiting to hear from another entity, **the doctors**. While there is evidence that the entity **We** is cognitively aware of the other entity **the doctors**, there is no evidence that even **the doctors** are cognitively aware of **We**. Therefore, the relevant social event is OBS. Furthermore, that is no evidence of physical proximity, so the relevant subcategory of the OBS event is FAR.

2.6 Comparison of Social Events with Conversations

Elson *et al.* [2010] introduce the notion of conversational networks. The authors extract these networks from nineteenth century British novels. The nodes in a network are characters and links are *conversations*. They define a conversation as:

> A continuous span of narrative time featuring a set of characters in which all of the following conditions are met: 1) The characters are either in the same place at the same time, or communicating by means of technology such as a telephone.

> 2) The characters take turns speaking. 3) The characters are mutually aware of each other and their dialogue is mutually intended for the other to hear. 4) Each character hears and understands the other's speech. A person present in a group is not counted in the conversation unless he or she speaks. Conversations that are related solely through a character's narration (i.e., stories told by characters) do not count.

Consider the following excerpt from the novel *Emma* by Jane Austin:

> "Especially when one of those two is such a fanciful, troublesome creature!" said **Emma** playfully. "That is what you have in your head, I know – and what you would certainly say if my father were not by."
>
> "I believe it is very true, my dear, indeed," said **Mr. Woodhouse**, with a sigh. "I am afraid I am sometimes very fanciful and troublesome."

In this excerpt, two entities, **Emma** and **Mr. Woodhouse**, are having a conversation (as defined by the four conditions above). Elson *et al.* [2010] extract a network of two nodes (**Emma** and **Mr. Woodhouse**) and one conversational link.

Definitionally, all conversations are INR social events. However, not all INR social events are conversations (as explicated below). Furthermore, OBS social events are not conversations (because conversations require mutual awarenesses of the characters). We conclude that networks in which links are social events are significantly different from conversational networks. In fact, the set of links in a conversational network is a proper subset of the set of links in our definition of a social network. The following examples provide further justification for our conclusion above.

(20) [Mr. Micawber] $\{\text{said}\}_{OBS}$, that [he] had $\{\text{gone home with}\}_{INR}$ [Uriah] **OBS** and **INR**

In the above example, **Mr. Micawber** is talking about going home with **Uriah**. Since **Mr. Micawber** is talking about **Uriah**, there is a directed OBS link from **Mr. Micawber**

to *Uriah*. In what **Mr. Micawber** is saying, there is an INR link between **Mr. Micawber** and *Uriah*. This is because the two entities went home together and going home together is evidence that both were aware of each other and of their mutual awarenesses. However, this example does not fit the definition of a conversation.

(21) [Mr. Elton] {was speaking with animation, [Harriet] listening with a very pleased attention}; and [Emma], having sent the child on, was beginning to think how she {might draw back a little more, when they both looked around, and she was obliged to join them}. **INR**

In the above example, there is evidence that all three entities, **Mr. Elton**, **Harriet**, and **Emma**, are mutually aware of one another and of their mutual awarenesses. However, this example does not fit the definition of a conversation because the characters do not take turns speaking.

(22) "[Emma] never thinks of herself, if she can do good to others," {rejoined} [Mr. Woodhouse] **OBS**

In the above example, **Mr. Woodhouse** is talking about **Emma**. He is therefore cognitively aware of Emma. However, there is no evidence that Emma is also aware of **Mr. Woodhouse**. Since only one character is aware of the other, this is an OBS event directed from **Mr. Woodhouse** to **Emma**. This example does not fit the definition of a conversation because conversations require that the characters are mutually aware of each other.

(23) [Elton]'s manners are superior to [Mr. Knightley]'s or [Mr. Weston]'s. **NoEvent**

In the above example, the author (Jane Austen) is stating a fact about three characters (**Elton**, **Mr. Knightley**, and **Mr. Weston**). However, the author provides no insight into the cognitive states of the characters, and thus there is no social event between the characters. There is also no conversation.

2.7 Conclusion

In this chapter we introduced our working definition of social networks for the rest of the thesis. We defined social networks to be networks in which nodes are entities and links are social events. We borrowed the definition of an entity and entity mention from the ACE guidelines and restricted our notion of an entity to be of type person (individual or group). We introduced the notion of social events and differentiated it from other notions of interactions, specifically from ACE events and the one defined by Elson *et al.* [2010]. Definitionally, social events have two broad categories and several sub-categories. But due to the lack of presence of sub-categories in the data-set that we annotated, we work with only the two broad categories of social events, namely observation (OBS) and interaction (INR).

Part II

Extracting Social Networks from Unstructured Text

Chapter 2 of this thesis introduced the definition of a new kind of social network – a network in which nodes are entities (people or groups of people) and links are social events. This part of this thesis introduces a novel machine learning approach for automatically extracting these social networks from unstructured texts. Unstructured texts such as novels often tell stories about people (real or fictional). These stories revolve around people and social events between these people. Social events aggregate over the course of the story to form a social network. In order to automatically extract social networks – the aggregation of social events – we build models to detect and classify social events expressed using language. For example, given the sentence, *John and Mary are having dinner together*, we want our models to detect a social event between the two entities and classify the social event as an interaction event. We use these models for extracting social networks from nineteenth century British literature and study some of the long standing literary theories that comment on the structure of social networks in novels.

This part of the thesis is organized as follows: Chapter 3 introduces the task definition along with literature survey on relation extraction, Chapter 4 provides details about the data, our annotation effort, our machine learning approach, and experiments, Chapter 5 presents an application of automatic social network extraction for validating literary theories.

Chapter 3

Introduction

In this chapter we provide a formal definition for our tasks. Since our task definitions are closely related to the well studied task of relation extraction, we provide literature survey of the techniques developed for relation extraction. When applicable, we use the technique proposed for relation extraction as a baseline for our tasks.

3.1 Task Definition

There are two broad categories of social events: Observation (OBS) and Interaction (INR). Since OBS is a directed social event (directed from the entity that is observing the other entity to the entity that is being observed), we define two classes for the OBS class: $\overrightarrow{OBS}$ and $\underleftarrow{OBS}$. $\overrightarrow{OBS}$ stands for the OBS social event that is directed from the first entity (first in terms of surface word order) to the second entity. $\underleftarrow{OBS}$ stands for the OBS social event that is directed from the second entity to the first entity. In order to extract social networks from unstructured texts, we build machine learning models for three classification tasks:

- Social Event Detection: detecting whether or not there is a social event between a pair of entities in a sentence.
- Social Event Classification: for the pair of entities that have a social event, classifying the event into one of {INR, OBS}.

- Directionality Classification: for the pair of entities that have an OBS social event, classifying the directionality of the event ($\overrightarrow{OBS}$ or $\overleftarrow{OBS}$).
- Social Network Extraction: combining the models developed for the aforementioned three tasks into one model.

These tasks are closely related to a well studied task in the information extraction and computational linguistics literature: relation extraction. We review the vast amount of literature regarding relation extraction in the next section.

3.2 Related Work on Relation Extraction

Relation extraction, the task of finding relations between entities, started as a series of Message Understanding Conferences (MUC) in 1987. We refer the reader to Grishman and Sundheim [1996] for an excellent review and historical account of the MUC conferences. Grishman and Sundheim [1996] note:

> MUC-1 (1987) was basically exploratory; each group designed its own format for recording the information in the document, and there was no formal evaluation. By MUC-2 (1989), the task had crystalized as one of template filling. One receives a description of a class of events to be identified in the text; for each of these events one must fill a template with information about the event. The template has slots for information about the event, such as the type of event, the agent, the time and place, the effect, etc. For MUC-2, the template had 10 slots. Both MUC-1 and MUC- 2 involved sanitized forms of military messages about naval sightings and engagements.

Since its inception, researchers have proposed several approaches for the task. Figure 3.1 presents one way of characterizing the vast literature. Table 3.1 provides citations for each of the categories displayed in Figure 3.1. We discuss the related work in each of these categories in turn.

3.2.1 Bootstrapping

Bootstrapping is a self-sustaining process that starts with an initial set of examples, called the seed set, and spools through the data to gather *similar* examples, then use the knowledge obtained from these gathered examples to increase the coverage of the initial seed set, and continue spooling until a stopping criteria is met. One of the earliest bootstrapping systems for information extraction was introduced by Brin [1999]. Brin's system, called DIPRE (Dual Iterative Pattern Relation Extraction), extracts (author, title) pairs from the world wide web (WWW). The input to the system is a small set of seeds (e.g. (*Arthur Conan Doyle*, *The Adventures of Sherlock Holmes*)) and the output is a long list of (author, title) pairs. The system spools through the WWW taking note of webpages that mention both elements of any seed pair. During the spooling process, the system creates a number of six-tuples: $[order, author, book, prefix, suffix, middle]$. These tuples are then used as wild card expressions (e.g. $[author, .*?, prefix, .*?, middle]$) to collect more instances of (author, book) pairs. The process finally halts when a stopping criteria is met (such as the number of passes over the data exceeds a predefined number). Agichtein and Gravano [2000] build on the architecture of DIPRE and incorporate several extensions into a system called SNOWBALL. SNOWBALL employs novel strategies for representing and evaluating patterns

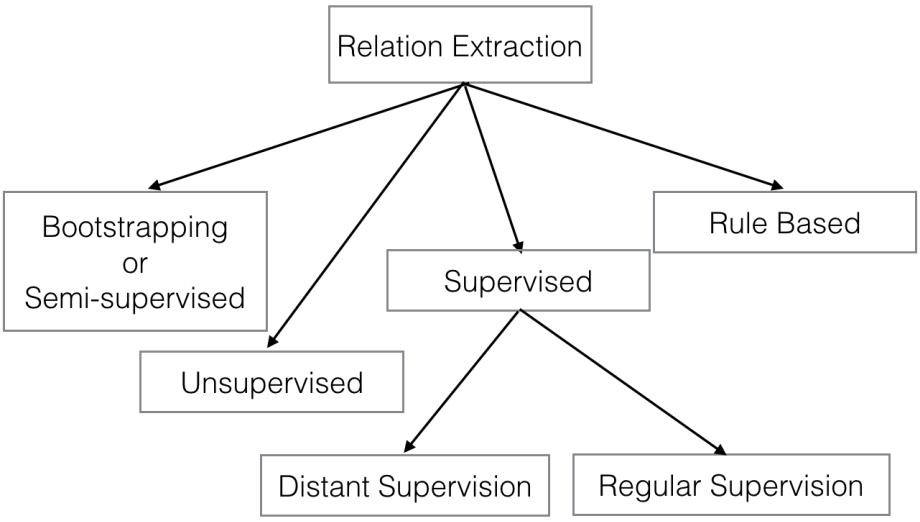


Figure 3.1: Categorization of related work on relation extraction.

Category	Citations
Bootstrapping	[Brin, 1999; Riloff *et al.*, 1999; Agichtein and Gravano, 2000]
Unsupervised	[Hasegawa *et al.*, 2004; Etzioni *et al.*, 2005; Paşca *et al.*, 2006; Sekine, 2006; Shinyama and Sekine, 2006; Banko *et al.*, 2007; Hoffmann *et al.*, 2010; Wu and Weld, 2010]
Distant Supervision	[Wu and Weld, 2007; Bunescu and Mooney, 2007; Mintz *et al.*, 2009; Riedel *et al.*, 2010; Hoffmann *et al.*, 2011; Surdeanu *et al.*, 2011; Nguyen and Moschitti, 2011; Wang *et al.*, 2011; Min *et al.*, 2013; Riedel *et al.*, 2013; Fan *et al.*, 2014]
Regular Supervision	Feature based approaches: [Kambhatla, 2004; GuoDong *et al.*, 2005; Boschee *et al.*, 2005; Jiang and Zhai, 2007; Chan and Roth, 2010; Sun *et al.*, 2011; Chan and Roth, 2011]. Convolution kernel based approaches: [Zelenko *et al.*, 2002; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; Mooney and Bunescu, 2005; Zhao and Grishman, 2005; Zhang *et al.*, 2006; Harabagiu *et al.*, 2005; Zhang *et al.*, 2008; Nguyen *et al.*, 2009; Zhou *et al.*, 2010; Plank and Moschitti, 2013]
Rule based	Co-occurrence based: [Ding *et al.*, 2002; Jelier *et al.*, 2005; Jenssen *et al.*, 2001]. Linguistic rule based: [Fundel *et al.*, 2007]

Table 3.1: Citations for work on relation extraction.

and tuples. These strategies enable the system to gather a larger and more accurate set of tuples.

None of these techniques may be used for social event detection or classification because of the following reasons. First, unlike author book pairs, where a book has only one author, a person might be related to multiple people with social events. Second, these systems are designed to extract universally known facts about universally known entities. For example,

Arthur Conan Doyle is the author of The Adventures of Sherlock Holmes and *Microsoft is headquartered in Seattle.* Arthur Conan Doyle, The Adventures of Sherlock Holmes, Microsoft, and Seattle are universally known entities. We want to build a system that is able to extract social events even between entities that may not be universally known. For example, between entities in an organizational email corpus. Furthermore, these techniques require a web scale corpus that offers linguistic redundancy. We want to be able to extract social networks even from a small corpus such as a short novel.

3.2.2 Unsupervised

Several researchers point out a few major limitations with DIPRE and SNOWBALL [Hasegawa *et al.*, 2004; Paşca *et al.*, 2006; Sekine, 2006; Shinyama and Sekine, 2006; Banko *et al.*, 2007]. First, a small amount of human labor is still required to define the initial set of seeds. Second, extracting tuples for a new relation requires creation of a new seed set along with re-running the systems. Third, and the biggest, the systems only extract a pre-defined list of relations between pre-defined types of entities. Hasegawa *et al.* [2004] deal with all of the aforementioned limitations by *clustering* pairs of named entity mentions based on their contexts of appearance. Their approach does not require a manually created seed set and is able to discover *all* relations in one run of the system. Etzioni *et al.* [2005] deal with these limitations differently. They introduce a system called KNOWITALL that uses the WWW as an oracle for evaluating the plausibility of automatically generated candidate facts about entities.

Banko *et al.* [2007] note that scale and speed are two major limitations of previous relation agnostic work [Hasegawa *et al.*, 2004; Sekine, 2006; Shinyama and Sekine, 2006]. They introduce an open information extraction (OIE) system called TEXTRUNNER. TEXTRUNNER, unlike Shinyama and Sekine [2006], does not require "heavy" linguistic operations such as deep linguistic parsing, named entity recognition, and co-reference resolution. Furthermore, TEXTRUNNER does not require clustering of documents thus pushing the complexity down from $O(D^2)$ to $O(D)$ (where D is the number of documents). Banko *et al.* [2007] state the advantages of TEXTRUNNER over KNOWITALL as follows:

> However, KNOWITALL requires large numbers of search engine queries and Web

> page downloads. As a result, experiments using KnowItAll can take weeks to complete. Finally, KnowItAll takes relation names as input. Thus, the extraction process has to be run, and re-run, each time a relation of interest is identified. The OIE [Open Information Extraction] paradigm retains KnowItAll's benefits but eliminates its inefficiencies.

TextRunner consists of three main components: (1) Self-supervised learner, (2) Single-pass extractor, and (3) Redundancy-based assessor. The self-supervised learner uses dependency parse of sentences along with a handful of heuristics to automatically label a seed set of examples as "trustworthy" or "untrustworthy". For example, sentences are labeled trustworthy "if there exists a dependency chain between e_i and e_j that is no longer than a certain length" (e_i and e_j are the two entities). Sentences are also labeled trustworthy if "the path from e_i to e_j along the syntax tree does not cross a sentence-like boundary (e.g. relative clauses)." The self-labeled examples are used to train a Naive Bayes (NB) classifier with a simple set of linguistic features that can be extracted quickly. The Single-pass extractor makes a single pass over the corpus and using the self-trained NB model labels each extracted tuple as trustworthy or untrustworthy. The trustworthy tuples are stored while the others are discarded. "The Redundancy-based assessor assigns a probability to each retained tuple based on a probabilistic model of redundancy in text introduced in [Downey *et al.*, 2006]."

Much like the bootstrapping, none of these systems may be used for social event detection and classification. These systems are designed to discover all possible relations between all possible types of entities. For example, TextRunner would auto-label the following sentence as trustworthy and try to learn a relation even when there is no social event between the two entities: *Sara is older than Cherry.* However, we take motivation from the idea of looking at the dependency paths between entities to experiment with a baseline that utilizes the path information (details deferred to section Section 4.2.2).

3.2.3 Distant Supervision

Wu and Weld [2007] propose the idea of utilizing Wikipedia's info-boxes and articles for automatically creating training data for learning supervised classifiers. Mintz *et al.* [2009]

formally introduce the term *distance supervision* in the context of relation extraction. The authors build on past work by utilizing a larger resource, *Freebase*, for gathering a set of entity pairs and then using the web as a whole (not just one Wikipedia page) for acquiring relation instances between the entities. The authors note:

> The intuition of distant supervision is that any sentence that contains a pair of entities that participate in a known Freebase relation is likely to express the relation in some way.

Since then several researchers have employed this idea for automatically collecting training data coupled with different machine learning techniques for training classifiers. For example, Riedel *et al.* [2010] suggest the use of "matrix factorization models that learn latent feature vectors for entity tuples and relations." Nguyen and Moschitti [2011] use Yago [Suchanek *et al.*, 2007] for collecting training instances along with convolution kernels and SVM for training. This specific idea of using a knowledge base for automatically collecting training examples is not applicable to our tasks; these knowledge bases to do not contain relations relevant for detecting and classifying social events. However, the general idea of distant supervision is applicable. The general idea behind distant supervision is to use heuristics for automatically creating a training dataset from a large corpus. This training set may be noisy (because no human annotation is involved), but the hope is that this heuristically annotated dataset contains useful patterns for the end classification task. The general idea behind distant supervision does not require the use of a knowledge base. We use the idea of distant supervision for extracting social networks from movie screenplays (see Chapter 10 for details).

3.2.4 Feature Based Supervision

Two of the earliest and most notable works on feature based relation extraction for ACE relation types and subtypes is by Kambhatla [2004] and GuoDong *et al.* [2005]. Kambhatla [2004] introduce a wide range of features, ranging from shallow lexical level to deep syntactic level. Table 3.2 presents the full set of features introduced by Kambhatla [2004]. Out of these features, Entity Type, Overlap, and Dependency features contribute the most for the

Feature Name	Feature Description
Words	The words of both mentions and all the words in between.
Entity Type	The entity type (one of PERSON, ORGANIZATION, LOCATION, FACILITY, Geo-Political Entity or GPE) of both the mentions.
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 and part-of-speech and chunk labels of the words on which the mentions are dependent in the dependency tree derived from the syntactic parse tree.
Parse Tree	The path of non-terminals (removing duplicates) connecting the two mentions in the parse tree, and the path annotated with head words.

Table 3.2: Feature set introduced by Kambhatla 2004 for the ACE relation extraction task. This table is taken as-is from their paper.

ACE relation classification task. GuoDong *et al.* [2005] build on the work of Kambhatla [2004] and introduce two novel sets of features: (1) features derived from shallow parse of sentences or sentence chunking and (2) features derived from semantic resources. As GuoDong *et al.* [2005] report, the rationale behind incorporating shallow parse features is that ACE relations have a short span: 70% of the entities (that participate in a relation) are embedded within one noun phrase or separated by just one word. GuoDong *et al.* [2005] incorporate two semantic resources: (1) a "Country Name List" that is used to differentiate the relation subtype ROLE.Citizen-Of from other subtypes, especially ROLE.Residence, and (2) WordNet [Miller, 1995] that is used to differentiate between six personal social relation subtypes (Parent, Grandfather, Spouse, Sibling, Other-Relative, and Other-Personal).

We experiment with the features proposed by GuoDong *et al.* [2005] for our tasks (details

deferred to Section 4.2.3).

Chan and Roth [2010] build on the work of GuoDong *et al.* [2005] and introduce novel features that incorporate background or world knowledge. Given a pair of entity mentions, the authors use their *Wiki* system [Ratinov *et al.*, 2010] to find Wikipedia pages for entities. If one of the two entities is found to be mentioned in the Wikipedia page of the other entity, a binary feature is turned on. Chan and Roth [2010] introduce another feature that is derived from the Wikipedia ontology. Using their previous system, Do and Roth [2010] find if a parent-child relationship exists between the two entities in Wikipedia (if Wikipedia page of one entity points to the the Wikipedia page of the other entity, the first entity is said to be the parent of the second entity). The authors find this feature to be especially useful for the Part-of ACE relation. None of these features are applicable to our tasks; we are interested in detecting and classifying social events even between entities that may not have Wikipedia pages.

3.2.5 Convolution Kernels Based Supervision

Since their introduction for Natural Language Processing (NLP) [Collins and Duffy, 2002], convolution kernels[1] have been widely used for the ACE relation extraction task [Zelenko *et al.*, 2002; Culotta and Sorensen, 2004; Bunescu and Mooney, 2005; Mooney and Bunescu, 2005; Zhao and Grishman, 2005; Zhang *et al.*, 2006; Harabagiu *et al.*, 2005; Zhang *et al.*, 2008; Nguyen *et al.*, 2009; Zhou *et al.*, 2010; Plank and Moschitti, 2013]. Text has at least two natural representations: strings and trees. Convolution kernels for NLP applications thus fall into two broad categories: sequence kernels (for strings) and tree kernels (for trees). Over the years, researchers have introduced both, new *types of kernels* and new *types of data representations*. We discuss these in the following paragraphs.

Types of String Kernels: Lodhi *et al.* [2002] introduce two types of string kernels: Contiguous and Sparse. Table 3.3 shows the implicit feature space for each of these kernels for the string "cat". The Contiguous kernel only considers contiguous subsequences. For example, the subsequence "ct", that skips the letter "a" is excluded from the implicit feature space of the Contiguous kernel. The subsequence "ct", however, appears in the feature space

[1]For a general introduction on convolution kernels please refer to the Appendix A.

of the Sparse kernel.

Type of Kernel	Implicit Feature Space
Contiguous	ca, at, cat
Sparse	ca, at, ct, cat

Table 3.3: Types of string kernels and their implicit feature space for the string "cat". Both these kernels were introduced by Lodhi *et al.* 2002.

Data Representations for String Kernels: Zhao and Grishman [2005] introduce the following data representations for string kernels: (a) surface order of entity mention *tokens* and the tokens for intertwining words (*seq*), (2) surface order of entity mention tokens and the tokens for *important* intertwining words (*link*), and (3) the sequence of tokens on the dependency path between the entity mentions (*path*). Zhao and Grishman [2005] define different types of tokens. For instance, a token for words (not entity mentions) is defined as a three-tuple: (word, part of speech tag for the word, morphological base form of the word). A token for an entity mention has additional attributes such as entity type, entity sub-type, and entity mention type. Zhao and Grishman [2005] consider words other than "the words and constituent types in a stop list, such as time expressions" as *important* words.

Table 3.4 summarizes the work of Zhao and Grishman [2005] and other researchers along three dimensions: (1) the type of string kernel, (2) the type of string, and (3) the type of operation applied to strings. For instance, the table shows that Zhao and Grishman [2005] primarily use contiguous kernels ("Type of String Kernel" → "Contiguous" → "Zhao and Grishman [2005]"), consider sequences of words in the surface order and the dependency path between the entity mentions ("Type of String" → "Surface order" → "Zhao and Grishman [2005]" and "Type of String" → "Path in Dependency Tree (DT)" → "Zhao and Grishman [2005]"), add attributes (part of speech tag, entity type, etc.) to tokens, and *prune* the sequence of tokens by retaining only the important tokens.

Mooney and Bunescu [2005] consider surface order sequences of tokens before and between, tokens *only* between, and tokens between and after the entity mentions. The authors define a token to be one of the following: word, its POS tag, a generalized POS tag (i.e. Noun, Verb, etc. instead of their finer categories), entity and chunk types. In the same year,

Dimension	Type	References
Type of String Kernel	Contiguous	[Zhao and Grishman, 2005]
	Sparse	[Mooney and Bunescu, 2005; Nguyen *et al.*, 2009]
Type of String	Surface order	[Zhao and Grishman, 2005; Mooney and Bunescu, 2005; Nguyen *et al.*, 2009]
	Path in Phrase Structure Tree (PST)	[Nguyen *et al.*, 2009]
	Path in Dependency Tree (DT)	[Zhao and Grishman, 2005; Bunescu and Mooney, 2005; Nguyen *et al.*, 2009]
Type of String Operation	Add attributes to tokens	[Zhao and Grishman, 2005; Mooney and Bunescu, 2005; Bunescu and Mooney, 2005]
	Prune string	[Zhao and Grishman, 2005; Nguyen *et al.*, 2009]

Table 3.4: Data representations for string kernels.

Bunescu and Mooney [2005] introduce a new type of data representation, called the shortest dependency path representation. The authors represent each sentence as a dependency *graph* (not tree) and consider the sequence of tokens on the shortest undirected path from one entity to the other. However, in this representation, tokens have multiple attributes (all of the attributes mentioned above). The authors employ a simple linear kernel $K(x, y)$:

$$K(x,y) = \begin{cases} 0 & m \neq n \\ \prod_{i=1}^{n} c(x_i, y_i) & \text{m = n} \end{cases}$$

where x and y are sequences of length m and n respectively and $c(x_i, y_i)$ is the count of common attributes for the i^{th} token.

Nguyen *et al.* [2009] build on past work and introduce an array of novel sequence

kernels over the surface order and paths over novel representations of phrase structure and dependency trees. We use all the kernels and data representations introduced by Nguyen *et al.* [2009] as a baseline for social event detection and classification (details deferred to Section 4.2.4).

Types of Tree Kernels: Collins and Duffy [2002] introduce a tree kernel called the Sub-set Tree (SST) kernel. Table 3.5 shows the implicit feature space for this kernel. The sub-trees in the implicit feature space are such that entire (not partial) rule productions must be included. For example, the sub-tree [A B] is excluded because it contains only part of the production A $\rightarrow$ B C D. This kernel was designed specifically to be used for phrase structure trees, which are created using the production rules of a grammar.

Zelenko *et al.* [2002] introduce two tree kernels: the Contiguous Tree (CT) kernel and the Sparse Tree (ST) kernel. Table 3.5 shows the implicit feature space for each of these kernels. The CT kernel is more *flexible* than the Sub-set tree kernel in that it enumerates sub-trees that may violate the production rules. While the subtree [.A B] is excluded from the implicit feature space of the SST kernel, it is part of the implicit feature space of the CT kernel (see rows 1 and 2 of Table 3.5). The CT kernel is less flexible than the ST kernel. The CT kernel considers contiguous sequences of daughters while the ST kernel may *skip* over the daughters. For example, the sub-tree [.A B D] skips over the daughter C. This subtree is therefore absent from the implicit feature space of the CT kernel (see rows 2 and 3 of Table 3.5).

Data Representations for Tree Kernels: Table 3.6 attempts to summarize the vast literature on the use of convolution kernels for relation extraction. Researchers have primarily utilized three types of trees (Shallow, Phrase Structure, and Dependency) for deriving tree based data representations. Common operations on trees may be categorized into three broad categories: (a) addition of attributes such as part of speech tags, grammatical roles, to nodes in the tree, (b) instead of adding attributes to nodes, creation of nodes with attribute information and addition of these nodes to the tree, and (c) tree pruning. Additionally, researchers have explored the use of external resources such as WordNet, Brown Clusters, and Latent Semantic Analysis (LSA), PropBank, and FrameNet to deal with issues concerning feature sparsity and for incorporating semantic information.

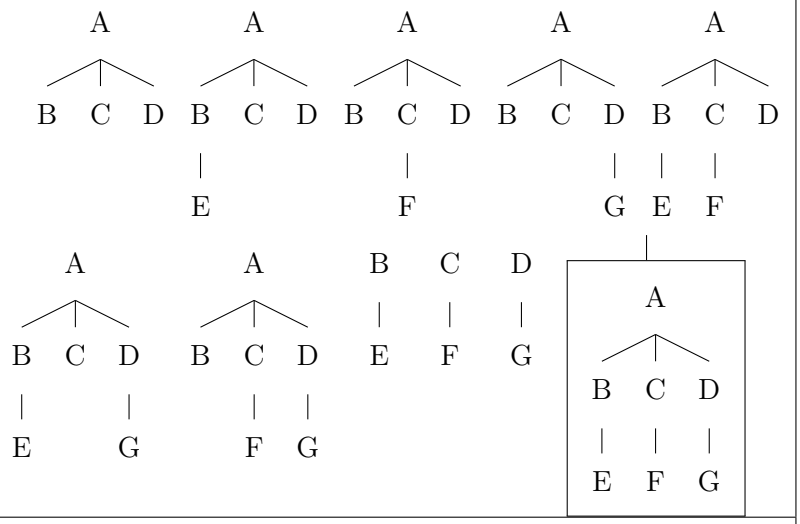
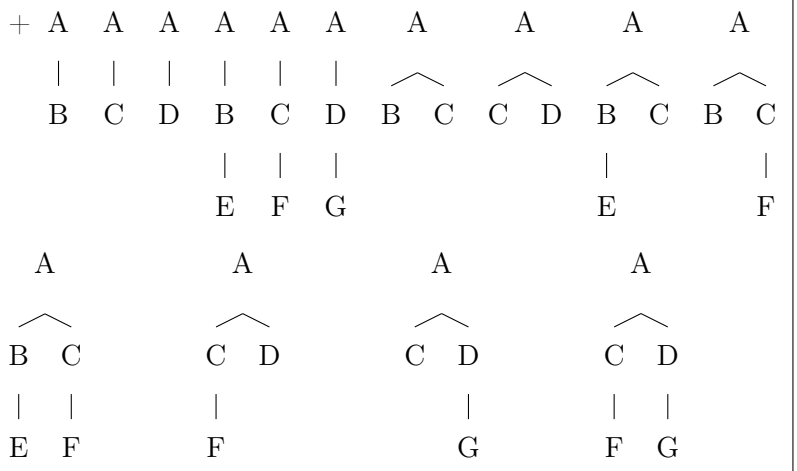
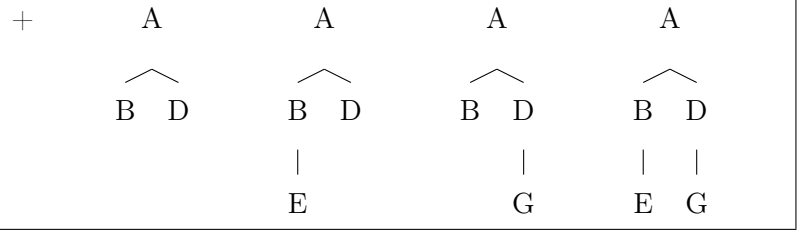
Type of Kernel	Implicit Feature Space
Sub-set Tree (SST) [Collins and Duffy, 2002]	
Contiguous Tree (CT) [Zelenko *et al.*, 2002]	
Sparse Tree (ST) [Zelenko *et al.*, 2002]	

Table 3.5: Types of tree kernels and their implicit feature space for the tree [.A [.B E] [.C F] [.D G]] (framed tree in the second row of the table). The plus sign (+) indicates "in addition to above structures." So the implicit feature space of SST is smaller than that of CT and the feature space for CT is smaller than that of ST.

Zelenko *et al.* [2002] propose the use of a shallow parse trees with part of speech and entity type information added as attributes to nodes. The authors experiment with contiguous and sparse tree kernels and report results on two classification tasks (person-affiliation, yes or no, and organization-location, yes or no). For both tasks, the Sparse tree kernel outperforms both the Contiguous kernel and the feature based baselines. Culotta and Sorensen [2004] employ a dependency tree representation for the ACE relation detection and classification tasks. The authors add a wide range of attributes to the nodes of dependency trees: part-of-speech (24 values), general-pos (5 values), chunk-tag, entity-type, entity-level, Wordnet hypernyms, and relation-argument. Culotta and Sorensen [2004] experiment with composite kernels (linear combinations of various kernels) and report that a linear combination of Contiguous and bag-of-words kernel performs slightly better than other kernel combinations for both the tasks. The authors also report that relation detection is a much harder task than relation classification. Harabagiu *et al.* [2005] build on the work of Culotta and Sorensen [2004] by adding more attributes to the nodes in dependency trees. Specifically, Harabagiu *et al.* [2005] add grammatical function, frame information from PropBank and FrameNet, and a larger set of features derived from WordNet. Zhang *et al.* [2006] explore the use of phrase structure trees and introduce an array of tree pruning operations. The authors report that the Path Enclosed Tree ("the smallest common sub-tree including the two entities") outperforms other tree pruning operations. Furthermore, instead of adding attributes to nodes, the authors propose adding attributes as separate nodes to trees.

Nguyen *et al.* [2009] re-visit much of past literature, propose novel data representations, and perform a comprehensive set of experiments for achieving the state-of-the-art ACE relation extraction system. We thus closely follow their work to design and engineer data representations and kernel combinations for our tasks. Details of these representations are deferred to Section 4.2.4.

3.2.6 Rule Based

In one of sub-fields of bioinformatics, researchers are interested in studying the physical or regulatory interactions of genes and proteins. Two of the most widely cited techniques for automatically obtaining an interaction network of genes and proteins are: (1) co-occurrence

based [Ding *et al.*, 2002; Jelier *et al.*, 2005] and (2) rule based [Fundel *et al.*, 2007]. Jelier *et al.* [2005] create a co-occurrence matrix of genes where two genes are said to co-occur if they both appear in "the abstract, title or MeSH headings of one document. The matrix contains the number of times genes from the set co-occur." We experiment with a co-occurrence based baseline for our social event detection task (details deferred to Section 4.2.1).

Fundel *et al.* [2007] note that while this simple approach of finding co-occurrences has high recall, the approach has low precision. The authors therefore propose a more sophisticated technique where they hand-craft a set of syntactic rules over dependency trees. Fundel *et al.* [2007] use domain knowledge and a small "list of restriction-terms that are used to describe relations of interest." These specific syntactic rules over dependency trees are not directly applicable for our problem, mainly because we cannot use a restricted list of words to describe social events; while the interaction of genes may be expressed using a restricted set of terms, the interaction of humans is far more general. However, the idea of using syntactic rules over dependency trees is useful and we experiment with one such baseline presented in Section 4.2.2.

3.3 Conclusion

In this chapter, we presented a formal definition of our machine learning tasks. Since these tasks are closely related to the well studied task of relation extraction in the NLP community, we presented a survey of the vast amount of literature available for relation extraction. We organized the relation extraction literature into five broad categories: Bootstrapping, Unsupervised, Distant Supervision, Regular Supervision (feature based and convolution kernel based), and Rule based. We argued that bootstrapping and unsupervised techniques are not applicable for our tasks. Techniques that use distant supervision are also not directly applicable. However, the general concept of distant supervision is applicable and we use this concept for extracting social networks from movie screenplays in the last part of this thesis. Feature based and convolution kernel based approaches are most directly applicable. We use these approaches as baselines and experiment with novel kernel combinations in the next chapter. We take motivation from unsupervised and rules based approaches to experiment

with a baseline that utilizes paths on dependency trees.

Dimension	Type	References
Type of Tree Kernel	Sub-set	Zhang *et al.* [2006], Nguyen *et al.* [2009], Plank and Moschitti [2013]
	Contiguous	Zelenko *et al.* [2002], Culotta and Sorensen [2004], Harabagiu *et al.* [2005]
	Sparse	Zelenko *et al.* [2002], Culotta and Sorensen [2004], Harabagiu *et al.* [2005], Nguyen *et al.* [2009]
Type of Tree	Shallow Parse	Zelenko *et al.* [2002],
	Phrase Structure Tree (PST)	Zhang *et al.* [2006], Nguyen *et al.* [2009], Plank and Moschitti [2013]
	Dependency Tree (DT)	Culotta and Sorensen [2004], Harabagiu *et al.* [2005], Nguyen *et al.* [2009]
Type of Tree Operation	Add attributes to nodes	Zelenko *et al.* [2002], Culotta and Sorensen [2004], Harabagiu *et al.* [2005], Zhang *et al.* [2006]
	Add nodes to tree	Zhang *et al.* [2006], Nguyen *et al.* [2009], Plank and Moschitti [2013]
	Prune tree	Zhang *et al.* [2006], Nguyen *et al.* [2009], Plank and Moschitti [2013]
Use other Resources	WordNet	Culotta and Sorensen [2004], Harabagiu *et al.* [2005], Zhang *et al.* [2006]
	Brown Clusters	Plank and Moschitti [2013]
	LSA	Plank and Moschitti [2013]
	PropBank	Harabagiu *et al.* [2005]
	FrameNet	Harabagiu *et al.* [2005]

Table 3.6: Data representations for tree kernels.

Chapter 4

Machine Learning Approach

In this chapter, we present our machine learning approach for automatically extracting social networks from unstructured texts. Our general approach is based on the use of convolutions kernels – both subsequence kernels and tree kernels. This chapter extends the work first presented in Agarwal and Rambow [2010], Agarwal [2011], Agarwal *et al.* [2013], and Agarwal *et al.* [2014a].

We experiment with a wide variety of data representations already introduced for relation extraction and propose four new structures: one subsequence structure that is a sequence of nodes on a special dependency tree (details deferred to later) and three tree kernel representations that attempt to combine the feature spaces from all levels of language abstractions (lexical, syntactic, and semantic). By semantics we mean frame semantics, specifically the ones derived from the FrameNet annotations. We further introduce a set of linguistically motivated hand-crafted frame semantic features and compare their performance with other baselines and systems. Our results show that hand-crafted frame semantic features add less value to the overall performance in comparison with the frame-semantic tree kernels. We believe this is due to the fact that hand-crafted features require frame parses to be highly accurate and complete. In contrast, tree kernels are able to find and leverage less strict patterns without requiring the semantic parse to be entirely accurate or complete. In summary, following are the contributions of this chapter:

- We show that convolution kernels are task independent. This is a nice property to have

because the same kernel representations may be used for different tasks (relation extraction and social network extraction). We show that fine grained feature engineering based approaches do not adapt well to a new task. They tend to be task dependent.

- We show that linguistically motivated semantic rules do not perform well. In contrast, trees that incorporate semantic features outperform other systems by a significant margin for the social event detection task. However, for the overall task of social network extraction, they perform at par with the pure syntax based structures. We believe, this is due to the performance of the semantic parsers and the sparsity of FrameNet.

The rest of the chapter is organized as follows: Section 4.1 provides details about the data, the data distribution, our annotation procedure, and the inter-annotator agreement for annotating social events in the data. Section 4.2 presents our baselines: co-occurrence based, syntactic rule based, feature based, and convolution kernel based. We introduce two other baselines that make use of frame semantics: bag-of-frames and semantic rule based baseline in the same section. Section 4.3 presents the new data representations that we propose for our tasks. We mainly propose two kinds of structures: a sequence structure on dependency trees and three tree based semantic structures. We conclude and provide future directions of research in Section 4.5.

4.1 Data

We annotate the English part of the Automatic Content Extraction (ACE) 2005 Multilingual Training Corpus[1] for creating a training and test set for our tasks. We refer to this corpus as the **ACE-2005 corpus** throughout this document. We choose to annotate this corpus for the following three reasons:

- The ACE-2005 corpus already contains annotations for entities and their mentions. This makes the overall annotation task of annotating social events easier; social events

[1] `https://catalog.ldc.upenn.edu/LDC2006T06`. LDC Catalog number: LDC2006T06

are between entity mentions and thus social event annotations require entity mention annotations.

- The ACE-2005 corpus has a wide variety of data sources including weblogs, broadcast news, newsgroups, and broadcast conversations.
- The ACE-2005 corpus is a well-distributed and a widely used corpus in the NLP community.

As an example of the annotations that already exist and the ones we annotate, consider the following sentence from a news article in the ACE-2005 corpus:

> [Hu, who was appointed to the top job in March], will meet his [Russian counterpart Vladimir Putin] during his three-day state visit from May 26 to 28

Entity mention annotations are in square brackets [...]. The *heads* of entity mentions are underlined. These annotations already exist. For annotating a social event, we simply highlight the word *meet* as the trigger word for an INR social event between the two entity mentions. Following sections provide more details about our annotation procedure and reliability.

4.1.1 Annotation Procedure

We use Callisto [Day *et al.*, 2004], a configurable annotation tool, for our annotation purposes. We work with two annotators. The annotators import each file in the ACE-2005 corpus into Callisto, perform annotations, and save the annotated files in LDC's standard XML file format. Figure 4.1 shows an example of such an XML file. This example shows the offset annotations for one entity and two of its mentions. An entity has an identifier, referred to as "ID', and other attributes such as "TYPE", "SUBTYPE", and "CLASS". Similarly, entity mentions have attributes such as "ID", "TYPE", "LDCTYPE", and "LDCATR". Each mention has an *extent* and a *head*. The XML tag "<extent>" specifies the span of the entity mention in the document. The XML tag "<head>" specifies the span of the head of the entity mention in the document. Both extent and head have a *charseq*. The XML tag "charseq" has two attributes: "START" and "END". The attribute "START" specifies the

number of characters from the beginning of the file until the start of the entity mention. The attribute "END" specifies the number of characters from the beginning of the file until the end of the entity mention. For example, the head of the first listed mention (*minister*) starts at character 916 and ends at character 923 in the document.

```
<entity ID="XIN_ENG_20030616.0274-E1" TYPE="PER" SUBTYPE="Individual" CLASS="SPC">
  <entity_mention ID="XIN_ENG_20030616.0274-E1-5" TYPE="NOM" LDCTYPE="BAR" LDCATR="TRUE">
    <extent>
      <charseq START="908" END="974">foreign minister of Greece, the current holder of
the EU presidency</charseq>
    </extent>
    <head>
      <charseq START="916" END="923">minister</charseq>
    </head>
  </entity_mention>
  <entity_mention ID="XIN_ENG_20030616.0274-E1-11" TYPE="PRO" LDCTYPE="PRO" LDCATR="FALSE">
    <extent>
      <charseq START="982" END="983">he</charseq>
    </extent>
    <head>
      <charseq START="982" END="983">he</charseq>
    </head>
  </entity_mention>
  <entity_mention ID="XIN_ENG_20030616.0274-E1-40" TYPE="NAM" LDCTYPE="NAM" LDCATR="FALSE">
    <extent>
      <charseq START="889" END="905">George Papandreou</charseq>
    </extent>
    <head>
      <charseq START="889" END="905">George Papandreou</charseq>
    </head>
  </entity_mention>
  <entity_attributes>
    <name NAME="George Papandreou">
      <charseq START="889" END="905">George Papandreou</charseq>
    </name>
  </entity_attributes>
</entity>
```

Figure 4.1: Snippet of an offset XML file distributed by LDC.

When this offset annotation file is imported into Callisto, it appears as in Figure 4.2. Callisto highlights entity mentions in different shades of blue (complete entity mentions in light blue and their head in dark blue). For example, in Figure 4.2, the entity mention, **foreign minister of Greece, the current holder of the EU presidency**, is highlighted in light blue, whereas the head of this mention, **minister**, is highlighted in a darker shade of blue.

A social event annotation is a quadruple, (TARGET$_1$, TARGET$_2$, TYPE, SUBTYPE). TARGET$_1$ and TARGET$_2$ are entities of type PERSON (defined in Section 2.1), TYPE refers to the type of social event (we have two broad categories of social events, OBS and INR, as defined in Section 2.2), and SUBTYPE refers to the subtype of social events (as defined

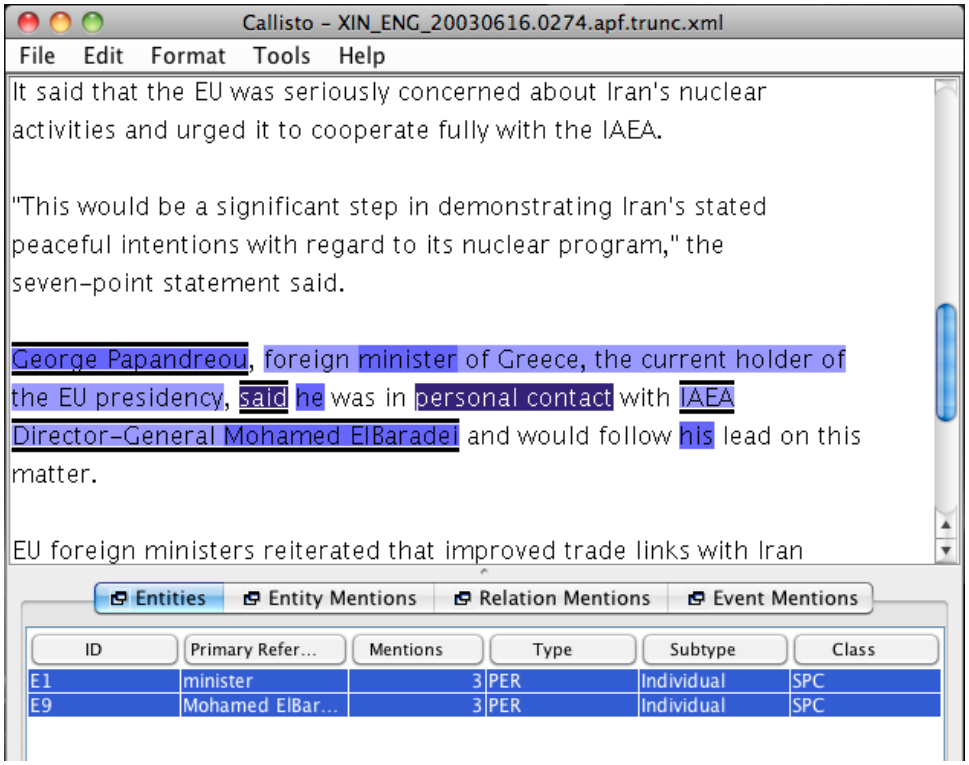


Figure 4.2: Snapshot of Callisto. Top screen has the text from a document. Bottom screen has tabs for Entities, Entity Mentions etc. An annotator selected text *said*, highlighted in dark blue, as an event of type OBS.FAR between entities with entity ID E1 and E9.

in Section 2.3). We ask our annotators to select the *span of text* (span is defined below) that triggers a social event, select the two entities that participate in this event, and select the type and subtype of the event to complete one social event annotation. For example, in Figure 4.2, one of our annotators selects the word *said* as a trigger for a social event of type OBS and subtype FAR between the entity mentions **George Papandreou** and **Mohamed ElBaradei**. The same annotator selects the phrase *personal contact* as a trigger for a social event of type INR.VERBAL.FAR between the entity mentions **he** and **Mohamed ElBaradel**.

Span of Social Events: We define the span of a social event as the *minimum* span of text that *best* represents the event being recorded. The span may be a word, a phrase or the whole sentence. Usually, spans are verbs and associated auxiliaries in a sentence. Since we do not use social event spans for any purpose, we do not enforce an exact selection of

event spans by our annotators. As long as the spans selected by the two annotators overlap, we accept the span annotation.

4.1.2 Additional Annotation Instructions

In the course of several annotation rounds, we added the following set of instructions to our annotation manual. The following examples and instructions are representative of the major sources of confusion that our annotators faced while annotating social events.

Specific People in a Larger Group: There are examples in which specific people in a group participate in social events. In such cases, we ask our annotators to record a social event between the specific person or specific group of people and the other entity. For example, in the following sentence, we ask our annotators to record a social event only between entities **2 sisters** and **Anna**.

(24) [2 sisters] out of [8 siblings] went and {talked} to [Anna]. INR.VERBAL.near

Similarly, in the following sentence, we ask our annotators to record a social event only between entities **At least three members** and **tribal mob**.

(25) [At least three members] of a [family in Indians northeastern state of Tripura] were {hacked to death} by a [tribal mob] for allegedly practicing witchcraft, police said Thursday. INR.NON-VERBAL.near

Legal Actions: We ask our annotators to annotate all legal actions such as "sue", "convict", etc. as social events of type INR.VERBAL. Legal actions may be of subtype FAR or NEAR. We ask the annotators decide the subtype from the context. For example, in the following sentence, an entity (**Anne**) sues another entity (**Crichton**). Since there is no evidence that the two entities are in physical proximity, the relevant subtype is FAR and not NEAR. However, in Example 27, the subtype is NEAR because the two entities are in a courtroom.

(26) [Anne] {sued} [Crichton] of alimony payments. INR.VERBAL.far

(27) [Anne] {accused} [Crichton] of robbery in the courtroom. INR.VERBAL.near

4.1.3 Inter-annotator Agreement

We work with two annotators. After several rounds of training, we ask the two annotators to annotate the same set of 46 documents. Out of these, one document does not contain any entity annotations. The average number of entities per document in the remaining set of 45 documents is 6.82 and the average number of entity mentions per document is 23.78. The average number of social events annotated per document by one annotator is 3.43. The average number of social events annotated per document by the other annotator is 3.69.

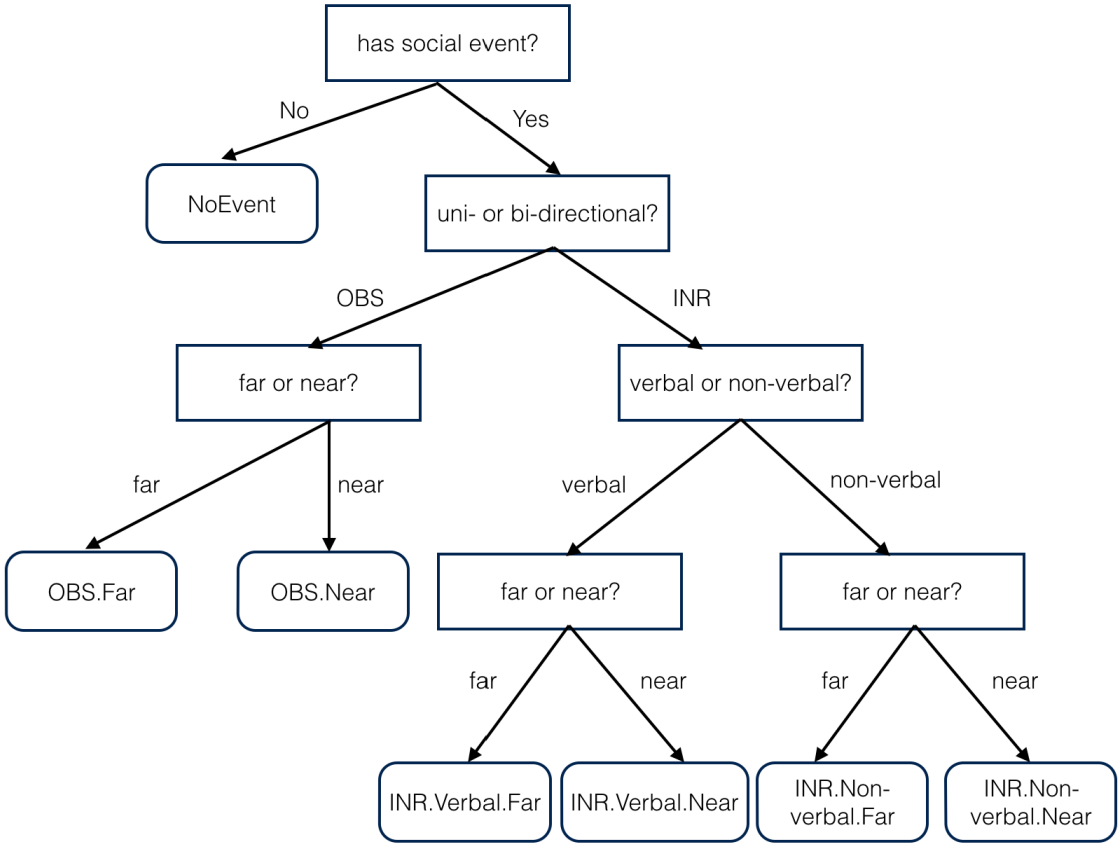


Figure 4.3: Set of decisions that an annotator makes for selecting one of seven social event categories.

Figure 4.3 presents the set of decisions that an annotator has to make before selecting one of seven social event categories (the leaves of this tree). The first decision that an annotator has to make is whether or not two there is a social event between the two entity

mentions. If the answer is yes, then the annotator has to decide the category of the social event: OBS (a unidirectional event) or INR (a bidirectional event). The decision process continues until the annotator reaches one of the leaves of the tree.

Due to the novelty of the annotation task, and the conditional nature of the labels, we assess the reliability of the annotation of each decision point. We report Cohen's Kappa [Cohen, 1960] for each independent decision. We use the standard formula for Cohen's Kappa (κ) given by:

$$\kappa = \frac{P(a) - P(e)}{1 - P(e)}$$

where $P(a)$ is probability of observed agreement and $P(e)$ is probability of chance agreement.

In addition, we present the confusion matrix for each decision point to show the absolute number of cases considered, and F-measure to show the proportion of cases agreed upon. For most decision points, the Kappa scores are at or above the 0.67 threshold recommended by Krippendorff [1980] with F-measures above 0.90. Where Kappa is low, F-measure remains high. As discussed below, we conclude that our annotations are reliable.

Decision Point		Confusion Matrix				Kappa	F1
S.No.	Decision	Y, Y	Y, N	N, Y	N, N		
1	has social event?	133	31	34	245	0.68	0.80
2	is bi-directional (INR)?	51	8	1	73	0.86	0.91
3	is INR.VERBAL?	40	4	0	7	0.73	0.95
4	is INR.VERBAL.NEAR?	30	1	2	7	0.77	0.95
5	is INR.NON-VERBAL.NEAR?	6	0	1	0	0.00	0.92
6	is OBS.FAR?	71	0	1	1	0.66	0.99

Table 4.1: This table presents two inter-annotator agreement measures (Kappa in Column 7 and F1 measure in the last column). Columns 3-6 show the flattened confusion matrix for each decision point. Y, Y refers to Yes, Yes i.e. both annotators say Yes to a question in Column 2. Y, N refers to Yes, No i.e. the first annotator says Yes but the second annotator says No to a question, and so on.

Table 4.1 presents the results for the six binary decisions that annotators make for

arriving at their final social event annotation. The number of the decision points in the table correspond to the six interior nodes in the decision tree shown in Figure 4.3. The (flattened) confusion matrices in column two present annotator two's choices by annotator one's, with positive agreement in the upper left (cell Y, Y) and negative agreement in the lower right (cell N, N). For example, for the first decision point, both annotators say yes (denoted by Y, Y) to the question "has social event?" Both annotators agree that there is a social event between 133 out of 443 examples. Both annotators agree that there is no social event in 245 out of 443 examples. There are 65 (31 + 34) examples on which the annotators disagree. The Kappa for deciding whether or not there is a social event is 0.68 and the F1-measure is 0.80. Note that the sum of values in row two is 133 (51 + 8 + 1 + 73). For calculating the Kappa for the second decision point, we consider all the examples where the two annotators agree to have a social event.

In all cases, the cell values on the agreement diagonal (Y, Y; N, N) are much higher than the cells for disagreement (Y, N; N, Y). Except for the first two decisions, the agreement is always unbalanced towards agreement on the positive cases, with few negative cases. The case of the fifth decision, for example, reflects the inherent unlikelihood of the INR.NON-VERBAL.FAR event. In other cases, it reflects a property of the genre. When we apply this annotation schema to fiction, we find a much higher frequency of OBS.NEAR events.

For the fifth decision, the Kappa score is low but the confusion matrices and high F-measures demonstrate that the absolute agreement is high. Kappa measures the amount of agreement that would not have occurred by chance, with values in [-1,1]. For binary data and two annotators, values of -1 can occur, indicating that the annotators have perfectly non-random disagreements. The probability of an annotation value is estimated by its frequency in the data (the marginals of the confusion matrix). It does not measure the actual amount of agreement among annotators, as illustrated by the rows for the fifth decision. Because NON-VERBAL.FAR is chosen so rarely by either annotator (never by the second annotator), the likelihood that both annotators will agree on NON-VERBAL.NEAR is close to one. In this case, there is little room for agreement above chance, hence the Kappa score of zero.

The five cases of high Kappa and high F-measure indicate aspects of the annotation where annotators generally agree, and where the agreement is unlikely to be accidental. We

conclude that these aspects of the annotation can be carried out reliably as independent decisions. The case of low Kappa and high F-measure indicate aspects of the annotation where, for this data, there is relatively little opportunity for disagreement.

We note that in the ACE annotation effort, inter-annotator agreement (IAA) is measured by a single number, but this number does not take chance agreement into account: it simply uses the evaluation metric normally used to compare systems against a gold standard.[2] Furthermore, this metric is composed of distinct parts which are weighted in accordance with research goals from year to year, meaning that the results of applying the metric change from year to year.

We present a measure of agreement for our annotators by using the ACE evaluation scheme. We consider one annotator to be the gold standard and the other to be a system being evaluated against the gold standard. For the calculation of this measure we, first take the union of all event spans. As in the ACE evaluation scheme, we associate penalties with each wrong decision annotators take about the entities participating in an event, type and sub-type of an event. Since these penalties are not public, we assign our own penalties. We choose penalties that are not biased towards any particular event type or subtype. We decide the penalty based on the number of options an annotator has to consider before making a certain decision. For example, we assign a penalty of 0.5 if one annotator records an event which the other annotator does not. If annotators disagree on the event type, the penalty is 0.50 because there are two options to select from (INR, OBS). Similarly, we assign a penalty of 0.25 if the annotators disagree on the event sub-types (VERBAL.NEAR, VERBAL.FAR, NON-VERBAL.NEAR, NON-VERBAL.FAR). We assign a penalty of 0.5 if the annotators disagree on the participating entities (incorporating the directionality in directed relations). Using these penalties, we achieve an agreement of 69.74% on all social event categories and subcategories. This is a high agreement rate as compared to that of ACE's event annotation, which was reported to be 31.5% at the ACE 2005 meeting.[3]

[2] http://www.itl.nist.gov/iad/mig//tests/ace/2007/doc/ace07-evalplan.v1.3a.pdf

[3] Personal communication, Rebecca Passonneau.

4.2 Baselines

This section presents all the baselines that we use for assessing the complexity of the task. The first baseline, CoOccurN, is a rule based baseline that simply counts the number of co-occurances of two entities in a sentence to predict whether or not these entities are participating in a social event. This baseline does not use any syntax or other linguistic features. The second baseline, SynRule, uses paths on dependency trees for making predictions. The third baseline, GuoDong05, is the state-of-the-art feature based system for ACE relation extraction. This baseline uses bag-of-words, parts-of-speech tags, and shallow syntactic parses for extracting features. The fourth baseline, Nguyen09, is the state-of-the-art convolution kernel based system for ACE relation extraction. This baseline uses phrase structure trees and dependency trees for creating data representations that are used by subsequence and tree kernels for prediction. The last set of baselines, BOF and SemRules, which we introduce in this work, make use of frame semantics.

4.2.1 Co-occurrence Based (CoOccurN)

A trivial way of extracting a network from text is by connecting all pairs of entities that appear together in a sentence. For example, this co-occurrence based approach will create connections between all pairs of entities in sentence 28: **Elton** and **Mr. Knightley**, **Elton** and **Mr. Weston**, and **Mr. Knightley** and **Mr. Weston**.

(28) [Elton]'s manners are superior to [Mr. Knightley]'s or [Mr. Weston]'s.

The predictions made by this baseline are incorrect. There is no social event between any pair of entities in the sentence because the author simply states a fact; the author does not give the reader any clue about the cognitive states of these entities. It should be clear that this technique can only be used for social event detection (and not for social event classification or directionality classification). We refer to this technique as CoOccurN with $N = 1$. We additionally experiment with other values for N by connecting all pairs of entities that appear together in at least N sentences in a document.

4.2.2 Syntactic Rule Based (SynRule)

We take motivation from work in information extraction [Banko *et al.*, 2007] and bioinformatics [Fundel *et al.*, 2007] for the design of this baseline. Banko *et al.* [2007] use the following heuristics for the self-training of their relation extractors (taken verbatim from their paper):

- There exists a dependency chain between e_i and e_j that is no longer than a certain length.
- The path from e_i to e_j along the syntax tree does not cross a sentence-like boundary (e.g. relative clauses).
- Neither e_i nor e_j consist solely of a pronoun.

Fundel *et al.* [2007] consider dependency paths between entities. The kind of paths they are interested in are domain and relation specific (like effector-relation-effectee). Nonetheless, it seems that heuristics concerning paths in dependency trees are important. Given this motivation, we formulate the SynRule baseline.

Consider a dependency parse tree with two entity mentions marked with tags $T1$ and $T2$ ($T1$ appears before $T2$ in the surface order). Figure 4.4 shows one such dependency parse tree of the sentence *Military officials say a missile hit his warthog and he was forced to eject.* Notice the nodes T1-Group and T2-Individual. The tag T1 marks the first target with the entity type Group. The tag T2 marks the second target with the entity type Individual.

Define path $P12$ to be the *downward* path from the parent of $T1$ to $T2$; a downward path means a path from the current node towards the leaves. Similarly, define path $P21$ to be the downward path from the parent of $T2$ to $T1$. It is possible that both or neither paths exist. Given the dependency tree (or example $\vec{x}$) and this definition of path, SynRule is defined as follows:

$$\text{SynRule}(\vec{x}) = \begin{cases} INR & \text{both paths P12 and P21} \\ \overrightarrow{OBS} & \text{only P12 exists} \\ \overleftarrow{OBS} & \text{only P21 exists} \\ \text{NoEvent} & \text{neither path exists} \end{cases}$$

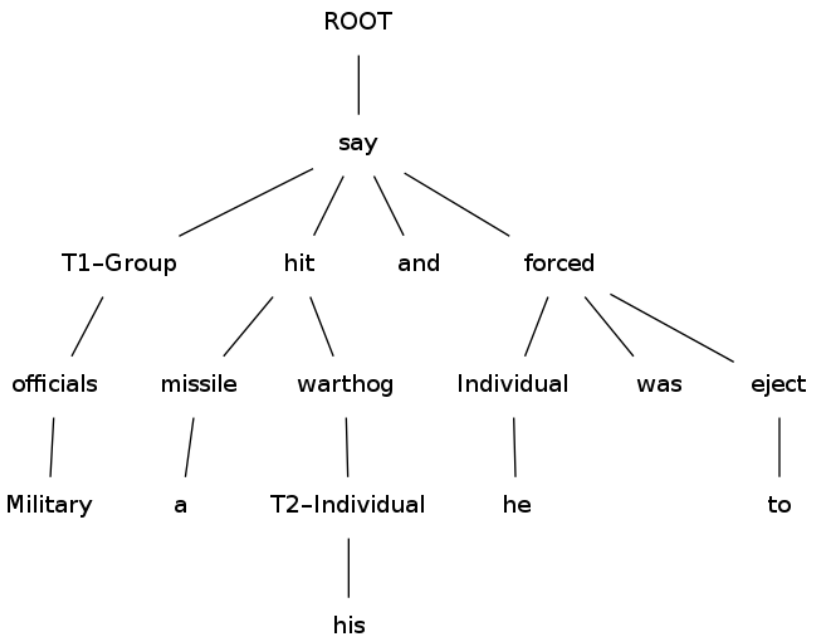


Figure 4.4: A full dependency parse for the sentence *Military officials say a missile hit his warthog and he was forced to eject.* There is an OBS social event between the entities **Military officials** and **his**. The SynRule baseline makes the correct prediction because path $P12$ exists but path $P21$ does not exist.

Figures 4.4, 4.5, and 4.6 illustrate the functionality of the SynRule baseline. Starting with Figure 4.4, there are two targets in the sentence: *Military officials* (marked with tag T1-Group) and *his* (marked with tag T2-Individual). In this example, the baseline makes a correct prediction; there is a $\overrightarrow{OBS}$ social event from **Military officials** to **his**. This is because the officials are talking about the person. Path $P12$ – the path from the parent of the first target to the other – exists; $P12 = say \rightarrow hit \rightarrow warthog \rightarrow T2 - Individual$. Path $P21$ does not exist; there is no downward path from *warthog* to $T1 - Group$. Since $P12$ exists but $P21$ does not exist, the SynRule baseline predicts the social event to be $\overrightarrow{OBS}$.

Figure 4.5 shows a full dependency parse for the sentence *He had to say to her.* There are two targets in the sentence: *He* (marked with tag T1-Individual) and *her* (marked with tag T2-Individual). In this example, the baseline makes an incorrect prediction; there is an INR social event between **He** and **her** but the baseline predicts an $\overrightarrow{OBS}$ social event. Path

$P12$ exists; $P12 = had \rightarrow say \rightarrow to \rightarrow T2 - Individual$. Path $P21$ does not exist. Since $P12$ exists but $P21$ does not exist, the SynRule baseline predicts the social event to be $\overrightarrow{OBS}$.

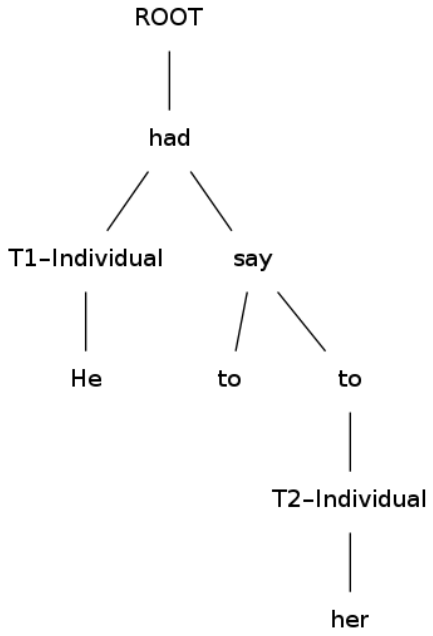


Figure 4.5: A full dependency parse for the sentence *He had to say to her.* There is an INR social event between the entities **he** and **her**. The SynRule baseline makes an incorrect prediction; it predicts $\overrightarrow{OBS}$. This is because the path $P12$ exists but path $P21$ does not exist.

Figure 4.6 shows a full dependency parse of the sentence *On behalf of republican candidates and I tend to do a lot of campaigning in the next year for the president.* There are two targets in the sentence: *I* (marked with tag T1-Individual) and *the president* (marked with tag T2-Individual). In this example, the baseline makes an incorrect prediction; there is an $\overrightarrow{OBS}$ social event from **I** to **the president** but the baseline predicts there to be no social event. This is because neither of the paths P12 and P21 exists.

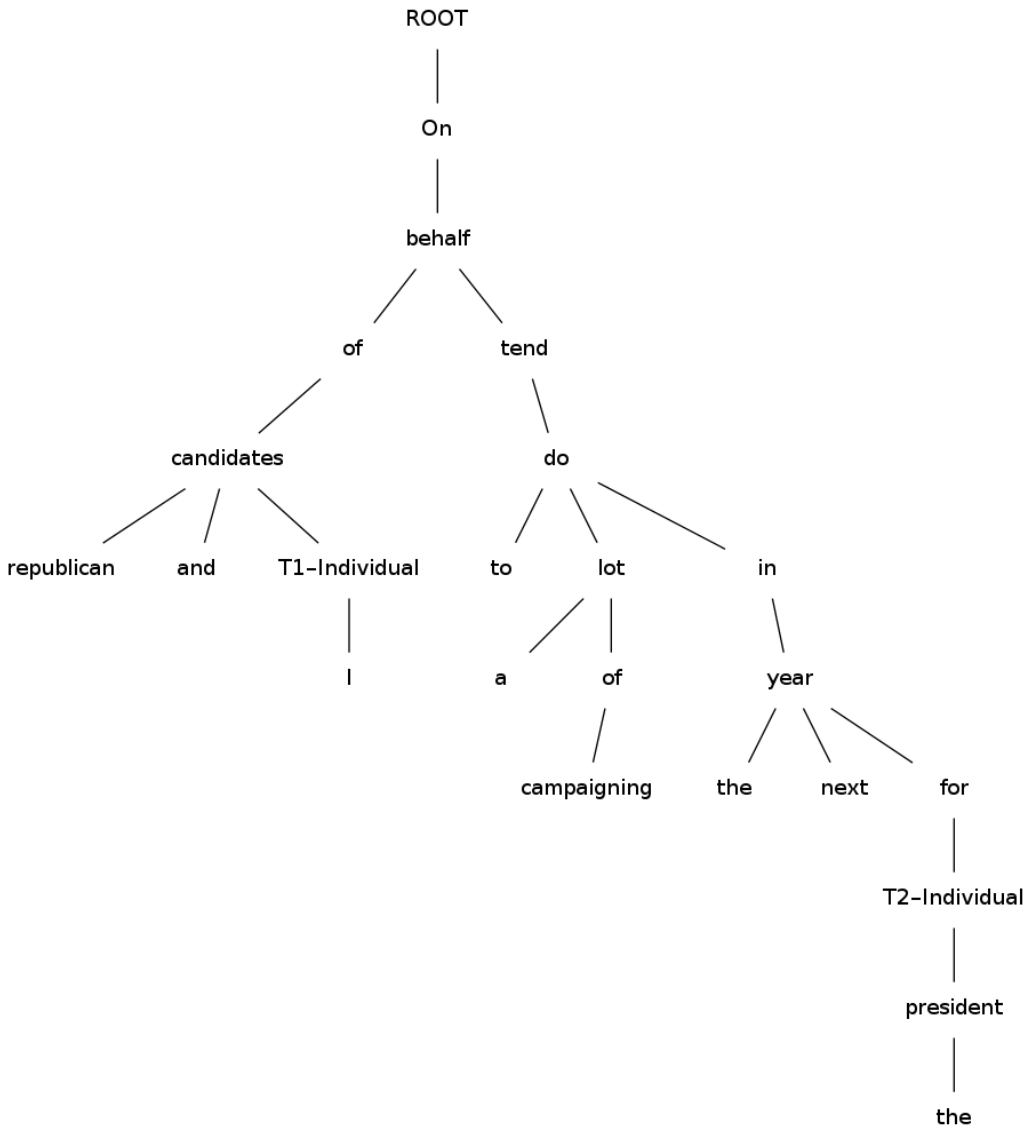


Figure 4.6: A full dependency parse for the sentence *On behalf of republican candidates and I tend to do a lot of campaigning in the next year for the president.* There is an $\overrightarrow{OBS}$ social event from **I** to **the president**. The SynRule baseline makes an incorrect prediction; it predicts NoEvent. This is because neither of the paths $P12$ and $P21$ exists.

4.2.3 Feature Based Supervision (GuoDong05)

GuoDong *et al.* [2005] introduce a state-of-the-art feature based supervised system for ACE relation extraction. Some of the features they introduce are too specific to the ACE relation

extraction task. For example, the set of features that capture the words of entity mentions. These features are predictive of ACE classes but are irrelevant for our tasks (words in "my sister" are indicative of the PER-SOC ACE relation but are irrelevant for our tasks). Apart from such task dependent features, we experiment with all other features that GuoDong *et al.* [2005] introduce. We refer to the baseline that uses the union of all these features as GuoDong05.

The first feature vector that GuoDong *et al.* [2005] propose make use of words. This feature vector is a concatenation of three boolean vectors: $\{\vec{b_1}, \vec{b_2}, \vec{b_3}\}$. $\vec{b_1}$ captures the words before the first target ($T1$), $\vec{b_2}$ between the two targets and $\vec{b_3}$ after the second target ($T2$). Here, the *first target* and the *second target* are defined in terms of the surface word order; the first target appears before the second target in the surface order. We refer to this feature vector as BOW_GuoDong05. We experiment with this feature vector alone, in addition to experimenting with the entire feature set GuoDong05. Other features derived from words include (verbatim from GuoDong *et al.* [2005]):

- WBNULL: when no word in between
- WBFL: the only word in between when only one word in between
- WBF: first word in between when at least two words in between
- WBL: last word in between when at least two words in between
- WBO: other words in between except first and last words when at least three words in between
- BM1F: first word before M1
- BM1L: second word before M1
- AM2F: first word after M2
- AM2L: second word after M2

Mention Level: There are three types of entity mentions: {Name, Nominal, Pronoun}. We record the combination of entity mentions that appear in an example. There are a total of nine combinations such as Name-Name, Name-Nominal, Nominal-Name, and so on.

Base Phrase Chunking: GuoDong *et al.* [2005] use a perl script[4] for obtaining a shallow parse of sentences. This script requires a phrase structure tree as input and produces shallow parse along with phrase heads. We use Stanford's parser for obtaining the phrase structure tree. Following is the set of base phrase chunking features (verbatim from GuoDong *et al.* [2005]):

- CPHBNULL when no phrase in between
- CPHBFL: the only phrase head when only one phrase in between
- CPHBF: first phrase head in between when at least two phrases in between
- CPHBL: last phrase head in between when at least two phrase heads in between
- CPHBO: other phrase heads in between except first and last phrase heads when at least three phrases in between
- CPHBM1F: first phrase head before M1
- CPHBM1L: second phrase head before M1
- CPHAM2F: first phrase head after M2
- CPHAM2F: second phrase head after M2
- CPP: path of phrase labels connecting the two mentions in the chunking
- CPPH: path of phrase labels connecting the two mentions in the chunking augmented with head words, if at most two phrases in between

4.2.4 Convolution Kernel Based Supervision (NGUYEN09)

Nguyen *et al.* [2009] introduce a comprehensive set of tree based and string based representations for relation extraction. The most appealing aspect of convolution kernels is that they obviate the need for fine grained feature engineering. Large classes of features may be represented in form of coarser *structures* (strings and trees) and depending on the end task,

[4] `http://ilk.kub.nl/~sabine/chunklink/`

the classifier identifies the set of fine grained features that are essential for classification. We experiment with all the structures that Nguyen *et al.* [2009] introduce for the task of ACE relation extraction. Following sections provide a description of these structures.

4.2.4.1 Tree Based Structures

Figure 4.7 presents variations of the tree data representations proposed by Nguyen *et al.* [2009]. The first structure (Figure 4.7 (a)) is a constituent parse of the sentence *In Washington, U.S. officials are working overtime.* According to ACE annotations, there is a directed *Physical.Located* relation between the entities *officials* and *Washington.* The dotted line indicates the Path Enclosed Tree (**PET**) structure. Notice the addition of nodes (T2-LOC, GPE, T1-PER) to the tree. These nodes capture the entity type information that has been shown to be a useful feature for ACE relation extraction [Kambhatla, 2004; GuoDong *et al.*, 2005]. The tag T1 marks the first target. The tag T2 marks the second target.

The second structure (shown in Figure 4.7 (b)) is the dependency parse of the same sentence. The third structure (Figure 4.7 (c)), called **DW** (dependency word tree), is derived from the dependency parse tree by adding target node annotations ($T2-LOC$ and $T1-PER$) to the dependency tree. The fourth structure (Figure 4.7 (d)) is derived from DW: the words are replaced with the grammatical roles of words. This structure is called **GR** (grammatical relation). The fifth structure (Figure 4.7 (e)) is combination of DW and GR: grammatical roles are added as separate nodes over the words in the DW tree. This structure is called **GRW** (grammatical relation word).

4.2.4.2 Sequence or String Based Structures

Nguyen *et al.* [2009] propose the following sequence structures (taken as is from their paper):

SK1 Sequence of terminals (lexical words) in the PET, e.g.: *T2-LOC Washington , U.S. T1-PER officials.*

SK2 Sequence of part-of-speech (POS) tags in the PET, i.e. the SK1 in which words are replaced by their POS tags, e.g.: *T2-LOC NN , NNP T1-PER NNS.*

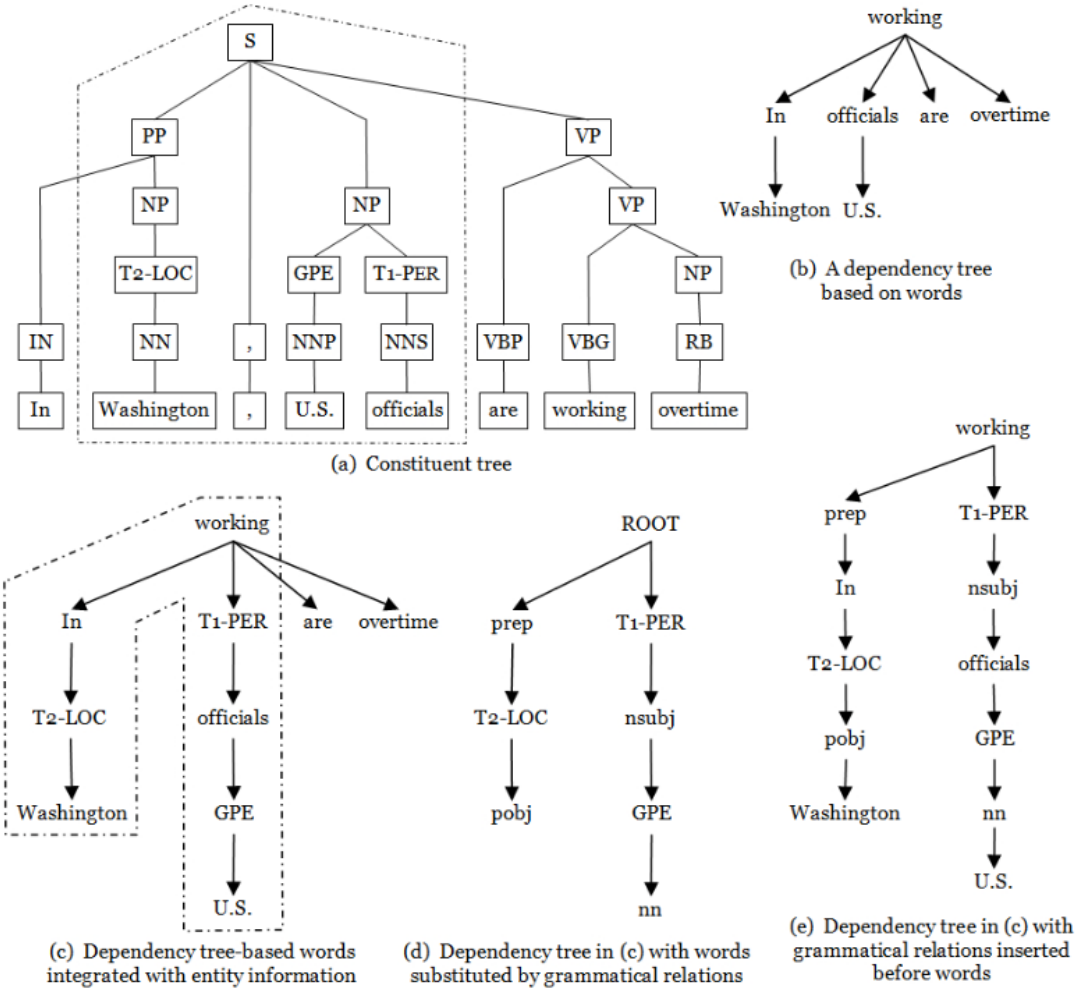


Figure 4.7: Tree kernel data representations proposed by Nguyen *et al.* 2009. This figure is taken from their paper. The dotted subtree in (a) is referred to as PET (path enclosed tree), the tree in (c) is referred to as DW (dependency word), the tree in (d) is referred to as GR (grammatical role), and the tree in (e) is referred to as GRW (grammatical role word).

SK3 Sequence of grammatical relations in the PET, i.e. the SK1 in which words are replaced by their grammatical functions, e.g.: *T2-LOC pobj , nn T1-PER nsubj.*

SK4 Sequence of words in the DW, e.g.: *Washington T2-LOC In working T1-PER officials GPE U.S.*

SK5 Sequence of grammatical relations in the GR, i.e. the SK4 in which words are replaced by their grammatical functions, e.g.: *pobj T2-LOC prep ROOT T1-PER nsubj GPE nn.*

SK6 Sequence of POS tags in the DW, i.e. the SK4 in which words are replaced by their POS tags, e.g.: *NN T2-LOC IN VBP T1-PER NNS GPE NNP.*

4.2.5 Features derived from FrameNet (BOF and SemRules)

FrameNet [Baker *et al.*, 1998] is a resource which associates words of English with their meaning. Word meanings are based on the notion of "semantic frame". A frame is a conceptual description of a type of event, relation, or entity, and it includes a list of possible participants in terms of the roles they play; these participants are called "frame elements". By way of an example, we present the terminology and acronyms that will be used throughout this document.

(29) [**FE−Speaker** Toujan Faisal] [**FEE−Statement** said] [**FE−Message** she was informed of the refusal by an Interior Ministry committee]

Example (29) shows frame annotations for the sentence *Toujan Faisal said she was informed of the refusal by an Interior Ministry committee.* One of the semantic frames in the sentence is **Statement**. The frame evoking element (FEE) for this frame is *said.* It has two frame elements (FE): one of type **Speaker** (*Toujan Faisal*) and the other of type **Message** (*she was informed ... by an Interior Ministry committee*). In example (29), the speaker of the message (*Toujan Faisal*) is *mentioning* another group of people (the *Interior Ministry committee*) in her message. By definition, this is a social event of type OBS. In general, there is an OBS social event between any **Speaker** and any person mentioned in

the frame element **Message** of the frame **Statement**. This close relation between frames and social events is the reason for our investigation and use of frame semantics for our tasks.

4.2.5.1 Bag of Frames (BOF)

We use Semafor [Chen *et al.*, 2010] for obtaining the semantic parse of a sentence. Using Semafor, we find 1,174 different FrameNet frames in our corpus. We convert each example into a vector of dimensionality 1,174 ($\vec{x}$). In this vector, x_i (the i^{th} component of vector $\vec{x}$) is 1 if the frame number i appears in the example, and 0 otherwise.

4.2.5.2 Hand-crafted Semantic Features (SemRules)

We use the manual of the FrameNet resource to hand-craft 240 rules that are intended to detect the presence and determine the type of social event between two entities mentioned in a sentence. Following are examples of two hand-crafted rules. Rule 30 applies to situations in which one entity is talking about another entity. For example, in the sentence *John said Mary is great*, Semafor detects a **Statement** frame evoked by the frame evoking element *said*, it detects the **Speaker** as *John*, and **Message** as *Mary is great*. Rule 30 fires and system correctly predicts an $\overrightarrow{OBS}$ relation from **John** to **Mary**.

(30) If the frame is **Statement**, and the first target entity mention is contained in the FE **Speaker**, and the second is contained in the FE **Message**, then there is an OBS social event from the first entity to the second.

(31) If the frame is **Commerce_buy**, and one target entity mention is contained in the FE **Buyer**, and the other is contained in the FE **Seller**, then there is an INR social event between the two entities.

Each rule corresponds to a binary feature: it takes a value 1 if the rule fires for an input example, and 0 otherwise. For example, in sentence 32, Semafor correctly detects the frame **Commerce_buy**, with *he* as the **Buyer**, *drugs* as the **Goods** and *the defendants* as the **Seller**. The hand-crafted rule (31) fires and the corresponding feature value for this rule is set to 1. Firing of these rules (and thus the effectiveness these features) is of course highly dependent on the fact that Semafor provides an accurate frame parse for the sentence.

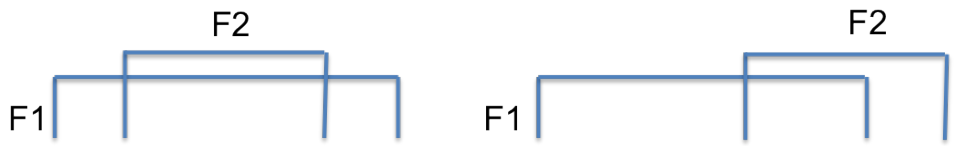


Figure 4.8: Two overlapping scenarios for frame annotations of a sentence, where $F1$ and $F2$ are frames.

(32) Coleman claimed [he] {bought} drugs from the [defendants].

Appendix B.1 presents the complete list of these 240 semantic rules.

4.3 Structures We Introduce

This section provides a description of four data representations that we propose for our tasks. The first representation, SqGRW, is a sequence on a special kind of dependency tree. The next three representations are tree representations constructed from frame parses of sentences.

4.3.1 Sequence on Grammatical Relation Dependency Tree (SqGRW)

This structure is the sequence of words and their grammatical relations in the GRW tree (Figure 4.7 (e)). For the example in Figure 4.7 (e), SqGRW is *Washington pobj T2-LOC In prep working T1-PER nsubj officials GPE nn U.S.*

4.3.2 Semantic trees (FrameForest, FrameTree, FrameTreeProp)

Semafor labels *text spans* in sentences as frame evoking elements (FEE) or frame elements (FE). A sentence usually has multiple frames and the frame annotations may overlap. There may be two ways in which spans overlap (Figure 4.8) : (a) one frame annotation is completely embedded in the other frame annotation and (b) some of the frame elements overlap (in terms of text spans). We now present the three frame semantic tree kernel representations that handle these overlapping issues, along with providing a meaningful semantic kernel representation for the tasks addressed in this work.

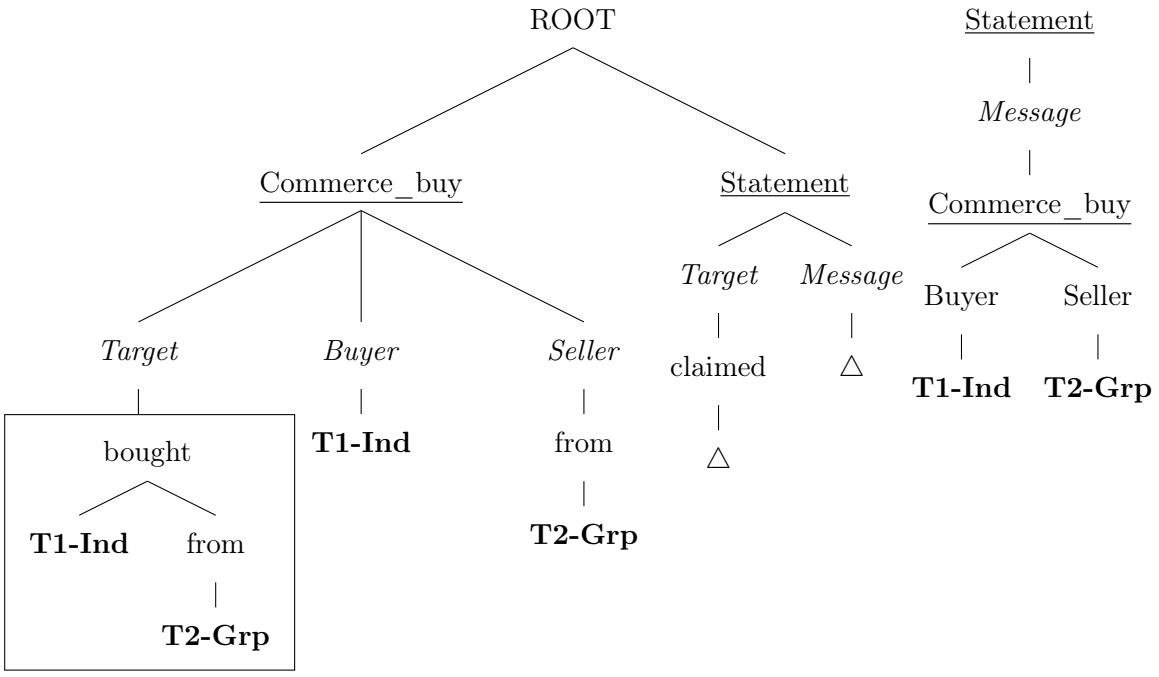


Figure 4.9: Semantic trees for the sentence "Coleman claimed [he]$_{T1-Ind}$ bought drugs from the [defendants]$_{T2-Grp}$, but offered little or no supporting evidence.". This example is annotated as INR. Clearly, if two entities are in a commercial transaction, they are mutually aware of each other and of the transaction taking place. The tree on the left is FrameForest and the tree on the right is FrameTree. △ in FrameForest refers to the boxed subtree. **Ind** refers to individual and **Grp** refers to group.

For each of the following representations, we assume, that for a sentence s, we have the set of semantic frames, $\mathbb{F}_s = \{F = \langle FEE, [FE_1, FE_2, \ldots, FE_n]\rangle\}$ with each frame F having an FEE (frame evoking element) and a list of FEs (frame elements). We explicate the structures using sentence (32): *Coleman claimed [he] {bought} drugs from the [defendants].*

4.3.2.1 FrameForest Tree Kernel

We first create a tree for each frame annotation F in the sentence. Consider a frame, $F = \langle FEE, [FE_1, FE_2, \ldots, FE_n]\rangle$. For the purposes of tree construction, we treat FEE as another FE (call it FE_0) of type *Target*. For each FE_i, we choose the subtree from the dependency parse tree that is the smallest subtree containing all words annotated as FE_i by Semafor. Call this subtree extracted from the dependency parse $DepTree\_FE_i$. We then create a larger tree by adding $DepTree\_FE_i$ as a child of frame element FE_i: (FE_i $DepTree\_FE_i$). Call this resulting tree $SubTree\_FE_i$. We then connect all the $SubTree\_FE_i$ ($i \in \{0, 1, 2, \ldots, n\}$) to a new root node labeled with the frame F: (F $SubTree\_FE_0$ ... $SubTree\_FE_n$) This is the tree for a frame F. Since the sentence could have multiple frames, we connect the *forest* of frame trees to a new node called $ROOT$. We prune away all subtrees that do not contain the target entities. The resulting tree is called the FrameForest Tree.

For example, in Figure 4.9, the left tree is the FrameForest tree for sentence (32). There are two frames in this sentence that appear in the final tree because both these frames contain the target entities and thus are not pruned away. The two frames are **Commerce_buy** and **Statement**. We first create trees for each of the frames. For the **Commerce_buy** frame, there are three frame elements (in our extended sense): *Target* (the frame evoking element), *Buyer* and *Seller*. For each frame element, we get the subtree from the dependency tree that contains all the words belonging to that frame element. The subtree for FEE *Target* is (*bought* T1-Ind (*from* T2-Grp)). The subtree for FE *Buyer* is (T1-Ind) and the subtree for FE *Seller* is (*from* T2-Grp). We connect these subtrees to their respective frame elements and connect the resulting subtrees to the frame (**Commerce_buy**). Similarly, we create a tree for the frame **Statement**. Finally, we connect all frame trees to the $ROOT$.

In this representation, we have avoided the frame overlapping issues by repeating the

common subtrees. In this example, the subtree (*bought* T1-Ind (*from* T2-Grp)) is repeated under the FEE **Target** of the **Statement** frame as well as under the FE **Message** of the **Statement** frame.

4.3.2.2 FrameTree Tree Kernel

For the design of this tree, we deal with the two overlapping conditions shown in Figure 4.8 differently. If one frame is fully embedded in another frame, we add it as a child of the appropriate frame element of the embedding frame. If the frames overlap partially, we copy over the overlapping portions to each of the frames. Moreover, we remove all lexical nodes and trees that do not span any of the target entities. As a result, this structure is the smallest purely semantic structure that contains the two target entities.

The right tree in Figure 4.9 is the FrameTree tree for sentence (32). Since the frame **Commerce_buy** is fully embedded in the FE **Message** of frame **Statement**, it appears as a child of the *Message* node. Also, *from* does not appear in the tree because we remove all lexical items.

4.3.2.3 FrameTreeProp Tree Kernel

We use a sparse tree kernel (ST, see Table 3.5) for calculating the similarity of trees. The ST kernel does not *skip* over nodes of the tree that lie on the same path. For example, one of the subtrees in the *implicit feature space* of FrameTree will be (Commerce_buy (Buyer T1-Ind) (Seller T2-Grp)) but it might be useful to have the following subtree in the implicit feature space: (Commerce_buy T1-Ind T2-Ind). For this reason, we copy the nodes labeled with the target annotations ($T1-*, T2-*$) to all nodes on the path from them to the root in FrameTree. We call this variation of FrameTree, in which we *propagate* $T1-*, T2-*$ nodes to the root, FrameTreeProp. For the running example, FrameTreeProp will be: (Statement T1-Ind T2-Grp (Message T1-Ind T2-Grp (Commercial_buy T1-Ind T2-Grp (Buyer T1-Ind) (Seller T2-Grp)))).

4.4 Experiments and Results: Intrinsic Evaluation

4.4.1 Task Definitions

Our overall task is to classify a social event between every pair of entity mentions (belonging to two different entities) in a sentence into one of four categories: {$\overrightarrow{OBS}$, $\underleftarrow{OBS}$, INR, NoEvent}. We explore two different methodologies for building a multi-class classifier: (1) one-versus-all (OVA) classifier and (2) hierarchal (HIE) classifier. Using the OVA approach, we build four models: {$\overrightarrow{OBS}$-versus-All, $\underleftarrow{OBS}$-versus-All, INR-versus-All, NoEvent-versus-All}. Using the HIE approach, we stack three classifiers in a hierarchy: NoEvent-versus-All followed by INR-versus-{$\overrightarrow{OBS}$,$\underleftarrow{OBS}$}, followed by $\overrightarrow{OBS}$-versus-$\underleftarrow{OBS}$. For the hierarchal methodology, we also report results for each of the three models: Social Event Detection (NoEvent-versus-All), Social Event Classification (INR-versus-{$\overrightarrow{OBS}$,$\underleftarrow{OBS}$}), and Directionality Classification ($\overrightarrow{OBS}$-versus-$\underleftarrow{OBS}$).

4.4.2 Data Distribution

An example is defined as a three tuple: (first entity mention, second entity mention, type of social event). We consider each pair of entity mentions (referring to different entities) in a sentence as an example. For instance, sentence 33 below contains three entity mentions: **My**, **President Bush**, and **Dick Cheney**. In this example, the entity **My** is *talking about* the other two entities. Therefore, there is an $\overrightarrow{OBS}$ social event directed from **My** to the other two entities. Since there is no evidence about the cognitive states of **President Bush** and **Dick Cheney**, there is no social event between these two entities. We create three examples from this sentence: (**My**, **President Bush**, $\overrightarrow{OBS}$), (**My**, **Dick Cheney**, $\overrightarrow{OBS}$), and (**President Bush**, **Dick Cheney**, NoEvent).

(33) [My] {focus is on} re-electing [President Bush] and [Dick Cheney] next year, the convention is going to be here in the City of New York.

Given this methodology for creating examples, Table 4.2 presents the distribution of our gold standard used for building and evaluating our classifiers. As the data distribution suggests, the number of examples that have an event between the entities are in minority

$(\frac{(382+63+356)*100}{382+63+356+4186} = 16.06\%)$.

#docs	#words	$\#\overrightarrow{OBS}$	$\#\underleftarrow{OBS}$	#INR	#NoEvent
265	109,698	382	63	356	4,186

Table 4.2: Data distribution of our gold standard.

4.4.3 Experimental Set-up

We use 5-fold cross-validation on the training set for parameter tuning, exploration of data representations, and search for the best combination of kernels. We use SVM-Light-TK [Joachims, 1999; Moschitti, 2004] for building our classifiers. Due to data skewness, we report F1-measure for all the tasks. Since the objective function of SVM optimizes for accuracy, in a skewed data distribution scenario, SVMs tend to learn a trivial function that classifies all examples into the majority class. To avoid this, we penalize mistakes on the minority class more heavily. This forces the classifier to learn a non-trivial function. We set the penalty on making a mistake on a minority class to the ratio of the number of examples in the majority class and the number of examples in minority class.

To avoid over-fitting to a particular partition of folds, we run each 5-fold experiment 50 times, for 50 randomly generated partitions. The results we report in the following tables are all averaged over these 50 partitions. The absolute standard deviation of F1-measures on average is less than 0.004. This is a small deviation, indicating that our models are robust. We use McNemar's significance test and refer to statistical significance as $p < 0.05$. For calculating significance across 50 partitions, we first calculate significance per partition. If $p > 0.05$ even for a single partition, we report that the results are not significantly different.

4.4.4 Experiments and Results for Social Event Detection

Figure 4.10 presents the precision (square markers), recall (circular markers), and F1-measure (triangular markers) curves for our CoOccurN baseline as N varies from 1 to 19. Recall, the CoOccurN baseline predicts a social event between two entities if the entities co-occur in at least N sentences in a document. The figure shows that, as expected, the

recall is highest for $N = 1$ (when we connect all pairs of entities that co-occur in a sentence). However, the precision for $N = 1$ is low (0.16). The F1-measure is 0.28. As N increases, the recall decreases monotonically, the precision increases, then decreases, and eventually increases for large N. The F1-measure increases from $N = 1$ to $N = 2$ and then decreases monotonically. The best F1-measure using this baseline is 0.29 which is worse than our best performing system by a large and significant margin (0.29 versus 0.56). This confirms that an obvious baseline that simply connects co-occurring characters together is not sufficient for the task of social event detection.

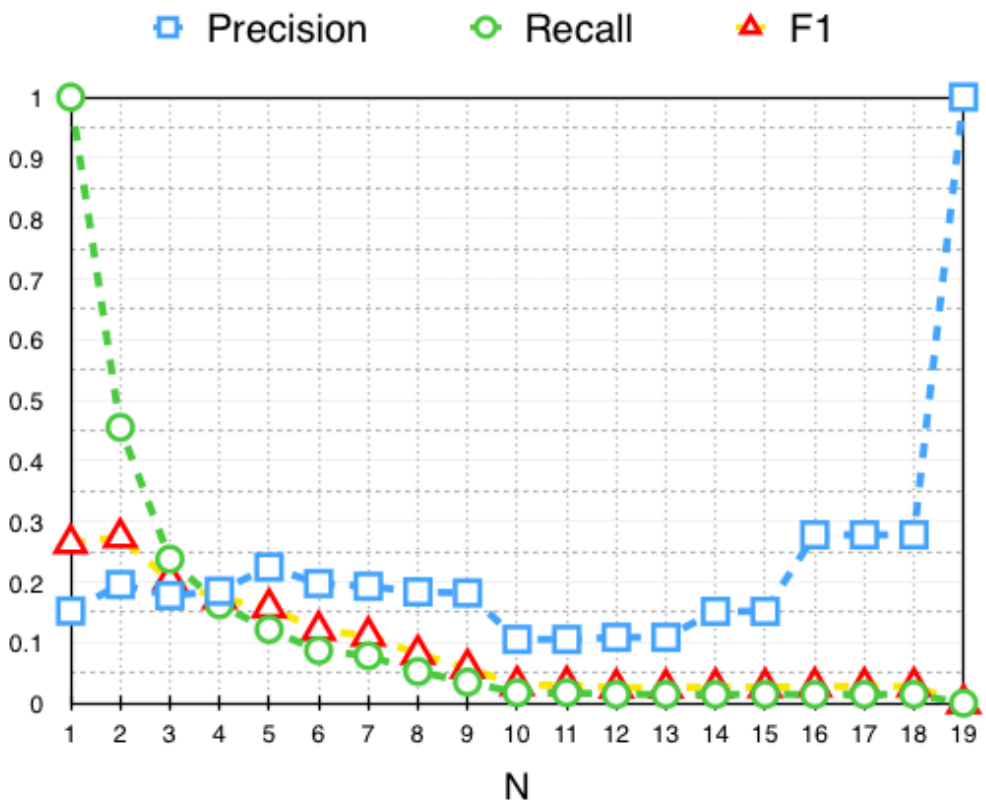


Figure 4.10: Precision, Recall, and F1 measure for the CoOccurN baseline. X-axis denotes the number of co-occurrences of entities in a document.

Table 4.3 presents a comparison of performance of all the baselines presented in Section 4.2 with our best performing system for the task of Social Event Detection. The results show that both rule based baselines, CoOccurN and SynRule, perform much worse than

feature and convolution kernel based systems. The CoOccurN baseline predicts that all pairs of entity mentions (belonging to different entities) in a sentence have a social event. While it trivially achieves the highest recall, the precision is low (0.16). CoOccurN achieves an F1-measure of 0.28. The syntactic rule based baseline (SynRule) that considers paths in dependency trees has the highest precision (0.89) but low recall (0.19). SynRule achieves an F1-measure of 0.31.

The state-of-the-art feature based system for relation extraction, GuoDong05, also performs much worse than several convolution kernel based systems. GuoDong05 achieves an F1-measure of 0.39 compared to, for example, 0.51 F1-measure achieved by the convolution kernel based system PET. This confirms that convolution kernel based approaches are task independent. Features may be represented as tree structures and depending on the task, the classifier learns to put more weight on fine grained features that are important for the task at hand, thus making it possible for the same data representations to be utilized for different tasks like ACE relation detection and social event detection.

One set of features that the GuoDong05 baseline uses is BOW. BOW performs at an F1-measure of 0.36 compared to the GuoDong05 baseline, which performs at an F1-measure of 0.39. This shows that the features other than BOW in the GuoDong05 baseline contribute only 0.03 F1-measure to the overall performance.

The results show that our purely semantic models (SemRules, BOF, FrameTree, FrameTreeProp) do not perform well alone. FrameForest, which encodes some lexical and syntactic level features (but is primarily semantic), also performs worse than other baselines but better than pure semantic based representations. It is a combination of lexical, syntactic and semantic structures that outperform all the baselines by a statistically significant margin, achieving an F1-measure of 0.56 (GRW_PET_FrameForest_FrameTree _FrameTreeProp_SemRules_SqGRW). The best system that does not utilize semantics, GR_GRW_PET_SK5_SK6, performs worse than the system that utilizes semantics (F1-measure of 0.54 versus 0.56). This difference is small but statistically significant. We conclude that information gathered from frame semantics is useful for the task of social event detection.

We see that the hand-crafted SemRules do not help in the overall task. We investigated

the reason for SemRules not being as helpful as we had expected. We found that when there is no social event, the rules fire in 10% of the cases. When there is a social event, they fire in 16% of cases. So while they fire more often when there is a social event, the percentage of cases in which they fire is small. We hypothesize that this is due the dependence of SemRules on the correctness of semantic parses. For example, Rule 34 correctly detects the social event in sentence 35, since Semafor correctly parses the input. In contrast, Semafor does not correctly parse the input sentence 37 (see Figure 4.11): it correctly identifies the **Telling** frame and its **Addressee** frame element, but it fails to find the **Speaker** for the **Telling** frame. As a result, Rule 36 does not fire, even though the semantic structure is partially identified. This, we believe, highlights the main strength of tree kernels – they are able to learn semantic patterns, without requiring correctness or completeness of the semantic parse. The rule based baseline, SemRules, that utilizes the same frame parses as the convolution kernel based structures does not perform well; it achieves an F1-measure of 0.2.

(34) If the frame is **Commerce_buy**, and one target entity mention is contained in the FE **Buyer**, and the other is contained in the FE **Seller**, then there is an INR social event between the two entities.

(35) Coleman claimed [he] {bought} drugs from the [defendants].

(36) If the frame is **Telling**, and the first target entity mention is contained in the FE **Addressee**, and the second is contained in the FE **Speaker**, then there is an OBS social event from the first entity to the second.

(37) Toujan Faisal said [she] {was informed} of the refusal by an [Interior Ministry committee].

Note that both the best performing systems (syntactic and semantic) contain a combination of phrase structure tree representation (PET) and dependency tree representations (GR, GRW). Even though there is a one-to-one mapping from one representation to the other (i.e. both representations encode the same information), the representations offer complementary information for the classifier.

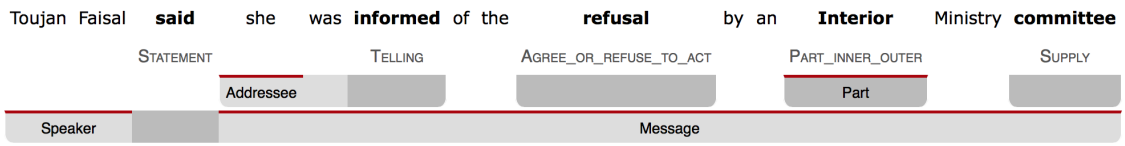


Figure 4.11: Semantic parse for the sentence *Toujan Faisal said [she] {was informed} of the refusal by an [Interior Ministry committee].*

4.4.5 Experiments and Results for Social Event Classification

Table 4.4 presents a comparison of performance of all the systems for the task of Social Event Classification. Recall, this is the task to classify a social event into one of two categories: {INR, OBS}. Our dataset has 445 social events of type OBS and 356 events of type INR (see Table 4.2). Unlike the previous task, where we reported the F1-measure for only the minority class, we report the macro-F1-measure for both classes. This is because it is important for us to perform well on both the classes. Since the distributions of INR and OBS are similar (about 44% INR and 56% OBS), averaging the precision and recall for the two classes results in average precision being similar to the average recall (notice that the numbers in the Precision and Recall columns are similar). These numbers may also be interpreted as the accuracy on this task.

As explained below, a random class baseline (referred to as RANDOMCLASSBASELINE) achieves a macro-F1-measure of 0.50 on this task. Assume that the random class baseline classifies half of the INR social events as OBS and half of the OBS social events INR. Under this assumption, the number of true positives for the INR class is $356/2 = 178$, the number of false positives for the INR class (equal to the number of false negatives for the OBS class) is $382/2 = 191$, and the number of false negatives for the INR class (equal to the number of false positives for the OBS class) is $356/2 = 178$. Also, the number of true positives for the OBS class is 191. Using this confusion matrix, the averaged precision for the two classes is 0.4995 and the averaged recall is 0.50, giving a macro-F1-measure of close to 0.5.

Unlike the previous task, evaluation of the SYNRULE baseline is not applicable for this task. Since SYNRULE is a rule based system, we cannot restrict its prediction to only two classes. SYNRULE classifies examples with gold class one of {INR, OBS} as NULL. It is not

possible to construct a confusion matrix for SynRule where the gold and predicted classes are restricted to one of two classes {INR, OBS} and hence an evaluation is not applicable.

As for the previous task, the feature based baselines, both syntactic (BOW, GuoDong05) and semantic (BOF, SemRules), perform worse than the kernel based approaches. BOW achieves an F1-measure of 0.72, GuoDong05 achieves an F1-measure of 0.72, BOF achieves an F1-measure of 0.56, and SemRules achieves an F1-measure of 0.58. The best performing kernel based approaches, GR_PET_SK4_SK5 and GRW_PET_BOF_SemRules_SqGRW, both achieve an F1-measure of 0.81.

Note that the overall performance on this task is relatively higher compared to the previous task (0.81 versus 0.56). This suggests that classifying social events into the two categories OBS and INR is easier than detecting whether or not there is a social event between two entities.

Also note that the state-of-the-art feature based approach for relation extraction, GuoDong05, performs as well as the BOW approach (both achieve an F1-measure of 0.72). This observation reinforces the conclusion that fine-grained feature engineered systems are not well equipped for task independence. In contrast, convolution kernel based approaches are well equipped.

As for the previous task, semantic features and structures do not perform well alone. But when combined with syntactic structures, they achieve the highest performance (F1-measure of 0.81). Unlike the previous task in which semantic structures were crucial in achieving the best performing system, a purely syntactic based model also achieves the highest performance for this task (GR_PET_SK4_SK5 achieves an F1-measure of 0.81).

4.4.6 Experiments and Results for Directionality Classification

Table 4.5 presents a comparison of performance of all the systems for the task of Directionality Classification. Recall, this is the task to classify an OBS social event into one of two categories: $\{\overrightarrow{OBS}, \underleftarrow{OBS}\}$. Our dataset has 63 social events of type $\underleftarrow{OBS}$ and 382 events of type $\overrightarrow{OBS}$ (see Table 4.2). Like the previous task, we report the macro-F1-measure for both classes. As for the previous task, evaluation of the SynRule baseline is not applicable for this task.

The BOW baseline performs relatively well. It achieves an F1-measure of 0.78 compared to the best performing kernel based method, BOW_GRW_SqGRW, which achieves an F1-measure of 0.82. The use of semantics actually hurts the performance for this task; the best structure that uses semantics achieves a performance of 0.81 compared to a purely syntactic based approach that achieves an F1-measure of 0.82. However, SqGRW, one of the structures we propose, is crucial in achieving the best performance. The best syntactic structure that does not use SqGRW performs significantly worse; BOW_PET achieves an F1-measure of 0.79.

4.4.7 Experiments and Results for Social Network Extraction (SNE)

Table 4.4 presents a comparison of performance of all the systems for the task of Social Network Extraction. Recall, this is the task to classify a social event into one of four categories: {INR, $\overrightarrow{OBS}$, $\underleftarrow{OBS}$, NULL}. We explore two methodologies for building a classifier for the end task, social network extraction: (1) one-versus-all (OVA) approach and (2) hierarchal (HIE) approach. Using the OVA approach, we build four models: {$\overrightarrow{OBS}$-versus-All, $\underleftarrow{OBS}$-versus-All, INR-versus-All, NoEvent-versus-All}. Using the HIE approach, we stack three classifiers in a hierarchy: NoEvent-versus-All followed by INR-versus-{$\overrightarrow{OBS}$,$\underleftarrow{OBS}$}, followed by $\overrightarrow{OBS}$-versus-$\underleftarrow{OBS}$. We use the set-up described in Section 4.4.3 for reporting results.

Table 4.6 shows that a rule based system SynRule performs much worse than our feature based baselines and kernel methods. The SynRule baseline achieves a macro-F1-measure of 0.20 where as our best performing system achieves an F1-measure of 0.41 (structures GR_GRW_PET_SqGRW and GR_PET_FrameForest _FrameTreeProp_SqGRW). Other baselines such as BOW and GuoDong05 also perform significantly worse achieving an F1-measure of 0.28 and 0.32 respectively. The results for a rule based system for HIE and OVA are the same. For other structures, we note that the performance of OVA approach is always better than the HIE approach (with the exception of SK6).

Note that the precision of the rule based system SynRule is the highest – 0.51. This is expected as rule based systems are usually highly precise but lack recall.

The results show that semantic structures do not help significantly for the overall task.

GR_PET_FrameForest _FrameTreeProp_SqGRW achieves the same F1-measure as the best syntactic structure based system, GR_GRW_PET_SqGRW, both achieving an F1-measure of 0.41. However, the best system that does not utilize any of our structures performs, GR_GRW_PET, performs significantly worse, achieving an F1-measure of 0.39.

4.5 Conclusion and Future Work

We showed that convolution kernels are task independent. Several string and tree structures, previously proposed for the task of ACE relation extraction adapt well for our tasks. On each of our four tasks, the kernel based approaches outperformed the rule based and the feature based approaches by large and significant margins. We also showed that linguistically motivated hand-crafted semantic rules did not perform well. In contrast, trees that incorporated semantic features outperformed other systems by a significant margin for the social event detection task and performed at par for other tasks. Our experiments and results also showed that as a result of how language expresses the relevant information, dependency-based structures are best suited for encoding this information. Furthermore, because of the complexity of the task, a combination of phrase based structures and dependency-based structures performed the best. This re-validates the observation of Nguyen *et al.* [2009] that phrase structure representations and dependency representations add complimentary value to the learning task. We introduced a new sequence structure (SqGRW) which plays a role in achieving the best performing system for the overall task of social network extraction.

As future work, we would like to explore the and experiment with recent advancements in using distributional semantics for NLP tasks. The limitations of Semafor, which partly have to do with the sparsity of FrameNet are real challenges in using frame semantics. We would like to explore if distributional semantics can help us alleviate this limitation.

Type	Model for Social Event Detection	Precision	Recall	F1-measure
Rule based	CoOccurN with $N = 1$	0.16	**1.0**	0.28
	SynRule	**0.89**	0.19	0.31
Lex. and Syn. feature based	BOW	0.32	0.4	0.36
	GuoDong05	0.36	0.42	0.39
Syntactic kernel based	PET	0.4	0.68	0.51
	DW	0.36	0.56	0.43
	GR	0.37	0.7	0.48
	GRW	0.39	0.65	0.49
	SK1	0.31	0.68	0.43
	SK2	0.3	0.71	0.42
	SK3	0.29	0.69	0.41
	SK4	0.32	0.68	0.44
	SK5	0.31	0.76	0.44
	SK6	0.24	0.78	0.37
	⋆ SqGRW	0.36	0.74	0.48
	GR_GRW_PET_SK5_SK6 †	0.43	0.74	0.54
Semantic feature based	⋆ SemRules	0.32	0.14	0.2
	⋆ BOF	0.17	0.04	0.07
Semantic kernel based	⋆ FrameForest	0.32	0.45	0.38
	⋆ FrameTree	0.25	0.33	0.29
	⋆ FrameTreeProp	0.28	0.4	0.33
Kernel combination	⋆ GRW_PET_FrameForest _FrameTree _FrameTreeProp _SemRules_SqGRW	0.45	0.72	**0.56**

Table 4.3: A comparison of performance of all the baselines with our best performing system for the task of Social Event Detection. † refers to a novel kernel combination. The basic structures in this combination have been proposed by Nguyen et. al 2009. ⋆ refers to the new structures and combinations we propose in this work.

Type	Models for Social Event Classification	Precision	Recall	F1-measure
Rule based	RandomClassBaseline	0.5	0.50	0.50
	SynRule	N/A	N/A	N/A
Lex. and Syn. feature based	BOW	0.73	0.71	0.72
	GuoDong05	0.73	0.72	0.72
Syntactic kernel based	PET	0.75	0.76	0.75
	DW	0.74	0.74	0.74
	GR	0.75	0.75	0.75
	GRW	0.78	0.78	0.78
	SK1	0.74	0.75	0.75
	SK2	0.64	0.64	0.64
	SK3	0.67	0.66	0.67
	SK4	0.74	0.75	0.74
	SK5	0.72	0.72	0.72
	SK6	0.7	0.7	0.7
	⋆ SqGRW	0.77	0.77	0.77
	GR_PET_SK4_SK5†	**0.81**	0.81	**0.81**
Semantic feature based	⋆ SemRules	0.61	0.55	0.58
	⋆ BOF	0.57	0.55	0.56
Semantic kernel based	⋆ FrameForest	0.64	0.63	0.64
	⋆ FrameTree	0.55	0.54	0.54
	⋆ FrameTreeProp	0.61	0.60	0.60
Kernel combination	⋆ GRW_PET_BOF_SemRules_SqGRW	**0.81**	**0.82**	**0.81**

Table 4.4: A comparison of performance of all the baselines with our best performing system for the task of Social Event Classification. † refers to a novel kernel combination. The basic structures in this combination have been proposed by Nguyen et. al 2009. ⋆ refers to the new structures and combinations we propose in this work.

Type	Models for Directionality Classification	Precision	Recall	F1-measure
Rule based	SynRule	N/A	N/A	N/A
Lex. and Syn. feature based	BOW	0.90	0.69	0.78
	GuoDong05	0.90	0.69	0.78
Syntactic kernel based	PET	0.65	0.74	0.69
	DW	0.65	0.72	0.68
	GR	0.59	0.69	0.63
	GRW	0.59	0.68	0.64
	SK1	0.69	0.74	0.71
	SK2	0.60	0.69	0.64
	SK3	0.63	0.74	0.68
	SK4	0.64	0.66	0.65
	SK5	0.55	0.60	0.58
	SK6	0.56	0.61	0.59
	⋆ SqGRW	0.61	0.65	0.63
	BOW_PET†	0.85	0.74	0.79
	⋆ BOW_GRW_SqGRW	**0.93**	0.73	**0.82**
Semantic feature based	⋆ SemRules	0.57	0.58	0.58
	⋆ BOF	0.52	0.53	0.53
Semantic kernel based	⋆ FrameForest	0.56	0.6	0.58
	⋆ FrameTree	0.54	0.57	0.55
	⋆ FrameTreeProp	0.57	0.62	0.6
Kernel combination	⋆ BOW_GRW_FrameTree _SqGRW	0.91	0.73	0.81

Table 4.5: A comparison of performance of all the baselines with our best performing system for the task of Directionality Classification. † refers to a novel kernel combination. The basic structures in this combination have been proposed by Nguyen et. al 2009. ⋆ refers to the new structures and combinations we propose in this work.

	Social Network Extraction					
Model	Hierarchal Approach			One-versus-All Approach		
	P	R	F1	P	R	F1
SynRule	**0.51**	0.13	0.20	**0.51**	0.13	0.20
BOW	0.25	0.3	0.27	0.26	0.3	0.28
GuoDong05	0.28	0.33	0.31	0.29	0.36	0.32
PET	0.24	0.4	0.30	0.25	0.55	0.34
DW	0.22	0.36	0.27	0.25	0.38	0.30
GR	0.22	0.43	0.29	0.23	**0.58**	0.33
GRW	0.24	0.39	0.30	0.26	0.45	0.33
SK1	0.19	0.40	0.26	0.21	0.49	0.29
SK2	0.15	0.35	0.21	0.13	0.53	0.21
SK3	0.15	0.34	0.21	0.13	0.55	0.21
SK4	0.19	0.43	0.26	0.21	0.47	0.29
SK5	0.15	0.4	0.22	0.15	0.56	0.24
SK6	0.13	0.44	0.2	0.12	0.51	0.19
⋆ SqGRW	0.21	0.45	0.28	0.21	0.52	0.3
GR_GRW_PET†	0.28	0.47	0.35	0.30	0.53	0.39
⋆ GR_GRW_PET_SqGRW	0.29	0.5	0.37	0.32	0.57	**0.41**
⋆ SemRules	0.1	0.01	0.01	0.14	0.14	0.14
⋆ BOF	0.09	0.025	0.038	0.08	0.21	0.11
⋆ FrameForest	0.15	0.23	0.18	0.15	0.38	0.21
⋆ FrameTree	0.13	0.17	0.15	0.11	0.28	0.16
⋆ FrameTreeProp	0.14	0.2	0.16	0.12	0.42	0.19
⋆ GR_PET_FrameForest _FrameTreeProp_SqGRW	0.28	0.5	0.36	0.33	0.57	**0.41**

Table 4.6: A comparison of performance of all the baselines with our best performing system for the overall task of Social Network Extraction. † refers to a novel kernel combination. The basic structures in this combination have been proposed by Nguyen et. al 2009. ⋆ refers to the new structures and combinations we propose in this work.

Chapter 5

Application: Validating Literary Theories

5.1 Introduction

In his book *Graphs, Maps, Trees: Abstract Models for Literary History*, literary scholar Franco Moretti proposes a radical transformation in the study of literature [Moretti, 2005]. Advocating a shift from the close reading of individual texts in a traditionally selective literary canon, to the construction of abstract models charting the aesthetic form of entire genres, Moretti imports quantitative tools to the humanities in order to inform what he calls "a more rational literary history." While Moretti's work has inspired both support and controversy, this reimagined mode of reading opens a fresh direction from which to approach literary analysis and historiography.

By enabling the "distant reading" of texts on significantly larger scales, advances in Natural Language Processing (NLP) and applied Machine Learning (ML) can be employed to empirically evaluate existing claims or make new observations over vast bodies of literature. In a seminal example of this undertaking, Elson *et al.*; Elson [2010; 2012] set out to validate an assumption of structural difference between the social worlds of rural and urban novels using social networks extracted from nineteenth-century British novels. Extrapolating from the work of various literary theorists, Elson *et al.* [2010] hypothesize that nineteenth-century British novels set in urban environments feature numerous characters who share little con-

versation, while rural novels have fewer characters with more conversations. Using quoted speech attribution, the authors extract *conversational networks* from 60 novels (hereafter referred to as **LSN corpus** for Literary Social Networks), which had been manually classified by a scholar of literature as either rural or urban. Through the analysis of these conversational networks, Elson *et al.* [2010] conclude that their analysis provides no evidence to support the literary hypotheses that they derived from original theories. Specifically, their analysis indicates no difference between the social networks of rural and urban novels.

In this chapter, we employ SINNET for extracting a larger set of interactions (beyond conversations) and observations. This allows us to examine a wider set of hypotheses and thus gain deeper insights into the original theories. Our findings confirm that the setting (rural versus urban) of a novel in the LSN corpus has no effect on its social structure, even when one goes beyond conversations to more general notions of interactions and to a different notion of cognitive awareness, namely observations. Specifically, we extend the work of Elson *et al.* [2010] in five significant ways: (1) we extract *interaction networks*, a conceptual generalization of conversation networks; (2) we extract *observation networks*, a new type of network with directed links; (3) we consider unweighted networks in addition to weighted networks; (4) we investigate the number and size of communities in the extracted networks; and (5) propose and validate a wider set of literary hypotheses. This work was introduced in Jayannavar *et al.* [2015].

The rest of this chapter is organized as follows. In Section 5.2 we briefly present the literary theories that were validated by Elson *et al.* [2010]. Section 5.3 reminds the reader about the definitions of conversational, observation, and interaction networks. We present an evaluation of SINNET on the LSN corpus in Section 5.4. Section 5.5 presents our expanded set of literary hypotheses. Section 5.6 presents the methodology that Elson *et al.* [2010] use for validating their literary hypothesis. We use the same methodology for validating our expanded set of literary hypotheses. Section 5.7 presents the results for validating these literary hypotheses. We conclude and provide future directions for research in Section 5.8.

5.2 Literary Theories

In section 3 of their paper, Elson *et al.* [2010] present a synthesis of quotations from literary theorists Mikhail Bakhtin [Bakhtin, 1937], Raymond Williams [Williams, 1975], Franco Moretti [Moretti, 1999; 2005] and Terry Eagleton [Eagleton, 1996; 2013]. Elson *et al.* [2010] simplify the quotations to derive the following hypotheses (taken from Section 3 and page 4 of their paper):

- There is an inverse correlation between the amount of dialogue in a novel and the number of characters in that novel.
- Novels set in urban environments depict a complex but loose social network, in which numerous characters share little conversational interaction, while novels set in rural environments inhabit more tightly bound social networks, with fewer characters sharing much more conversational interaction.

Elson *et al.* [2010] define an *urban* novel to be "a novel set in a metropolitan zone, characterized by multiple forms of labor (not just agricultural). Here, social relations are largely financial or commercial in nature. Elson *et al.* [2010] conversely define a *rural* novel to be a novel set in a country or village zone, where agriculture is the primary activity, and where land-owning, non-productive, rent-collecting gentry are socially predominant. Social relations here are still modeled on feudalism (relations of peasant-lord loyalty and family tie) rather than the commercial cash nexus."

5.3 Conversational, Interaction, and Observation Networks

Before presenting our expanded set of hypotheses, we remind the reader about the definitions of– and the differences between– conversational, interaction, and observation networks. A more detailed account of the differences was presented in Section 2.6 of this thesis.

A conversational network is a network in which nodes are characters and links are conversations. Elson *et al.* [2010] define a conversation as follows:

> A continuous span of narrative time featuring a set of characters in which all of the following conditions are met: 1) The characters are either in the same place

> at the same time, or communicating by means of technology such as a telephone. 2) The characters take turns speaking. 3) The characters are mutually aware of each other and their dialogue is mutually intended for the other to hear. 4) Each character hears and understands the other's speech. A person present in a group is not counted in the conversation unless he or she speaks. Conversations that are related solely through a character's narration (i.e., stories told by characters) do not count.

As an example, consider the following excerpt from the novel *Emma* by Jane Austin. There are two entities in the excerpt: **Emma** and **Mr. Woodhouse**. These entities having a conversation (as defined by the four conditions above). The conversational network extracted from this excerpt will contain two nodes (**Emma** and **Mr. Woodhouse**) and one conversational link between the two nodes.

> "Especially when one of those two is such a fanciful, troublesome creature!" said **Emma** playfully. "That is what you have in your head, I know – and what you would certainly say if my father were not by."
>
> "I believe it is very true, my dear, indeed," said **Mr. Woodhouse**, with a sigh. "I am afraid I am sometimes very fanciful and troublesome."

Conversations are defined as contiguous spans of dialogues. Dialogues are spans of text spoken by characters that are orthographically expressed using quotation marks. Elson *et al.* [2010] first use regular expressions for detecting dialogues. They then utilize their quoted speech attribution system [Elson and McKeown, 2010] for assigning speakers to dialogues. Finally, they connect characters that exchange dialogues to obtain a conversational network. We refer to their system as CINNET. Note that any conversations or interactions that are not expressed using a dialogue structure are not captured in a conversational network. For example, Elson *et al.* [2010]'s system will not extract conversational links from the following text:

(38) [Mr. Elton] was speaking with animation, [Harriet] listening with a very pleased attention; and [Emma], having sent the child on, was beginning to think how she

> might draw back a little more, when they both looked around, and she was obliged to join them.

SINNET, in contrast, extracts interaction links from text that may not have a dialogue structure. For example, SINNET extracts interaction links between all three entities (**Mr. Elton**, **Harriet**, and **Emma**) in the aforementioned sentence 38. Furthermore, SINNET not only extracts conversational interactions, but also other types of interactions that may not be conversational, for example, *having dinner* with someone or *dancing* with someone. Finally, SINNET not only extracts interactions but also observations, for example, some one *talking about* another person.

Note that SINNET will be unable to extract interaction links from a certain category of conversations that are expressed using dialogue structure. These are conversations in which the two conversing entities are not mentioned (as named mentions) in the same sentence or dialogue. Following is an example:

> "Poor Miss Taylor!–I wish she were here again. What a pity it is that Mr. Weston ever thought of her!"

> "I cannot agree with you, papa; you know I cannot. . . ."

In the above conversation, unless our off-the-shelf named entity disambiguator is able to resolve **I** to **Emma** and **you** to **Mr. Woodhouse**, SINNET will not be able to extract an interaction link between the two entities.

In line with the terminology presented in Section 2.2, we refer to the networks in which all pairs of entities are mutually aware of one another and of their mutual awarenesses, as *interaction networks* (networks consisting of people and INR links between them). We refer to networks in which only one entity is cognitively aware of the other as *observation networks* (networks consisting of people and OBS links between them).

5.4 Evaluation of SINNET

SINNET is trained on news articles. In this section, we present an evaluation of SINNET

Novel Excerpt	# of char. pairs	# of links	
		CONV-GOLD	SOCEV-GOLD
Emma	91	10	40
Study in Scarlet	55	8	22
David Copperfield	120	10	32
Portrait of a Lady	55	6	18

Table 5.1: A comparison of the number of links in the two gold standards.

on the literary genre. Elson *et al.* [2010] introduced a gold standard for measuring the performance of CINNET. We refer to this gold standard as CONV-GOLD. This gold standard is not suitable for measuring the performance of SINNET because SINNET extracts a larger set of interactions (beyond conversations) and observations. Interactions and observations combined are social events. We therefore create another gold standard for evaluating SINNET. We refer to this gold standard as SOCEV-GOLD.

5.4.1 Gold standards: CONV-GOLD and SOCEV-GOLD

CONV-GOLD consists of excerpts from four novels: Jane Austen's *Emma*, Conan Doyle's *A Study in Scarlet*, Charles Dickens' *David Copperfield*, and Henry James' *The Portrait of a Lady*. Elson *et al.* [2010] enumerate all pairs of characters for each novel excerpt. If a novel features n characters, its corresponding list contains $\frac{n*(n-1)}{2}$ elements. For each pair of characters, annotators mark "1" if the characters *converse* (as defined in Section 5.3) and "0" otherwise. Annotators are asked to identify conversations framed with both direct (quoted) and indirect (unquoted) speech.

SINNET aims to extract the entire set of interactions and observations. For each pair of characters, we ask the annotators to mark "1" if the characters *observe* or *interact* and "0" otherwise.

Table 5.1 presents the number of character pairs in each novel excerpt, the number of character pairs that converse according to CONV-GOLD and the number of character pairs that observe or interact according to SOCEV-GOLD. For example, the excerpt from the

novel *Emma* has 91 character pairs (for 14 different characters). Only 10 out of 91 pairs of characters have a conversation. In contrast, 40 out of 91 character pairs either interact or observe one another. The difference in the number of links between Conv-Gold and SocEv-Gold suggests that conversations form only a fraction of all type of interactions and observations. Note that Conv-Gold is a proper subset of SocEv-Gold; anything that is a conversation is also an interaction.

5.4.2 Evaluation and Results

Table 5.2 presents the results for the performance of CINNET and SINNET on the two gold standards (Conv-Gold and SocEv-Gold). We report precision (P), recall (R), and F1-measure (F1). The results show, for example, the recall of CINNET on Conv-Gold created from *Emma* is 0.4. The recall of SINNET on the same gold standard is 0.7. In general, the recall of SINNET is significantly higher than the recall of CINNET on Conv-Gold (columns 2 and 3). This suggests that most of the links expressed as quoted conversations are also expressed as interactions via reported speech. Note that, because SINNET extracts a larger set of interactions, we do not report the precision and F1-measure of SINNET on Conv-Gold. By definition, SINNET will predict links between characters that may not be linked in Conv-Gold; therefore the precision (and thus F1-measure) of SINNET will be low (and uninterpretable) on Conv-Gold.

Novel Excerpt	Conv-Gold		SocEv-Gold			
	CINNET	SINNET	CINNET	SINNET		
	R	R	P	P	R	F1
Emma	0.40	**0.70**	1.0	0.86	0.48	**0.61**
Study in Scarlet	0.50	0.50	1.0	0.69	0.41	**0.51**
David Copperfield	0.70	**0.80**	1.0	0.80	0.63	**0.70**
Portrait of a Lady	0.66	0.66	1.0	0.73	0.44	**0.55**
Micro-Average	0.56	**0.68**	1.0	0.79	0.50	**0.61**

Table 5.2: Performance of the two systems on the two gold standards.

Table 5.2 additionally presents the performance of the two systems on SocEv-Gold (the last four columns). These results show that CINNET achieves perfect precision. Since CINNET is not trained (or designed) to extract any interactions besides conversations, we do not present the recall of CINNET on SocEv-Gold.

5.4.3 Discussion of Results

If there are any conversational links that CINNET detects but SINNET misses, then the two systems may be treated as complementary. To determine whether or not this is the case, we count the number of links in all four excerpts that CINNET detects but SINNET misses. For Austen's *Emma*, SINNET misses two links that CINNET detects. For the other three novels, the counts of links that SINNET misses but CINNET captures are two, zero, and one, respectively. In total, SINNET misses five out of 112 links that CINNET captures. Since the precision of CINNET is perfect, it seems advantageous to combine the output of the two systems.

5.5 Expanded Set of Literary Hypotheses

In light of the analysis from the previous section, conversations form a minority of other types of interactions that appear in literary texts. We extend the set of hypotheses proposed by Elson *et al.* [2010] to utilize a broader class of interactions and observations. Following the approach of Elson *et al.* [2010], our hypotheses concern (a) the implications of an increase in the number of characters, and (b) the implications of the dichotomy between rural and urban settings. Our formulation of the literary hypotheses regarding the settings of novels differs from the formulation suggested by Elson *et al.* [2010]. Following are the set of hypotheses that Elson *et al.* [2010] invalidate (reformulated from the original formulation to show contrast with the way we formulate our hypotheses; EDM stands for Elson, Dames, and McKeown):

- EDM1: There is an inverse correlation between the number of dialogues and the number of characters.

- EDM2: In novels set in urban environments, numerous characters share little conversational interactions. Rural novels, on the other hand, have fewer characters with more conversations.

In addition to EDM1 and EDM2, we attempt to validate the following hypotheses in our work:

H0 : *As setting changes from rural to urban, there is no change in the number of characters.*

H1.1 : *There is a positive correlation between the number of interactions and the number of characters.*

H1.2 : *There is a negative correlation between the number of characters and the average number of characters each character interacts with.*

H2.1 : *As setting changes from rural to urban, there is no change in the total number of interactions that occur.*

H2.2 : *As setting changes from rural to urban, there is no change in the average number of characters each character interacts with.*

H3.1 : *There is a positive correlation between the number of observations and the number of characters.*

H3.2 : *There is a negative correlation between the number of characters and the average number of characters a character observes.*

H4.1 : *As setting changes from rural to urban, there is no change in the total number of observations that occur.*

H4.2 : *As setting changes from rural to urban, there is no change in the average number of characters each character observes.*

H5 : *As the number of characters increases, the number of communities increases, but the average size of communities decreases.*

H6 : *As setting changes from rural to urban, there is no change in the number nor the average size of communities.*

5.6 Methodology for Validating Hypotheses

Elson *et al.* [2010] provide evidence to invalidate EDM1. They report a positive Pearson's correlation coefficient (**PCC**) between the number of characters and the number of dialogues to show that the two quantities are not inversely correlated. We use the same methodology for examining our hypotheses related to the number of characters, namely hypotheses H1.1, H1.2, H3.1, H3.2, H5.

Elson *et al.* [2010] also provide evidence to invalidate EDM2. The authors extract various features from the social networks of rural and urban novels and show that these features are not statistically significantly different for the two groups under consideration, the rural and urban novels. They use the **homoscedastic t-test** to measure statistical significance (with $p < .05 \implies$ statistical significance). We employ the same methodology for examining our hypotheses related to the rural/urban dichotomy, namely hypotheses H2.1, H2.2, H4.1, H4.2, H6.

H0 : *As setting changes from rural to urban, there is no change in the number of characters.* In a network denoted by $G = (V, E)$, the number of characters is given by V.

H1.1 : *There is a positive correlation between the number of interactions and the number of characters.* For validating this hypothesis, we utilize the weighted interaction network denoted by $G^w_{INR} = (V, E)$. The number of interactions are given by the formula

$$\sum_{e \in E} w_e \tag{5.1}$$

where w_e is the weight of edge $e \in E$. The number of characters is simply $|V|$, where $|.|$ denotes the cardinality of a set.

H1.2 : *There is a negative correlation between the number of characters and the average number of characters each character interacts with.* For validating this hypothesis, we

utilize the unweighted interaction network denoted by G^u_{INR}. We use the following formula to calculate the average degree (or the number of other characters a character interacts with):

$$\frac{\Sigma_{v \in V}|E_v|}{|V|} = \frac{2|E|}{|V|} \tag{5.2}$$

where V, E denotes the vertices and edges in graph G^u_{INR} respectively, $|.|$ denotes the cardinality of a set, and E_v denotes the edges incident on any vertex $v \in V$.

H2.1 : *As setting changes from rural to urban, there is no change in the total number of interactions that occur.* Similar to H1.1, for validating this hypothesis, we utilize the weighted interaction network denoted by G^w_{INR}. We use formula 5.1 for calculating the total number of interactions .

H2.2 : *As setting changes from rural to urban, there is no change in the average number of characters each character interacts with.* Similar to H1.2, for validating this hypothesis, we utilize the unweighted interaction network denoted by G^u_{INR}. We use formula 5.2 to calculate the average number of characters each character interacts with.

H3.1 : *There is a positive correlation between the number of observations and the number of characters.* For validating this hypothesis, we utilize the weighted observation network denoted by G^w_{OBS}. We use formula 5.1 for calculating the total number of observations.

H3.2 : *There is a negative correlation between the number of characters and the average number of characters a character observes.* For validating this hypothesis, we utilize the unweighted observation network denoted by G^u_{OBS}. We use formula 5.2 to calculate the average number of characters a character observes.

H4.1 : *As setting changes from rural to urban, there is no change in the total number of observations that occur.* Similar to H3.1, for validating this hypothesis, we utilize the weighted observation network denoted by G^w_{OBS}. We use formula 5.1 for calculating the total number of observations.

H4.2 : *As setting changes from rural to urban, there is no change in the average number of characters each character observes.* Similar to H3.2, for validating this hypothesis, we

utilize the unweighted observation network denoted by G^u_{OBS}. We use formula 5.2 to calculate the average number of characters a character observes.

H5 : *As the number of characters increases, the number of communities increases, but the average size of communities decreases.* We use the algorithm proposed in Newman [2004] for finding communities. This algorithm finds a partition of the graph (not overlapping communities). The average size of communities is simply the sum of sizes of all communities divided by the number of communities. We experiment with both the interaction network and the observation network. The results are similar. We report the results for the interaction network.

H6 : *As setting changes from rural to urban, there is no change in the number nor the average size of communities.* Similar to H5, we report the results for the interaction network.

5.7 Results for Testing Literary Hypotheses

Table 10.6 presents the results for validating the set of hypotheses that we propose in this work (H0-H6). There are two broad categories of hypotheses: (1) ones that comment on various SNA metrics based on the increase in the number of characters (columns 3 and 4), and (2) ones that comment on various SNA metrics based on the type of setting (rural versus urban, columns 5 and 6). As an example, hypothesis H0 is to be read as: *As settings go from rural to urban ... the number of characters does not change significantly.* Grayed out boxes are not valid hypotheses. For example, *As # of characters* $\uparrow$ ..... *# of characters* $\sim$ is not a valid hypothesis.

The results show, for example, that as settings change from rural to urban, there is no significant change in the number of characters (row H0, last two columns). We say there is no significant change because $p > 0.05$. Similarly, for all other hypotheses in this category (H2.1, H2.2, H4.1, H4.2, and H6), the relation between the number of characters and the setting of novels behaves as expected in terms of various types of networks and social network analysis metrics.

	Hypothesis	As # of characters ↑ ...		As settings go from rural to urban ...	
#		PCC	Valid?	t-test	Valid?
H0	... # of characters ∼			$p > 0.05$	✓
H1.1	... # of interactions ↑	0.83	✓		
H1.2	... # of characters interacted with ↓	-0.36	✓		
H2.1	... # of interactions ∼			$p > 0.05$	✓
H2.2	...# of characters interacted with ∼			$p > 0.05$	✓
H3.1	... # of observations ↑	0.77	✓		
H3.2	... # of characters observed ↓	-0.36	✓		
H4.1	... # of observations ∼			$p > 0.05$	✓
H4.2	... # of characters observed ∼			$p > 0.05$	✓
H5	... # of communities ↑	0.98	✓		
H5	...average size of communities ↓	-0.26	✓		
H6	...# of communities ∼			$p > 0.05$	✓
H6	... average size of communities ∼			$p > 0.05$	✓

Table 5.3: Hypotheses and results. All correlations are statistically significant. ∼ is to be read as *does not change significantly*. As an example, hypothesis H0 is to be read as: *As settings go from rural to urban ... the number of characters does not change significantly.* Grayed out boxes are not valid hypotheses. For example, *As # of characters ↑ ..... # of characters ∼* is not a valid hypothesis.

The results also show that as the number of characters increases, the number of interactions also increases with a high Pearson correlation coefficient of 0.83 (row H1.1, column PCC). Similarly, for all other hypotheses in this category (H1.2, H3.1, H3.2, and H5), the relation between the number of characters and the setting of novels behaves as expected in terms of various types of networks and social network analysis metrics.

Our results thus provide support for the cogency of our interpretation of the original theories. These results highlight one of the critical findings of our work: while network metrics are significantly correlated with the number of characters, there is **no correlation at all between setting and number of characters** (hypothesis H0 is valid). So if the number of characters did change significantly from a rural to an urban setting, we may also have seen changes in the social structures.

5.8 Conclusion and Future Work

In this chapter, we investigated whether social network extraction confirms long-standing assumptions about the social worlds of nineteenth-century British novels. We set out to verify whether the social networks of novels explicitly located in urban settings exhibit structural differences from those of rural novels. Elson *et al.* [2010] had previously proposed a hypothesis of difference as an interpretation of several literary theories, and provided evidence to invalidate this hypothesis on the basis of conversational networks. Following a closer reading of the theories cited by Elson *et al.* [2010], we suggested that their results, instead of invalidating the theories, actually support their cogency. To extend Elson *et al.* [2010]'s findings with a more comprehensive look at social interactions, we explored the application of another methodology for extracting social networks from text (called SINNET) which had previously not been applied to fiction. Using this methodology, we were able to extract a rich set of observation and interaction relations from novels, enabling us to build meaningfully on previous work. We found that the rural/urban distinction proposed by Elson *et al.* [2010] indeed has no effect on the structure of the social networks, while the number of characters does.

As our findings support our literary hypothesis that the urban novels within Elson *et al.* [2010]'s original corpus do not belong to a fundamentally separate class of novels, insofar as the essential experience of the characters is concerned, possible directions for future research include expanding our corpus in order to identify novelistic features that do determine social worlds. We are particularly interested in studying novels which exhibit innovations in narrative technique, or which occur historically in and around periods of technological innovation.

Lastly, we would like to add a temporal dimension to our social network extraction, in order to capture information about how networks transform throughout different novels.

Part III

Extracting Networks from Emails

The first part of this thesis introduced a novel methodology for extracting social networks from raw text. This part of the thesis introduces a novel technique for extracting social networks from electronic mails (emails). Emails, unlike raw text, have a structure; they contain meta-data information (that is well structured with fields such as *to*, *from*, *cc*, *subject*) and content (that is largely unstructured). By utilizing the well structured meta-information, specifically the fields *to*, *from*, *cc*, and *bcc*, one can easily create a social network of "who sends emails to whom." However, there is a rich social network in the unstructured content of emails; people *talk about* other people in the content of emails. By virtue of talking about other people, there is a social event directed from the sender to the mentioned person (and from the recipients to the mentioned person once the email is read or replied to). To extract these "who talks about whom" links, we must first resolve the people being *talked about* to real people. For example, in an email from **Marie Heard** to **Sara Shackleton** that *mentions* a person named *Jeff*, we must first determine the referent of this mention. After all, there may be hundreds of people with *Jeff* as their first name (as is the case in the Enron email corpus). The problem of extracting social networks from emails thus poses a new challenge – we need a mechanism to disambiguate entities mentioned in the content of emails to real people in the network. In monolithic, coherent bodies of text, such as novels, it is unlikely that two different characters are referred using the same name. In organizational emails, however, this phenomenon is common. An organization may have hundreds of people with *Jeff* as their first name who are referred as *Jeff* in several emails.

In this part of the thesis, we introduce a novel technique for disambiguating named mentions to real people in an email network. We use this technique for extracting what we call the *mention* network. Since the sender is *talking about* the mentioned person, by definition, a mention link has the same meaning as a social event of type observation. We use the mention network for predicting organizational dominance relations between employees of the Enron corporation. Our experiments show that by utilizing the mention network, we are better able to predict the dominance relations between pairs of employees.

This part of the thesis is organized as follows: Chapter 6 introduces the terminology regarding emails, their structure, and the problem definition, Chapter 7 presents our unsupervised approach to resolve named mentions to real people, and Chapter 8 uses these

extracted networks for predicting the organizational dominance relations between employees of the Enron corporation.

Chapter 6

Introduction

In this chapter, we introduce the terminology regarding emails followed by the task definition and literature survey on resolving named mentions (in the content of emails) to entities in the corpus.

6.1 Terminology

The term *email* stands for electronic mail. Emails have several *meta-data* fields, *content*, and *attachments*. Meta-data fields specify the sender, the recipients, the subject line, and several other attributes associated with emails. The content refers to the text of the message sent by a sender of an email to its recipients. The content often contains references to other entities, referred to as *mentioned* entities. Attachments are files that are sent along with the email message. We do not utilize attachments for any purpose in our current work.

We work with the same definition of *entity* and *entity mention* as defined in Section 2.1 and as used in the first part of this thesis. We repeat the definition here for convenience. According to the ACE Entity annotation guidelines:[1]

> An entity is an object or set of objects in the world. A mention is a reference to an entity. Entities may be referenced in a text by their name, indicated by a common noun or noun phrase, or represented by a pronoun. For example, the

[1] http://nlp.cs.rpi.edu/kbp/2014/aceentity.pdf

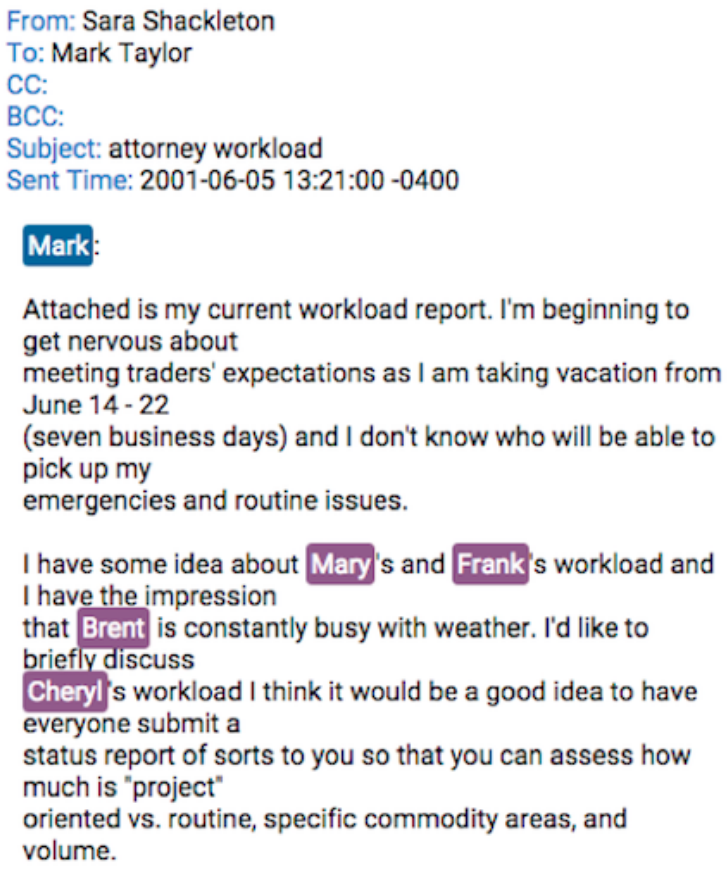
following are several mentions of a single entity:

Name Mention: Joe Smith

Nominal Mention: the guy wearing a blue shirt

Pronoun Mentions: he, him

ACE defines seven broad categories of entities but we are only concerned with the entities of type PERSON.


From: Sara Shackleton
To: Mark Taylor
CC:
BCC:
Subject: attorney workload
Sent Time: 2001-06-05 13:21:00 -0400

Mark:

Attached is my current workload report. I'm beginning to get nervous about
meeting traders' expectations as I am taking vacation from June 14 - 22
(seven business days) and I don't know who will be able to pick up my
emergencies and routine issues.

I have some idea about Mary's and Frank's workload and I have the impression
that Brent is constantly busy with weather. I'd like to briefly discuss
Cheryl's workload I think it would be a good idea to have everyone submit a
status report of sorts to you so that you can assess how much is "project"
oriented vs. routine, specific commodity areas, and volume.

Figure 6.1: A sample email from the Enron email corpus. The email is from **Sara Shackleton** to **Mark Taylor** regarding "attorney workload". The email contains first name references of five entities (all highlighted): *Mary*, *Frank*, *Brent*, and *Cheryl*.

Consider the email in Figure 6.1. The email has two parts: meta-data fields and content. The figure displays the following meta-data fields: "From", "To", "CC", "BCC", "Subject",

and "Sent Time". The content of the email starts with "Mark:". All the named mentions are highlighted. The email contains first name references (or named mentions) of five entities: *Mary*, *Frank*, *Brent*, and *Cheryl*. These named mentions refer to entities **Mary Cook**, **Frank Sayre**, **Brent Hendry**, and **Cheryl Nelson** respectively.

We define two types of networks:

- **Email network**: a network in which nodes are entities (people or groups of people) and links are directed connections from the sender to the recipients. When an entity sends an email to another entity, the sender entity has the recipient entity in its cognitive state and therefore these directed connections are OBS social events directed from the sender to the recipient.[2]

- **Mention network**: a network in which nodes are entities and links are directed connections from the sender to the mentioned entities. When an entity mentions or talks about another entity in the content of their email, the sender entity has the mentioned entity in its cognitive state and therefore these directed connections are OBS social events directed from the sender to the mentioned.

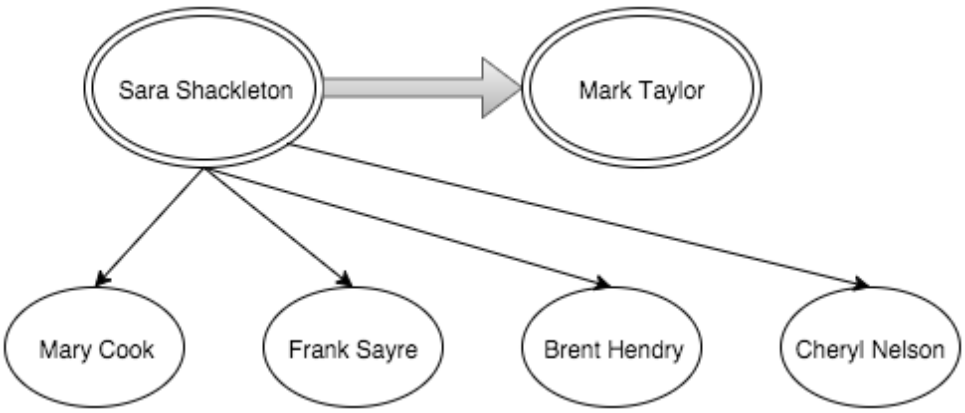


Figure 6.2: Email network (thick arrow) and mention network (thin arrows) for the sample email in Figure 6.1.

Figure 6.2 presents the email and mention network for the sample email shown in Figure 6.1. The email network for this email consists of two nodes (**Sara Shackleton** and

[2]Social events and their types (OBS and INR) are defined in Chapter 2.

Mark Taylor) and one directed link from **Sara Shackleton** to **Mark Taylor** (the thick arrow in Figure 6.2). The mention network consists of five nodes and four directed links, directed from **Sara Shackleton** to **Mary Cook**, **Frank Sayre**, **Brent Hendry**, and **Cheryl Nelson** (the thin arrows in Figure 6.2).

6.2 Task Definition

Constructing an email network from a corpus of emails is trivial – the sender and recipients for each email are present in the meta-data fields for that email. Since the meta-data fields are structured, extracting the sender and the recipients is easy. All one needs to do to create an email network is connect the sender to the recipients for each email.

Extracting a mention network, in contrast, is harder. What makes this problem hard is that an entity may be mentioned using his or her first name. More often than not, there are several people in an email corpus that have the same first name. For example, there are 180 entities with *Mary* as their first name in the Enron email corpus. To add a directed link from the sender to the mentioned entity, we first need to resolve the named mention to an entity. Differently put, for extracting a mention network, we need to accurately resolve the named entity mentions in the content of emails to entities in the network.

Given a named entity mention in an email, the task we address in this part of the thesis is to resolve this named entity mention to an entity.

6.3 Literature Survey

As Bhattacharya and Getoor [2007] note, the "entity resolution problem has been studied in many different areas under different names – co-reference resolution, de-duplication, object uncertainty, record linkage, reference reconciliation, etc." The techniques used for entity resolution are based on the general idea of gathering contextual cues around an entity mention and then utilizing these contextual cues for disambiguation. For example, Diehl *et al.* [2006] present a list of social and topical contextual cues:

- The participants in the conversation

- The larger group of entities known by the participants in the conversation and the types of relationships among them
- The individuals that the participants in the conversation have recently communicated with, either before or after the email was sent
- The topic of conversation in the email
- Recent topics of conversation among the participants and others outside the current conversation, either before or after the email was sent
- Cues contained within other emails in the thread
- Related name references within the current email
- Prior knowledge linking individuals to topics of conversation

Some of the popular techniques for utilizing contextual cues to disambiguate entities include scoring based [Fellegi and Sunter, 1969; Hernández and Stolfo, 1995; Winkler, 1999; Bekkerman and McCallum, 2005; Lee *et al.*, 2005; Diehl *et al.*, 2006; Kalashnikov and Mehrotra, 2006; On and Lee, 2007; Cucerzan, 2007; Han and Zhao, 2009], clustering based [Bekkerman *et al.*, 2005; Pedersen *et al.*, 2005; Yin *et al.*, 2007; Fan *et al.*, 2011], graphical modeling based [Minkov *et al.*, 2006; Chen and Martin, 2007], and other supervision based approaches [Mihalcea and Csomai, 2007; Kulkarni *et al.*, 2009; Dredze *et al.*, 2010; Chan and Roth, 2010; Ratinov *et al.*, 2011]. Out of these, the approach of Diehl *et al.* [2006] and Minkov *et al.* [2006] are closest to our domain – namely – resolving named entity mentions in the Enron email corpus.

Diehl *et al.* [2006] "introduce a class of scoring functions and explore the sensitivity of their performance along four dimensions." In this exploratory study, the authors use a set of 54 hand labeled examples to find "that by simply examining prior relationship strengths, as represented by the volume of communication," they "are able to successfully resolve a majority of first name references." The authors also find that by considering traffic only from the sender (and not from the sender and the recipients), their system achieves a better performance. Their approach is based on a strong assumption that at least the sender and the

mentioned entity communicate directly. We believe this is too strong an assumption. This is because, more often than not, the email datasets collected for regulatory and compliance purposes are not the entire collection of emails in an organization. The collection is usually a much smaller subset of the entire collection and thus emails between the sender and the true referent might actually be missing in the collection. We loosen this assumption by considering shortest paths (instead of direct connections) between the sender, the recipients, and the candidate referents.

Minkov *et al.* [2006] propose a lazy graph walk method with supervised re-ranking to resolve named mentions to nodes in the network. The authors apply their generic methodology to two tasks: email name disambiguation and email threading. Our algorithm is unsupervised and we compare our approach with the unsupervised part of Minkov *et al.* [2006]'s approach (their approach without supervised re-ranking) and show that our algorithm outperforms their algorithm by a large and statistically significant margin.

6.4 Conclusion

In this chapter, we introduced terminology regarding emails. We will use this terminology in the next two chapters. We also provided a formal definition for the task of disambiguating named entity mentions in the content of emails to entities in the corpus. Lastly, we presented a discussion of existing literature on the task. We compare our technique with an existing technique in the next chapter.

Chapter 7

Machine Learning Approach

This chapter presents our approach for disambiguating named mentions in the content of emails to actual entities in the network. We evaluate our technique on a test set created from the freely available Enron email corpus. The work presented in this chapter was introduced in Agarwal *et al.* [2012].

This chapter is organized as follows: Section 7.1 provides a brief history of the Enron corporation and essential statistics about the email dataset collected after Enron's collapse. Section 7.2 presents our unsupervised learning approach. Section 7.3 presents our experiments and results. We conclude and provide future directions of research in Section 7.4.

7.1 Data

7.1.1 A Brief History of the Enron Corporation

The following article provides an excellent and succinct summary of the Enron corporation and the major events that led to Enron's decline.[1]

> As 2002 began, energy trader Enron Corp. found itself at the centre of one of corporate America's biggest scandals. In less than a year, Enron had gone from being considered one of the most innovative companies of the late 20th century to being deemed a byword for corruption and mismanagement.

[1] http://www.britannica.com/topic/Enron-What-Happened-1517868

Enron was formed in July 1985 when Texas-based Houston Natural Gas merged with InterNorth, a Nebraska-based natural gas company. In its first few years, the new company was simply a natural gas provider, but by 1989 it had begun trading natural gas commodities, and in 1994 it began trading electricity.

The company introduced a number of revolutionary changes to energy trading, abetted by the changing nature of the energy markets, which were being deregulated in the 1990s and thus opening the door for new power traders and suppliers. Enron tailored electricity and natural gas contracts to reflect the cost of delivery to a specific destination – creating in essence, for the first time, a nationwide (and ultimately global) energy-trading network. In 1999 the company launched Enron Online, an Internet-based system, and by 2001 it was executing on-line trades worth about $2.5 billion a day.

By century's end Enron had become one of the most successful companies in the world, having posted a 57% increase in sales between 1996 and 2000. At its peak the company controlled more than 25% of the "over the counter" energy-trading market – that is, trades conducted party-to-party rather than over an exchange, such as the New York Mercantile Exchange. Enron shares hit a 52-week high of $84.87 per share in the last week of 2000.

Much of Enron's balance sheet, however, did not make sense to analysts. By the late 1990s, Enron had begun shuffling much of its debt obligations into offshore partnerships – many created by Chief Financial Officer Andrew Fastow. At the same time, the company was reporting inaccurate trading revenues. Some of the schemes traders used included serving as a middleman on a contract trade, linking up a buyer and a seller for a future contract, and then booking the entire sale as Enron revenue. Enron was also using its partnerships to sell contracts back and forth to itself and booking revenue each time.

In February 2001 Jeffrey Skilling, the president and chief operating officer, took over as Enron's chief executive officer, while former CEO Kenneth Lay stayed on as chairman. In August, however, Skilling abruptly resigned, and Lay resumed

the CEO role. By this point Lay had received an anonymous memo from Sherron Watkins, an Enron vice president who had become worried about the Fastow partnerships and who warned of possible accounting scandals.

As rumours about Enron's troubles abounded, the firm shocked investors on October 16 when it announced that it was going to post a $638 million loss for the third quarter and take a $1.2 billion reduction in shareholder equity owing in part to Fastow's partnerships. At the same time, some officials at Arthur Andersen LLP, Enron's accountant, began shredding documents related to Enron audits.

By October 22 the Securities and Exchange Commission had begun an inquiry into Enron and the partnerships; a week later the inquiry had become a full investigation. Fastow was forced out, while Lay began calling government officials, including Federal Reserve Chairman Alan Greenspan, Treasury Secretary Paul O'Neill, and Commerce Secretary Donald Evans. In some cases, officials said, Lay was simply informing them of Enron's troubles, but Lay reportedly asked for Evans to intervene with Moody's Investors Service, which was considering downgrading Enron bonds to noninvestment-grade status. Evans declined.

On November 8 Enron revised its financial statements for the previous five years, acknowledging that instead of taking profits, it actually had posted $586 million in losses. Its stock value began to crater – it fell below $1 per share by the end of November and was delisted on Jan. 16, 2002.

On Nov. 9, 2001, rival energy trader Dynegy Inc. said it would purchase the company for $8 billion in stock. By the end of the month, however, Dynegy had backed out of the deal, citing Enron's downgrade to "junk bond" status and continuing financial irregularities – Enron had just disclosed that it was trying to restructure a $690 million obligation, for which it was running the risk of defaulting.

On December 2 Enron, which a year before had been touted as the seventh largest company in the U.S., filed for Chapter 11 bankruptcy protection and

> sued Dynegy for wrongful termination of the failed acquisition. A month later Lay resigned, and the White House announced that the Department of Justice had begun a criminal investigation of Enron.
>
> By mid-2002 the once-mighty company was in tatters. Enron's energy-trading business had been sold off to the European bank UBS Warburg in January. Throughout the spring top Enron officials were subpoenaed to testify before congressional hearings. The majority of Enron's employees were unemployed, and their stock plans had become almost worthless. In June Arthur Anderson was convicted in federal court of obstruction of justice, while many other American companies scrambled to reexamine or explain their own accounting practices. As investigations continued into Enron's financial dealings, government connections, and possible involvement in California's energy problems, it appeared likely that the political and economic fallout would be making headlines for some time.

7.1.2 The Enron Email Corpus

After Enron's decline, the Federal Energy Regulation Commission (FERC) made available the messages belonging to 158 Enron employees. Klimt and Yang [2004] cleaned the dataset and provided a usable version of the dataset for the research community. Since then, the dataset has been used for a variety of natural language processing (NLP) and social network analysis (SNA) applications.

Klimt and Yang [2004] report a total of 619,446 emails in the corpus. Yeh and Harnly [2006] pre-process the dataset by combining emails into threads and restoring some missing emails from their quoted form in other emails. They also co-reference multiple email addresses belonging to one employee and assign unique identifiers and names to employees. Therefore, each employee is associated with a set of email addresses and names. We use this pre-processed dataset for our experiments and study. Our corpus contains 279,844 email messages that belong to 93,421 unique email addresses.

7.2 Name Disambiguation Approach

Our solution is based on the following intuition: if a sender mentions an entity to the recipient using the entity's first name, both the sender and the recipient are likely to know the mentioned entity. If both the sender and the recipient know the mentioned entity, they might have communicated with the true referent of the mentioned entity, either directly or indirectly. If the sender mentions an entity that he or she believes the recipient does not know, the sender is likely to use the full name of the mentioned entity. In this section we propose an unsupervised technique that uses this intuition to resolve named mentions in the content of emails to entities in the corpus. Before presenting our algorithms, we present some terminology that will be useful for describing our algorithms.

7.2.1 Terminology

Name Variants: We follow the methodology proposed by Minkov *et al.* [2006] for generating the set of name variants for a name. We denote the set of name variants for a name using notation NV_{name}. For example, for the name "Jeff Skilling", we generate the following set of name variants: $NV_{JeffSkilling}$ = {Jeff Skilling, Jeff, Skilling, JSkilling, J. S., Skilling Jeff}.

Candidates for a mention: We pre-calculate the name variants for all the entities in the network. Given a mention, we first calculate its set of name variants. We define the set of candidates for a mention to be the entities whose set of name variants have a maximal intersection with the set of name variants of the mention. As an example, consider the set of entities and their name variants given in Table 7.1 (**Chris Walker**, **Chris Dalton**, **Chris Ruf**, **Chris Bray**). Given a named mention *Chris*, its set of name variants NV_{Chris} = {Chris}. Since the cardinality of the intersection of NV_{Chris} with each of the sets $NV_{ChrisWalker}$, $NV_{ChrisDalton}$, $NV_{ChrisRuf}$, and $NV_{ChrisBray}$ is 1, all the entities have the maximal intersection and are hence valid candidates for the mention *Chris*. In contrast, for the mention *Chris Walker*, there is only one candidate – the entity with name **Chris Walker**. This is because $|NV_{ChrisWalker} \cap NV_{nameOf(E1)}| = 6$ is higher than the intersection of $NV_{ChrisWalker}$ with the other sets of name variants; $|NV_{ChrisWalker} \cap NV_{nameOf(Ei)}| = 1$, for $i \in \{2, 3, 4\}$.

Entity Id	Name	Name variants
E1	Chris Walker	Chris Walker, Chris, Walker, CWalker, C.W., Walker Chris
E2	Chris Dalton	Chris Dalton, Chris, Dalton, CDalton, C.D., Dalton Chris
E3	Chris Ruf	Chris Ruf, Chris, Ruf, CRuf, C.S., Ruf Chris
E4	Chris Bray	Chris Bray, Chris, Bray, CBray, C.S., Schidler Chris

Table 7.1: A table showing the full names of four entities and their name variants. All four entities have the same first name, *Chris*. Each entity is assigned an identifier ranging from E1 through E4.

7.2.2 Candidate Set Generation Algorithm

Algorithm 1 presents the pseudocode for our candidate generation algorithm. The algorithm requires two inputs: (1) the mention for which we wish to obtain the set of potential candidates and (2) a pre-computed input called the name variant map that maps name variants to the set of entities that contain those name variants. We denote this map using the notation Map_{nv}\{name variant, set of entity ids\}). The candidate set generation algorithm returns the set of candidates for the input mention.

For pre-computing Map_{nv}, we consider the name variants for all the entities in the corpus. Then, for each name variant (say nv), we collect the identifiers for all the entities that have nv as one of their name variants. For example, the name variant *Jeff* belongs to the set of name variants of both the entities, **Jeff Skilling** and **Jeff Donahue**. Therefore, one of the entries in Map_{nv} will be {Jeff, {entity id for **Jeff Skilling**, entity id for **Jeff Donahue**}}. Table 7.2 presents this map for the entities in Table 7.1.

In the first step of the algorithm (line 1 of Algorithm 1), we first generate the set of name variants for the input entity mention (denoted by NV_m). We initialize a map called Map_e in line 2 of the algorithm. This is a map from an entity identifier to the number of name variants that this entity shares with the entity mention. In lines 3 - 13 of Algorithm 1, we populate this map. Finally, we return the set of entities that have the highest count in Map_e. These are the entities whose set of name variants have the greatest intersection with the set of name variants of the input mention. For example, for the input mention *Chris*

Name variant	Entities that contain the name variant
Chris	E1, E2, E3, E4
Walker	E1
Chris Walker	E1
Dalton	E2
...	...

Table 7.2: The name variant map Map_{nv} for the entities in Table 7.1.

and Map_{nv} as in Table 7.2, Map_e will contain the following entries: {E1→ 1, E2→ 1, E3→ 1, E4→ 1}. Since all the entities have the maximum count, the set of generated candidates will consist of all four entities. As another example, for the input mention *Chris Walker* and Map_{nv} as in Table 7.2, Map_e will contain the following entries: {E1→ 6, E2→ 1, E3→ 1, E4→ 1}. In this case, only one entity has the maximum count and so the algorithm will return only one candidate, namely E1.

Algorithm 1 GetSetOfCandidates(mention m, Map_{nv}\{name variant, set of entity ids\})

NV_m = GetNameVariants(m) ▷ as suggested by Minkov *et al.* [2006]
Initialize Map_e\{entity id, count\}
for each name variant $n \in NV_m$ **do**
    $E_{ids} = Map_{nv}$.get(n)
    for each $e_{id} \in E_{ids}$ **do**
        Initialize $count = 0$
        if Map_e.containsKey(e_{id}) **then**
            $count = Map_e$.get(e_{id})
        end if
        $count = count + 1$
        Add key value pair $\{e_{id}, count\}$ to Map_e
    end for
end for
Return all entity ids with the highest count in Map_e

Algorithm 2 NameDisambiguationAlgorithm(set of emails E, Map_{nv})

for each email $e \in E$ **do**
  M_e = ExtractEntityMentionsUsingStanfordNER(e)
  for $m \in M_e$ **do**
    C_m = GetSetOfCandidates(m, Map_{nv})
    if $C_m = \emptyset$ **then**
      //Mention cannot be resolved: no potential candidates found
      continue
    else if $|C_m| == 1$ **then**
      //Mention uniquely resolved to candidate C_m.get[0]
      continue
    else
      $W_m = min_{\{p_k \in C_m\}}[d(p_s, p_k) + \Sigma_r d(p_r, p_k)]$ ▷ W_m is the set of winning candidates.
      if $W_m = \emptyset$ **then**
        //Mention cannot be resolved: joint distance of candidates is infinite
        continue
      else if $|W_m| == 1$ **then**
        //Mention uniquely resolved to candidate W_m.get[0]
        continue
      else
        //Mention cannot be resolved: too many candidates at the same joint distance
        continue ▷ Future Work
      end if
    end if
  end for
end for

7.2.3 Our Name Disambiguation Algorithm

Algorithm 2 presents the pseudocode for our name disambiguation algorithm. For each email, we first extract all the entity mentions from its content. We use Stanford's named entity recognizer and coreference resolution tool [Finkel *et al.*, 2005; Lee *et al.*, 2011]. For each mention, we get the set of candidates by using Algorithm 1. It is possible that for a mention no candidate is generated. After all, the Enron email corpus is a small sample of all corporate emails of the Enron corporation. Furthermore, the named entity recognizer falsely detects strings such as "Amount Due Employee" as a named mention of a person. Such mentions do not have a candidate set. If the candidate set generates only one candidate, then we resolve the mention to that candidate. The mentions that generate only one candidate are usually full name mentions such as *Sara Shackleton*. If a mention generates multiple candidates (denoted by C_m), then we find the subset of candidates that minimize the following function:

$$min_{\{p_k \in C_m\}}[d(p_s, p_k) + \Sigma_r d(p_r, p_k)]$$

Here, p_s denotes the sender, p_r denotes the recipient (an email can have multiple recipients), p_k denotes a candidate, and $d(.,.)$ denotes a distance function that measures the shortest path distance between two nodes in the email network. We follow the convention that $d(p_1, p_2) = \infty$ if the two nodes are not connected. We refer to the set of entities that minimize the joint distance between the sender and the recipient with W_m (winning candidates). If we are unable to find any winning candidate (it is possible that all candidates are disconnected from the sender and the recipients), we report that this mention cannot be resolved. If we find one winning candidate, we resolve the mention to that candidate. For handling the situation where we find multiple winning candidates, we need to utilize other contextual clues such as other people mentioned in the email, topical context, etc. We leave this work for the future.

7.3 Experiments and Results

7.3.1 Evaluation Set for Name Disambiguation

Minkov *et al.* [2006] note that:

> Unfortunately, building a corpus for evaluating this [name disambiguation] task is non-trivial, because (if trivial cases are eliminated) determining a name's referent is often non-trivial for a human other than the intended recipient.

Minkov *et al.* [2006] propose the following heuristic for creating a test set for the task:

> We collected name mentions which correspond uniquely to names that are in the email "Cc" header line; then, to simulate a non-trivial matching task, we eliminate the collected person name from the email header.

For evaluating our name disambiguation algorithm, we construct a test set using the heuristic suggested by Minkov *et al.* [2006]: we assume that if the name of the mentioned entity *matches* the name of one of the recipients, then that recipient is the true referent for the mentioned entity. For example, in the email in Figure 7.3, the entity **Chris Barbe** is one of the recipients. The email mentions *Chris* and since *Chris* matches with the name of **Chris Barbe**, we assume that the true referent for the mention *Chris* is **Chris Barbe**. At the time of evaluation, we do not use the fact that **Chris Barbe** is one of the recipients. We attempt to resolve *Chris* to one of the hundreds of people with *Chris* as their first name. We say our name disambiguation algorithm makes a correct prediction if *Chris* is resolved to **Chris Barbe**, and an incorrect prediction if *Chris* is resolved to any entity other than **Chris Barbe**.

The email mentions another entity *Liz* but since none of the recipients' name matches *Liz*, we do not know who *Liz* refers to, and therefore we do not add this mention to our evaluation set.

We say that a mention matches one of the recipients if the intersection of the set of candidates for the mention with the set of recipients is one. The mention *Chris* has hundreds of candidates but only one of those candidates is a recipient of the email under consideration (Table 7.3).

Using this heuristic on our email corpus, we are able to construct an evaluation set of 2,809 mentions. This means that we are able to find 2,809 named mentions in the content of emails whose name matches with the name of exactly one of the recipients on those emails.

From: Jeff Gobbell
To: Cindy Knapp, Tom Martin, Chris Barbe
... Chris, do you know Liz's (Cy Creek) email? ...

Table 7.3: An email from **jgobbel@flash.net** to **Cindy Knapp**, **Tom Martin**, and **Chris Barbe**. The content of the email mentions *Chris*, whose true referent is one of the recipients, **Chris Barbe**.

7.3.2 Experiments and Results

Table 7.4 presents the results for our name disambiguation approach in comparison with other baselines. Our name disambiguation algorithm achieves an accuracy of 69.7% on the test set of 2,809 mentions. This accuracy is significantly higher (using McNemar's significance test with $p < 0.05$) than the accuracy achieved by Minkov *et al.* [2006] – 62.3% on the same test set. We also report the performance of two simple variations of our name disambiguation algorithm: *B-Sender* and *B-Recipient*, in which we minimize the distance *only* from the sender and *only* from the recipients respectively. *B-Sender* achieves an accuracy of 60.4% and *B-Recipient* achieves an accuracy of 55.5%. Therefore, the intuition of minimizing the joint distance from both the sender and the recipients holds.

Approach	# of mentions (size of test set)	% accuracy
Minkov *et al.* [2006]	2,809	62.3%
Baseline B-Sender	2,809	60.4%
Baseline B-Recipient	2,809	55.5%
Our name disambiguation algorithm	2,809	69.7%

Table 7.4: A comparison of performance of name disambiguation techniques.

We use our best performing method to resolve the remaining 64,594 mentions in the entire Enron corpus. Our method is able to resolve 37,075 mentions, out of which 11,813 are unambiguous names (line 9 of Algorithm 2), while 25,262 are ambiguous and require the minimization of the joint distance (line 17 of Algorithm 2). Our method is unable to resolve 27,519 mentions (64,594 - 37,075), out of which 21,732 have multiple candidates at

the minimum joint distance (line 20 of Algorithm 2) and 5,658 have no candidates (line 6 of Algorithm 2). As alluded to in the previous paragraphs, over 40% of the errors due to multiple candidates at the same distance are caused by entity normalization. Those mentions for which there are no candidates are usually mentions that the named entity recognizer detects by mistake. A few examples are: *sorry* (this mention appears 1428 times), *matthew sorry* (964 times), *variances* (758 times), *thanks* (730 times), *regards* (728), etc. There are also mentions of celebrities for whom we cannot find any candidates such as **Dick Cheney**, **George Bush**, etc. Out of 27,519 mentions, 129 mentions have candidates that do not have paths from the sender and recipient and thus the joint distance of these candidates is infinity (line 14 of Algorithm 2).

7.3.3 Examples of the Name Disambiguation Algorithm in Action

We present three examples that illustrate the complexity of the task and the types of mistakes our algorithm commits. Figure 7.1 presents an example in which the name disambiguation algorithm makes the correct prediction. Figure 7.2 presents an example in which the name disambiguation algorithm makes an incorrect prediction. Figure 7.3 presents an example in which the name disambiguation algorithm is unable to make a prediction because it finds many candidates at the same distance. Table 7.5 presents the legend for the shapes and colors used in these figures.[2] Each graph shows the shortest paths of the top n candidates for a mention from the sender and recipients (of the email containing the mention).

Figure 7.1 shows the shortest paths of the top three candidates for the mention *Chris* from the sender, **Jeff Gobbell**, and the recipients, **Tom Martin** and **Cindy Knapp**. The three candidates are **Chris Barbe**, **Chris Stokley**, and **Chris Gaskill**. Based on the way we create the evaluation set (Section 7.3.1), we know that the mention *Chris* refers to the entity **Chris Barbe**. The length of the shortest path from the sender, **Jeff Gobbell** to **Chris Barbe** is 2 (**Jeff Gobbell** → **Cindy Knapp** → **Chris Barbe**). The length of the shortest path from **Cindy Knapp** to **Chris Barbe** is 1 and the length of the shortest path from **Tom Martin** to **Chris Barbe** is 3. Therefore, the joint distance of Chris Barbe from the sender and the recipients is 6 (2 + 1 + 3). The other two candidates are at a greater

[2]Shapes and colors are redundant i.e. each shape has a unique color.

Color and Shape	What is represented by the shape and color
Blue rectangle	Sender of an email.
Green parallelograms	Recipients of an email.
Red house	Gold entity.
Purple octagon	Top n candidates for a mention. The numbers in brackets represent the joint distance of a candidate from the sender and the recipients.
Purple triple octagon	Winning candidate predicted by our algorithm. Cases in which the winning candidate is the same as the gold entity (represented by a red house), we default to the red house.

Table 7.5: Legend for the graphs in Figure 7.1, 7.2, and 7.3.

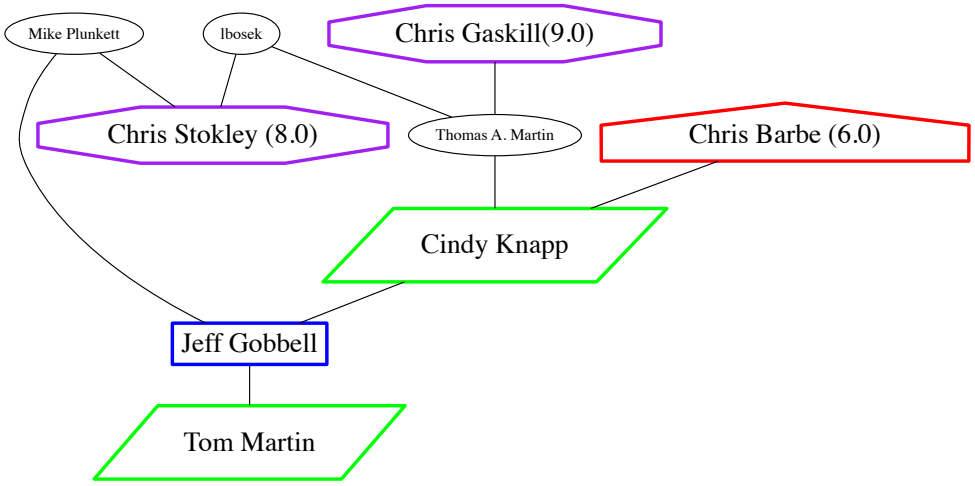


Figure 7.1: A graph showing the shortest paths of the top three candidates for the mention *Chris* from the sender, **Jeff Gobbell**, and the recipients, **Tom Martin** and **Cindy Knapp**. See Table 7.5 for the legend. The three candidates are **Chris Barbe** (at a joint distance of 6 from the sender and the recipients), **Chris Stokley** (at a joint distance of 8), and **Chris Gaskill** (at a joint distance of 9). **Chris Barbe** is the correct prediction.

joint distance; **Chris Stokley** is at a joint distance of 8 and **Chris Gaskill** is at a joint distance of 9. Therefore, our name disambiguation algorithm predicts **Chris Barbe** to be the winning candidate, which is the correct prediction.

Figure 7.2 shows the shortest paths of the top five candidates for the mention *Gary* from the sender, **Clem Cernosek**. The five candidates are **Gary E. Anderson**, **Gary Hanks**, **Gary Smith**, **Gary Hickerson**, and **Gary Lamphier**. Based on the way we create the evaluation set (Section 7.3.1), we know that the mention *Gary* refers to the entity **Gary E. Anderson**. Our name disambiguation algorithm incorrectly predicts **Gary Hanks** to be the referent for *Gary*. This is because **Gary Hanks** is at a joint distance of 1, which is smaller than the joint distance of other candidates. Specifically, it is smaller than the joint distance of the correct entity **Gary E. Anderson**, which is at a joint distance of 2.

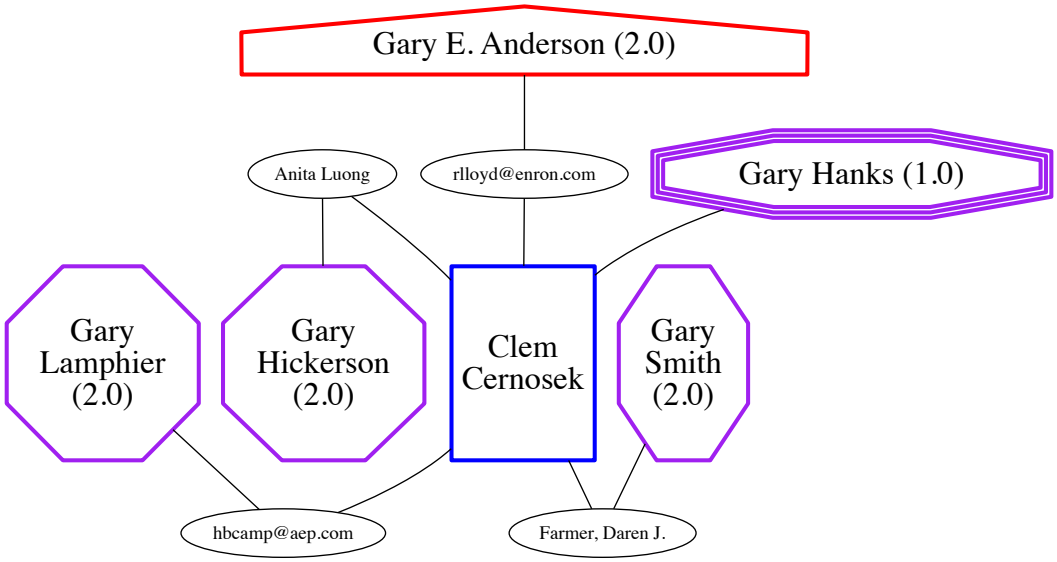


Figure 7.2: A graph showing the shortest paths of the top five candidates for the mention *Gary* from the sender, **Clem Cernosek**. The five candidates are **Gary Hanks**, **Gary E. Anderson**, **Gary Lamphier**, **Gary Hickerson**, and **Gary Smith**. The mention *Gary* refers to the entity **Gary E. Anderson**, who is at a joint distance of 2 from the sender. Our name disambiguation algorithm makes an incorrect prediction. It predicts **Gary Hanks** to be the referent who is at a shorter joint distance of 1.

Figure 7.3 shows an example in which we are unable to make a prediction because there are two winning candidates at the same joint distance from the sender (**Jason Williams**) and the recipient (**Spiro Spirakis**). Once again, this is a hard example – the correct candidate is much further away from the sender and the recipient. The correct candidate is at a joint distance of 9. There are five other **Philips** at a shorter joint distance from the sender and the recipient.

7.3.4 Error Analysis

Error analysis reveals two main sources of errors: (1) ones in which we are able to resolve the mention to a candidate but make an incorrect prediction (denoted by IncorrectPrediction), and (2) ones in which we are unable to make a prediction because there are multiple candidates at the same joint distance (denoted by ManyCandidatesAtSameDistance). The first category of errors constitutes 5.6% of the total errors (total errors = 30.3%, see Table 7.4) and the second category of errors constitutes 92.6% of the total errors. An analysis of a random sample of 60 erroneous cases (30 from each of the two categories) reveals that one of the major sources of these errors is unclean data. We refer to this source of error as the entity normalization error. Out of the 30 cases in the IncorrectPrediction category, 15 (or 50%) are due to the entity normalization error. Out of the 30 cases in the ManyCandidatesAtSameDistance category, 13 (or 43.34%) are due to the entity normalization error. We explain the entity normalization error below.

An entity may be referenced in the corpus in multiple ways. For example, **Sara Shackleton** is referenced as *Sara Shackleton*, *Sara Shackelton* (different spelling), *sshackl*, and in several other ways. Furthermore, an entity may have different email addresses, and since an email address (if present) is a unique identifier for an entity, the same entity with different email addresses may appear as two different entities. The goal of entity normalization is to resolve such mentions to the same entity. Entity normalization is a hard problem and out of scope for this thesis. The reason why entity normalization leads to errors that fall in the IncorrectPrediction category is that our name disambiguation algorithm resolves a mentioned entity to, lets say **Sara Shackleton**, but the ground truth is **sshackl**. Given an entity normalization module, that specifies that the entities **Sara Shackleton** and **sshackl**

are really the same entities, our algorithm will make the correct prediction.

Figure 7.4 presents an example for the kinds of errors in the ManyCandidatesAtSameDistance category that are caused due to the entity normalization problem. In this example, there are three candidates: **joe.parks@enron.com** at a joint distance of 5 from the sender and the recipients, **joe.parks@bridgeline.net** also at a joint distance of 5, and **joe parks** at a joint distance of 7. The ground truth is **joe parks**. The sender is **knipe3**, the red node, and the recipients are **brian constantine**, **cmccomb**, **erik wollam**, and **keith mccomb**, the blue nodes. Clearly, if the three different ways of referring to **Joe Parks** is normalized to one entity, name disambiguation will make the correct prediction.

7.4 Conclusion and Future Work

In this chapter, we presented the details of our unsupervised name disambiguation algorithm. We showed that our algorithm outperforms the algorithm suggested by Minkov *et al.* [2006] by a large and significant margin. Using our name disambiguation method, we are able to extract 37,075 mention links from the entire Enron corpus. Technically speaking, these are observation (OBS) links from the sender to the mentioned person. However, it may be argued that these are also OBS links from the recipients to the mentioned person (since the reader has the mentioned person in their cognitive state while reading the email). In the next chapter, we experiment with several ways of creating a mention network and show their utility on an extrinsic and well-studied task of organizational dominance prediction of Enron employees.

Our results showed that over 92% of the errors were caused by multiple candidates at the same joint distance from the sender and the recipients. Using a sample of 30 such errors, we also showed that about 40% of these errors were caused due to entity normalization – when one entity is being referenced in the corpus in multiple ways. In the future, we would like to tackle the problem of entity normalization for improving the effectiveness of our name disambiguation approach. We would also like to experiment with other features such as recency and volume of communication for resolving ties between multiple candidates.

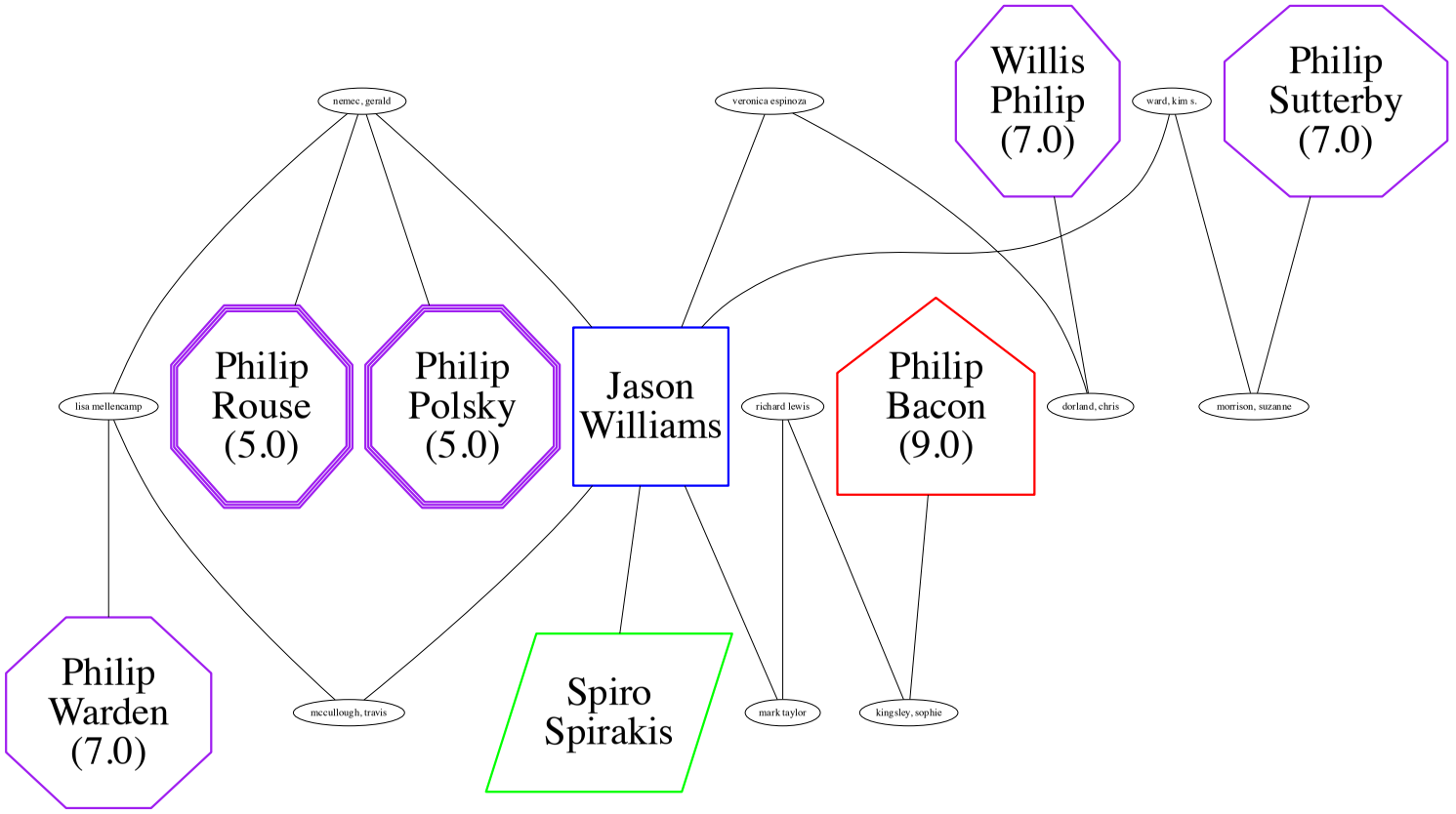


Figure 7.3: A graph showing the shortest paths of the top six candidates for the mention *Philip* from the sender, **Jason Williams**, and the recipient, **Spiro Spirakis**. The six candidates are **Philip Rouse**, **Philip Polsky**, **Philip Warden**, **Willis Philip**, **Philip Sutterby**, and **Philip Bacon**. The mention *Philip* refers to the entity **Philip Bacon**, who is at a distance 9 from the sender and the recipient. Our name disambiguation algorithm is unable to make a prediction because there are two candidates, **Philip Rouse** and **Philip Polsky**, at the same joint distance of 5.

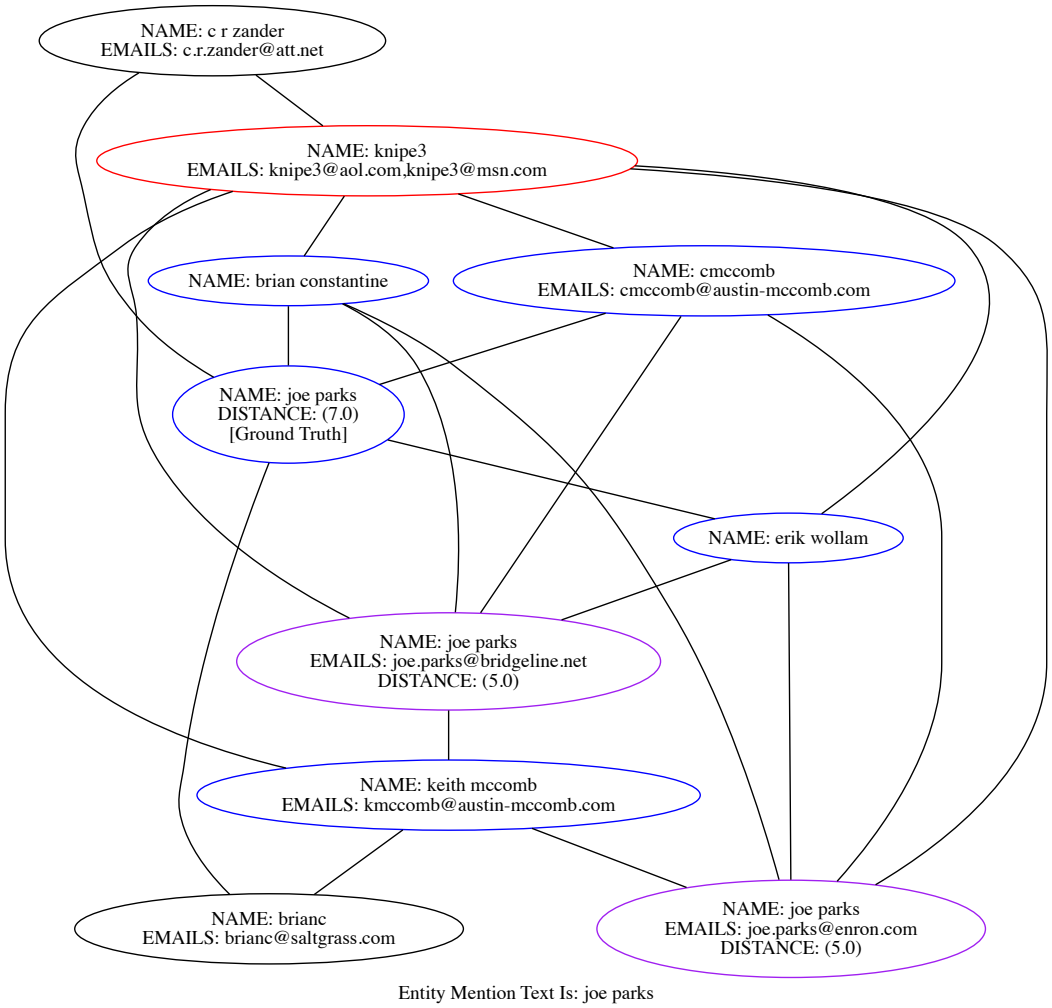


Figure 7.4: Name disambiguation error caused to do entity normalization error. There are three candidates: **joe.parks@enron.com** at a joint distance of 5 from the sender and the recipients, **joe.parks@bridgeline.net** also at a joint distance of 5, and **joe parks** at a joint distance of 7. The ground truth is **joe parks**. The sender is **knipe3**, the red node, and the recipients are **brian constantine**, **cmccomb**, **erik wollam**, and **keith mccomb**, the blue nodes. Clearly, if the three different ways of referring to **Joe Parks** is normalized to one entity, name disambiguation will make the correct prediction.

Chapter 8

Application: Predicting Organizational Dominance Relations

8.1 Introduction

In order to study the utility of the mention network, we use an extrinsic task: predicting organizational dominance relations between employees of the Enron corporation. The task of predicting dominance relation between pairs of employees has received much attention in the past [Rowe *et al.*, 2007; Diehl *et al.*, 2007; Creamer *et al.*, 2009; Bramsen *et al.*, 2011; Gilbert, 2012; Wang *et al.*, 2013; Prabhakaran and Rambow, 2014]. Given a pair of employees, the task is to predict whether or not one employee is higher up in the organizational hierarchy than the other. We note that this task has been referred to in the literature in various ways: predicting social power relations [Bramsen *et al.*, 2011], automatically extracting social hierarchy [Rowe *et al.*, 2007; Creamer *et al.*, 2009], predicting workplace hierarchy [Gilbert, 2012], predicting organization structure [Palus *et al.*, 2013], predicting hierarchical power [Prabhakaran and Rambow, 2014]. None of these works tackle the task of organizational hierarchy prediction. Predicting hierarchy has two sub-tasks: (1) predicting if two people are in the same managerial lineage and (2) predicting who is the boss of whom. To the best of our knowledge, the work to date (including ours) on the Enron email corpus tackles the second sub-task but not the first. The work presented in this chapter was introduced in Agarwal *et al.* [2012] and Agarwal *et al.* [2014d].

As mentioned earlier, the Enron email corpus is a small subset of all Enron emails. The corpus has all of the mailboxes of only 158 employees. We refer to this group of employees as the *public* group. Even though the corpus has mailboxes of only 158 employees, the corpus contains unique email addresses of 92,263 other entities that send or receive an email to or from this public group of employees. We refer to this group of employees as the *private* group. As one can imagine, we do not have all the communications of the employees in this private group. But can we extract missing links (email or non-email) from the content of emails? After all, people talk about other people and their interactions with other people in the content of emails.

Before experimenting with the mention network, we develop a general technique for predicting dominance relations between employees. For evaluating our technique, we introduce the largest known gold standard for hierarchy prediction of Enron employees (hierarchy relationships contain more information than simply dominance relationships). Our technique for predicting dominance relations using a network is simple and unsupervised; we sort all entities in the network based on their degree centralities, a popular social network analysis (SNA) metric, and predict a dominance relation between two entities based on their relative ranks in the sorted list (higher degree centrality means a higher dominance). We refer to this technique as the SNA-Based approach. We compare our approach with the state-of-the-art NLP-Based approach due to Gilbert [2012]. As a byproduct of this comparison, we highlight a general limitation of NLP-Based approaches; NLP-Based approaches are restricted to making predictions only on entity pairs that exchange emails (because if two entities do not exchange emails, their word based feature vector will be empty). SNA-Based approaches, in contrast, are not limited by missing or non-existent communications between entities. In a practical scenario, we seldom have access to communications between all the entities in a collected corpus. This makes the limitation of NLP-Based systems a significant disadvantage. In fact, we show that the upper bound performance for a perfect NLP-Based approach on our gold standard is significantly lower than the performance of our SNA-Based approach. Leaving the issue of missing communications aside, we further show that even if we restrict ourselves to entity pairs that exchange emails, our SNA-Based approach outperforms the state-of-the-art NLP-Based approach.

Using the name disambiguation technique introduced in the previous chapter, we create a variety of mention networks and test whether or not the mention network is useful for predicting dominance relations between Enron employees. We explore several ways of constructing the mention network: weighted versus unweighted network, directed versus undirected network, and others as discussed in Section 8.6. Our experiments and results show that (1) the mention network, even though sparser than the email network, is a better predictor of dominance relation between Enron employees, (2) unweighted networks outperform weighted networks: having many different email correspondents is a better indicator of higher organizational status than writing or receiving many emails, and (3) in order to exploit the mention network, the recipient of the email must be linked to the mentioned person, and we must use out-degree: having many people mentioned to you is a better indicator of higher organizational status than mentioning many people or being mentioned a lot.

The rest of this chapter is organized as follows: Section 8.2 discusses related work on dominance prediction and organizational hierarchy prediction, Section 8.3 provides details of our data and gold standard, Section 8.4 presents our technique for predicting dominance relations between entity pairs, Section 8.5 provides details of two baseline approaches, Section 8.6 provides details of our experiments and results for utilizing the mention network for dominance prediction. We conclude and provide future directions of research in Section 8.7.

8.2 Related

Since its introduction , the Enron email corpus [Klimt and Yang, 2004] has been used as a development and test set for a wide variety of applications and studies: automatically finding organizational roles of people [Keila and Skillicorn, 2005; McCallum *et al.*, 2007], studying the correlation between the major events at Enron and the communication patterns of senior personnel [Diesner *et al.*, 2005], discovering important nodes through graph entropy [Shetty and Adibi, 2005], studying email formality in workplace [Peterson *et al.*, 2011], studying the correlation between gender and types of emotions expressed in emails [Mohammad and Yang, 2011a], identifying spam [Cormack and Lynam, 2005; Martin *et al.*, 2005; Hershkop,

2006], summarizing emails [Carenini *et al.*, 2007; Murray and Carenini, 2008; Zajic *et al.*, 2008], and studying organizational power relations [Rowe *et al.*, 2007; Diehl *et al.*, 2007; Creamer *et al.*, 2009; Bramsen *et al.*, 2011; Gilbert, 2012; Wang *et al.*, 2013; Prabhakaran and Rambow, 2014].

Rowe *et al.* [2007] present a social network analysis (SNA) based approach for predicting organizational hierarchy. For each person (or node in the undirected email network), the authors calculate a normalized *social score* $S \in [0, 100]$. The social score is a weighted linear combination of the following SNA features: number of emails, average response time, response score, number of cliques, raw clique score, weighted clique score, degree centrality, clustering coefficient, mean of shortest path length from a specific node to all nodes in the graph, betweenness centrality, hubs-and-authorities importance. Once every person is assigned a social score, Rowe *et al.* [2007] arrange the people in a hierarchy – people with high social scores are at the top and the people with low social scores are at the bottom. We use a similar strategy for predicting dominance relations between pairs of people. However, we use much simpler network analysis measure, namely degree centrality, for ranking people. One advantage of using a simple ranking measure is that the results are more interpretable. Furthermore, our goal is not design the best possible system for predicting dominance relations between employees. Our goal is to show the utility of the mention network. We are able to show its utility with a simpler measure like degree centrality. One of our main contributions to the community of researchers who are interested in building a state-of-the-art system for dominance prediction is our gold standard. We make available a gold standard that contains dominance relations between 1518 entities in the Enron corpus. Most of the related works have either not evaluated their technique, like Rowe *et al.* [2007], or they have done so on a fairly small test set that consists of less than 158 entities, like Bramsen *et al.* [2011] and Gilbert [2012].

There is work in the literature that primarily utilizes language to deduce dominance relations [Bramsen *et al.*, 2011; Gilbert, 2012; Prabhakaran and Rambow, 2014]. Bramsen *et al.* [2011] train an SVM classifier to classy emails into two categories: *UpSpeak* and *DownSpeak*. They define UpSpeak as "communication directed to someone with greater social authority." They define DownSpeak as "communication directed to someone with

less social authority." The authors use unigrams, bigrams, parts-of-speech unigrams and bigrams, and polite imperatives (like "Thanks") as features for the classifier. After pre-processing the Enron email dataset with significant constraints – they only consider emails with one sender and one recipient, both of whom belong to the set of 158 employees, plus they should have exchanged at least 500 words of communication – the authors train and test on only 142 emails that satisfy all these criteria. Furthermore, Bramsen *et al.* [2011] do not actually classify relations between people. They only classify if an email between two people is UpSpeak or DownSpeak. Also, their work (code and test data) belongs to a private company and is not available for benchmark testing.

Prabhakaran and Rambow [2014] predict dominance relations between people using email thread information. The authors employ a wide variety of NLP features, including features that "capture the structure of message exchanges without looking at the content of emails (e.g. how many emails did a person send)" and features that "capture the pragmatics of the dialog and require an analysis of the content of emails (e.g. did they issue any requests)." The authors report the results on a subset of the gold standard for dominance relations introduced in our previous work [Agarwal *et al.*, 2012]. The subset contains only those entity pairs that are part of an email thread. For each pair of entities that are part of a thread and whose dominance relation is in the gold standard, Prabhakaran and Rambow [2014] predict whether one dominates the other using the information only in that thread. Of course, for the same pair of entities, their system might predict conflicting dominance relations in different threads. Since their evaluation is quite different from ours – we predict a global (not a thread level) dominance relation – our results are not comparable.

To the best of our knowledge, the only other work – other than the work of Bramsen *et al.* [2011] and Prabhakaran and Rambow [2014] – in the computational linguistics community that is about predicting dominance relations of Enron employees is due to Gilbert [2012]. We provide a detailed explanation of their technique in Section 8.5.2 and present a comparison of our systems in Section 8.5.3.

8.3 A New Gold Standard for Enron Hierarchy Prediction

As discussed above, several researchers attempt to predict the dominance relations between Enron employees using the Enron email corpus. For evaluation, they use the set of job titles of 158 Enron employees assembled by Shetty and Adibi [2004]. There are two limitations of this gold standard:

1. The gold standard is small: it has dominance relations of only 158 entities.

2. It does not have hierarchy information: the gold standard merely states the organizational titles of entities (like CEO, Manager, etc.). It does not state whether or not two entities are professionally related.

We introduce a new gold standard for both dominance and hierarchy prediction of Enron employees [Agarwal *et al.*, 2012]. We construct the gold standard by studying the original Enron organizational charts. We discover these charts by performing a manual random survey of a few hundred emails. After finding a few documents with organizational charts, we search all the remaining emails for attachments of the same file type, and exhaustively examine the search results for additional organizational charts. We then manually transcribe the information contained in the organizational charts into a database.

Our resulting gold standard has a total of 1518 employees who are described as being in immediate dominance relations (manager-subordinate). There are 2155 immediate dominance relations spread over 65 levels of dominance (CEO, manager, trader, etc.).[1] From these relations, we form a transitive closure and obtain 13,724 dominance relations. For example, if A immediately dominates B and B immediately dominates C, then the set of valid organizational dominance relations are A dominates B, B dominates C and A dominates C. We link this representation of the hierarchy to the threaded Enron corpus created by Yeh and Harnly [2006].[2]

[1]Note that the number of immediate dominance relations can be more than the number of nodes. This is because the dominance relation chart is a directed acyclic graph (DAG) and not simply a tree. Consider the following DAG with five nodes but eight immediate dominance relations: A dominates B and C, B dominates C, D, and E, C dominates D and E, and D dominates E.

[2]Our database is freely available as a MongoDB database and may be downloaded from `http://www1.`

For ease of reference, we categorize Enron employees into two categories: public and private. The dataset consists of inboxes of 158 Enron employees. The dataset has a complete collection of emails sent and received by this set of 158 employees. Since the communication among this group is almost completely known, we call this set of people the **public** group. The second group of people is made up of all other people who are senders or recipients of emails to this public group. These are people for whom we have some email correspondence with people in the public group. However, we have almost no email correspondence *among* them: the only email between people in this group are emails involving at least one person in the public group as a joint recipient. Since most of the email correspondence among people in this group is hidden to us, we call this the **private** group. These two groups are disjoint and together, form the nodes in the email network. As expected, the email network for the public group is denser (density of 20.997%) as compared to the private group (density of 0.008%). Given this terminology, the tuples in the gold standard may be categorized into three categories: **public-tuples**, **private-tuples**, and **public-private-tuples**. Public-tuples are those in which both the entities belong to the public group (the set of 158 entities), the private-tuples are those in which both the entities belong to the private group, and the public-private-tuples are those in which one entity belongs to the public group and the other entity belongs to the private group.

8.4 Dominance Prediction Technique

Our algorithm for predicting the dominance relations using social network analysis metrics is simple and unsupervised. We calculate the degree centrality of every node (or employee) in a network (email or mention network), and then rank the nodes by their degree centrality.

Let $C_D(n)$ be the degree centrality of node n, and let DOM be the dominance relation (transitive, not symmetric) induced by the organizational hierarchy. We then simply assume that for two people p_1 and p_2, if $C_D(p_1) > C_D(p_2)$, then DOM(p_1,p_2). For every pair of people who are related with an organizational dominance relation in the gold standard, we then predict which person dominates the other. Note that we do not predict if two people are in

---

`ccls.columbia.edu/~rambow/enron/enron_database.tar` (293MB).

a dominance relation to begin with. The task of predicting if two people are in a dominance relation is different and we do not address that task in this thesis. Therefore, we restrict our evaluation to pairs of people (p_1, p_2) who are related hierarchically (i.e., either DOM(p_1,p_2) or DOM(p_2,p_1) in the gold standard). Since we only predict the directionality of the dominance relation of people given they are in a hierarchical relation, the random baseline for our task performs at 50%.

8.5 Baseline Approaches

We experiment with two baseline approaches: SNA-BASED and a state-of-the-art NLP-BASED approach by Gilbert [2012]. We discuss them in turn. Note that our gold standard is a list of 13,724 dominance pairs (or tuples). Given a tuple from the gold standard, we want an automated approach for predicting whether or not the first entity dominates the second entity.

8.5.1 Unsupervised SNA-BASED Approach

We construct an undirected weighted network using email meta-data: nodes are people who are connected with weighted links representing the volume of emails sent or received between each pair of nodes. We then use the dominance prediction technique described above to make dominance predictions about a given pair of people.

8.5.2 Supervised NLP-BASED Approach

Gilbert [2012] create a list of phrases[3] that they deem important for predicting whether or not an email message is *upward* or *not-upward*. The author defines *upward* as an email message where "every recipient outranks the sender." and *not-upward* as an email message where "every recipient does not outrank the sender." They use the list of phrases as binary bag-of-words features for training an SVM and present three-fold cross-validation results. The SVM makes prediction at the message level i.e. all messages between two people are assigned one of two categories (upward/not-upward). Gilbert [2012] use voting to determine

[3]The list of phrases is freely available at `http://comp.social.gatech.edu/hier.phrases.txt`

the dominance relation between two people; if more number of messages from A to B are classified upward, then A dominates B. We follow the same approach for reporting results on our gold standard.

8.5.3 Experiments and Results

It is clear that current NLP-Based approaches can make dominance predictions only on pairs of people who exchange emails. Out of 13,724 pairs in our gold standard, only 2,640 pairs exchange emails. We refer to the gold set of 13,724 pairs as G and the subset of G in which pairs of people exchange emails as T. We report results on both these test sets.

Note that if we consider a perfect NLP-Based approach that makes a correct prediction on all the pairs in set T and randomly guesses the dominance relation of the remaining $11,084$ pairs in G, the system will achieve an accuracy of $(2640 + 11084/2)/13724 = 59.62\%$. We refer to this number as the upper bound of the best performing NLP-Based approach on our gold standard G.

Approach	Test set	# of test points	%Acc
NLP-Based [Gilbert, 2012]	T	2,640	82.37
SNA-Based	T	2,640	87.58
NLP-Based (upper bound)	G	13,724	59.62
SNA-Based	G	13,724	83.88

Table 8.1: Results of four experiments comparing the performance of purely NLP-based systems with simple SNA-based systems on two gold standards G and $T \in G$.

Table 8.1 presents the results for four experiments: {NLP-Based, SNA-Based} $\times$ {T, G}. As the results show (rows three and four), the SNA-Based approach outperforms the NLP-Based approach by a large and significant margin (83.88% versus 59.62%). Even if we restrict the test set to T (rows one and two), the SNA-Based approach outperforms the NLP-Based approach by a large and significant margin (87.58% versus 82.37%). This indicates that the dominance relation between people in our gold standard is better predicted using SNA statistics compared to more sophisticated supervised NLP methods.

8.6 Dominance Prediction Using the Mention Network

The previous section presents results for one possible configuration that may be used for predicting dominance relations using an SNA-Based approach (weighted, undirected, email-only network). However, there are several other possibilities: using a weighted versus unweighted network, using a directed versus undirected network, and using the email-only network versus the network that takes into account mention links. In this section, we explore a comprehensive set of possibilities that allows us to: (1) conclude that the mention network is a better predictor of dominance relations (compared to the traditionally used email network) and (2) discover an interesting characteristic of the Enron email corpus – "a person is a boss if other people get mentioned to him or her."

8.6.1 Set of Experiments

We experiment with a comprehensive set of parameter combinations. Following is a the set of parameters we consider:

1. Types of network: There are four basic networks we consider:

   (a) Email only (**E**)

   (b) Mention-only when we add a link between mentioned and recipient (**MR**)

   (c) Mention-only when we add a link between mentioned and sender (**MS**) and

   (d) Mention-only when we add a link between mentioned and both sender and recipient (**MSR**).

   We experiment with these networks alone (4 networks) and then combinations of email and the three types of mention networks, for a total of 7 networks.[4]

2. Weighted/Unweighted network: Networks may be weighted or unweighted. Weighted networks capture the volume of communication between people as well as the number of other people a person communicates with, whereas unweighted networks only capture the number of other people a person communicates with.

[4]The directionality (when directed) is always from the sender/recipient to the mentioned person.

System	Centrality	Network
Deg-MRU	Degree	Mention only network. Undirected links added between recipient and mentioned person.
Deg-EU	Degree	Email only network. Undirected links added between the sender and the receiver.
In-MRD	In-degree	Mention only network. Directed links added from the recipient to the mentioned person.
Out-MRD	Out-degree	Mention only network. Directed links added from the recipient to the mentioned person.
Deg-MSD	Degree	Mention only network. Directed links are added from the sender to the mentioned person.
Out-EDMRU	Out-degree	Network consisting of two types of links: directed email links added from the sender to the recipient, and undirected mention links added from the recipient to the mentioned person.

Table 8.2: Some examples of terminology used in this paper to refer to different types of systems.

3. Directed/Undirected network: Networks may either be directed (**D**) or undirected (**U**). When we combine a directed network with an undirected network, the resulting network is considered directed (an undirected link may be seen as two directed links.)

4. Centrality: We experiment with three different notions of centrality: In-degree (**In**), Out-degree (**Out**) and Degree (**Deg**). In undirected networks, all three notions are equivalent. In directed networks, degree of a node is the sum of its in-degree and out-degree.

Let $m \in M = \{\mathbf{In}, \mathbf{Out}, \mathbf{Deg}\}$ be the type of centrality and $t \in T = \{\phi, \mathbf{ED}, \mathbf{EU}\} \times \{\phi, \mathbf{MRD}, \mathbf{MRU}, \mathbf{MSD}, \mathbf{MSU}, \mathbf{MRSD}, \mathbf{MRSU}\}$ be the type of combined network. There are a total of $3 * 3 * 7 - 3 = 60$ parameter combinations (minus three is for an meaningless $t = \{\phi, \phi\}$.) We use $C_m^t(n)$ to denote the degree centrality of node n with respect to type of centrality measure m and the type of network t. According to this notation, the relation

DOM(p_1,p_2) holds between people p_1 and p_2 if $C_m^t(p_1) > C_m^t(p_2)$. We report percentage accuracy on our gold standard G. The random baseline is 50%.

Table 8.2 presents our naming convention for experiments. For example, **Deg-MRU** refers to the experiment where we use degree centrality as the measure of dominance in an undirected network that is created by connecting the recipients with the people mentioned in an email. We discuss the effect of parameters on performance in turn; while the reader may get the impression that we performed a greedy search through the search space of parameters, we in fact performed all experiments and only the presentation is greedy.

8.6.2 Weighted versus Unweighted Networks

The best unweighted network (**Deg-MRU**) performs at 87.3% accuracy and the best weighted (also **Deg-MRU**) at 86.7%. (In fact, 87.3% accuracy for the unweighted **Deg-MRU** system is the best result we report.) This difference is statistically significant with $p < 0.0001$ (using McNemar's test). If we turn our attention to the best email-only network (weighted and unweighted), we see a similar pattern: the best unweighted email-only network is **Deg-EU** with an accuracy of 85.2%, while the best weighted email-only network is **In-ED** with an accuracy of 83.9% (weighted **Deg-EU** also achieves an accuracy of 83.9%). Again, we see that the unweighted network outperforms the weighted network by a statistically significant margin ($p < 0.0001$).

We interpret these results as follows for the email network: what matters for dominance prediction is not the volume of emails from one person to the other, but the number of other people a person corresponds with. A similar interpretation applies to the mention network: what matters is the number of different people mentioned in emails and not the number of times one person is mentioned.

Table 8.3 presents two examples in which the weighted email network **Deg-EU** makes the wrong prediction but the same unweighted email network makes the right prediction. The table shows that Kenneth Lay (CEO) is predicted less dominant than Alan Comnes (a Public Relations (PR) Specialist) according to the degree centrality measure in the weighted email network; the degree centrality of Kenneth Lay is 92,079, which is lower than the degree centrality of Alan Comnes (146,085). However, according the unweighted email network,

Employee	Designation	Weighted	Unweighted
Kenneth Lay*	CEO	92,079	2,938
Alan Comnes	PR Specialist	146,085	841
Jeff Skilling*	COO	29,787	1,123
Sunil Abraham	Staff	107,298	693

Table 8.3: Two examples in which the degree centrality measure in an unweighted network makes the correct prediction compared with its weighted counter-part (Section 8.6.2). Asterisk (*) denotes higher up in the hierarchy.

Kenneth Lay is correctly predicted to be more dominant; the degree centrality of Kenneth Lay is 2,938, which is higher than the degree centrality of Alan Comnes (841). This example shows that a Public Relations Specialist sends and receives more emails but from fewer people (at least in the sample dataset we have access to).

8.6.3 Type of Network

Hereon, we present results only for the unweighted networks. In this sub-section, we compare three types of networks: email-only (the traditionally used network), mention-only, and a combination of the two.

Table 8.4 presents the performance of the best performing systems for these types of networks. We highlight three scenarios: public, for which we report the results only on the public-tuples (both people are in the public group), private, for which we report the results only on the private-tuples, and All, for which we report results on the whole gold standard G. The first row of the table presents results for the email-only network. The results show that the best performing parameter configuration for the email-only network is **Deg-EU** (email-only, undirected, unweighted network with degree centrality as the measure of dominance). As expected, the performance of the email-only network for the public group is significantly better than its performance for the private group.

We summarize the results from Table 8.4: (1) the email-only networks are never the best performers, (2) for the public group, a combined network of email and mentions outperforms

Network type	public (%Acc)	private (%Acc)	All (%Acc)
Email only	87.9 (**Deg-EU**)	76.3 (**Deg-EU**)	85.2 (**Deg-EU**)
Mention only	68.9 (**Deg-MRD-weighted**)	**81.6** (**Deg-MRU**)	**87.3** (**Deg-MRU**)
Combined	**88.1** (**Deg-EUMRD**)	79.3 (**Out-EDMSRU**)	86.8 (**Out-EDMSRU**)

Table 8.4: Results for the best performing systems based on three different network types and evaluation groups.

the email-only network: even though we have the complete set of communications among the people in this group, the mention network still adds value, and (3) for the private group and overall, the mention-only network performs significantly better than the email network. This result is surprising because the mention network is much sparser than the email network (density of 0.008% for the email network vs. 0.001% for the mention network). We conclude that the mention network provides useful information for predicting dominance relations.

Table 8.5 presents two examples in which the email-only network **Deg-EU** makes the wrong prediction but the mention-only network **Deg-MRU** predicts correctly. For example, John Lavorato (COO) is ranked below Phillip K Allen (a trader) using the email-only network (992 versus 1,771). However, John Lavorato is ranked above Phillip K Allen using the mention-only network (452 versus 248), which is the correct prediction. So while Phillip K. Allen sends/receives emails to/from more people as compared to John Lavorato, more people get mentioned to or mention John Lavorato in their emails.

In addition to the results presented here, we experimented with tuples in which one person belonged to the public group while the other belonged to the private group. All combination of parameters resulted in an extremely high performance at above 90%. The dominance prediction task is relatively easy for these pairs of people. This is explained by

Employee	Designation	**Deg-EU**	**Deg-MRU**
John Lavorato*	COO	992	452
Phillip K Allen	Trader	1771	248
David W Delainey*	COO	1093	298
Phillip K Allen	Trader	1771	248

Table 8.5: Two examples in which the degree centrality measure in an mention-only network makes the correct prediction compared with the email-only network (Section 8.6.3). Asterisk (*) denotes higher up in the hierarchy.

the fact that the public group was chosen by law enforcement because they were most likely to contain information relevant to the legal proceedings against Enron; i.e., the owners of the mailboxes were more likely more highly placed in the hierarchy.

8.6.4 Linking to the Mentioned Person

So far we have established that the best performing networks are undirected and include links resulting from mentions of other people in email. In this subsection, we investigate the mention links in more detail. Specifically, when an email from a sender to one or more recipients contains a mention of another person, we can add a link between the sender and the mentioned (***-MS***), we can add a link between each recipient and the mentioned (***-MR***), or we can add all of these links (***-MSR***). We notice a clear pattern: networks in which links between recipient and mentioned were missing perform much worse than networks where we add links between the recipient and the mentioned person. In fact, the worst performing network where we add links between the mentioned and recipient (***-MR***) outperforms the best performing network where we only add links between the mentioned and sender (***-MS***) by a statistically significant margin. The performance of the first system is 73.6% (**In-MRD**) as compared to the latter, which is 73.4% (**Deg-MSD**). This difference is statistically significant with $p < 0.0001$ (using McNemar's test). Clearly, it is crucial to add links between mentioned and the recipient(s) while establishing dominance relations between Enron employees. We interpret this result further in light of other results

in Section 8.6.6.

8.6.5 Type of Degree Centrality

We have established that the best performing systems use an unweighted mention network where the receiver is definitely linked to the mentioned person. Finally we show that out of the three types of centralities, In-degree centrality is a bad predictor of dominance relations. To make this point, we compare the worst mention network that uses Out-degree centrality (**Out-MR***) with the best mention network that uses In-degree centrality (**In-MR***). The performance of the former is 85.1% (**Out-MRD**) compared to 73.6% (**In-MRD**). This difference is statistically significant with $p < 0.0001$ (using McNemar's test). We note that Degree centrality subsumes Out-degree centrality, thus the fact that our best overall result (87.3%) uses **Deg-MRU**, i.e., Degree centrality, is compatible with this finding. We interpret this result further in the next section.

8.6.6 Summary: What Matters in Mentions

When we use the mention network for dominance prediction, we have seen that we need to include a link from the recipient to the person mentioned (Section 8.6.4), and that we need to include the Out-degree (Section 8.6.5). If links between the person mentioned in the email with the recipient of that email are absent from the mention network, then this underlying network will not be a good predictor of dominance relations. Similarly, if the centrality measure does not include the outgoing edges from nodes, then the mention network will not be a good predictor of dominance relations. Put succinctly, what is significant for dominance prediction in the mention network is the number of people mentioned to a person. Note that we have already determined that the unweighted graph is a better predictor, so it is the number of people mentioned, not the number of mention instances, that is relevant. The number of people who mention the person, and the number of people the person mentions, are less useful. These results lend support to our finding: *you're the boss if people get mentioned to you.*

Table 8.6 presents three examples for showing the importance of linking the recipient and the mentioned person. Network **Out-MSD** measures the number of people a person

Employee	Designation	**Out-MSD**	**In-MSD**	**Out-MRD**
Ken Lay*	CEO	1	1	237
Sara Shackleton	Senior Counsel	208	37	230
Michael Terraso*	VP	22	0	10
Lisa Jacobson	Manager	33	11	3
Greg Whalley*	President	11	11	99
Ed McMichael, Jr	Lead	16	104	28

Table 8.6: Three examples . showing the importance of linking recipient and the mentioned (Section 8.6.6) Asterisk (*) denotes higher up in the hierarchy.

mentions, system **In-MSD** measures the number of people that mention a person, and system **Out-MRD** measures the number of people that are mentioned to a person. Note that for these three sample pairs, the only correct predictor is the **Out-MRD** system. For example, Ed McMichael mentions more people than Greg Whalley, and is mentioned many more times than Greg Whalley, but Greg Whalley has many more people mentioned to him than Ed McMichael. And indeed, Greg Whalley is higher up in the hierarchy than Ed McMichael.

8.7 Conclusion and Future Work

In this chapter we showed the utility of a mention network by demonstrating the predictive power of the mention network for the task of organizational dominance prediction of employees in the Enron email data-set. We acknowledged the peculiarity of the Enron email data-set in that it has two types of people: one for whom we have all their email communications and the other who are simply either sender or recipients to the first set of people. We showed that adding comparatively few mention links to a much denser email network (between the 158 people whose inboxes were used to create the network) improves the performance of our system on the task. But for the private group of people (everyone other than the chosen 158), we showed that the mention network alone is the best predictor of

organizational dominance. By performing a comprehensive set of experiments we were able to conclude the key insight we get about the Enron email corpus: *you are the boss if people get mentioned to you*. We believe this may be attributed to the corporate reporting culture where the managers report to their senior about the performance of their team.

Recall, the mention network is a network in which nodes are entities and links are social events of type OBS. To extract these OBS social events, we utilized a technique quite different from the technique used to build SINNET. We were required to resolve named mentions in the content of emails to actual entities in the corpus. To this end, we developed a name disambiguation technique that was presented in the previous chapter. We also experimented with running SINNET on the content of emails to mine more INR and OBS links. We did this after auto-resolving the mentions of "I" to the sender and the mentions such as "you" and "your" to the recipient (if there was only one person in the To field). However, extracting these links lead to no significant difference in the results of predicting organizational dominance relations.

In the future, we would like to predict organizational hierarchies, not only dominance relations.

Part IV

Extracting Networks from Screenplays

The first part of this thesis introduced a novel type of social network – a network in which nodes are entities and links are social events. Recall, we were only concerned with entities of type Person. We defined two broad categories of social events: observation (OBS) and interaction (INR). Observation social events are events in which only one entity is cognitively aware of the other. Interaction social events are events in which both entities are mutually aware of one another and of their mutual awarenesses. The first part of the thesis also introduced a novel machine learning approach for automatically extracting social networks from unstructured text such as novels.

The second part of this thesis proposed a novel technique for extracting social networks from emails. Emails, unlike novels, have a structure – a network structure that specifies the sender and the recipients of messages. For example, if **John Powell** sends an email to **Mary Heard**, there is a directed link from **John Powell** to **Mary Heard** in the network structure. Such directed links are social events of type OBS. Extracting these directed links is trivial because the information that **John Powell** sends an email to **Mary Heard** is recorded in a structured format. However if, in the content of the email, **John Powell** mentions a person with their first name, say **Sara**, we want to add an OBS directed link from **John Powell** to **Sara**. The problem is that we do not know which **Sara**, out of hundreds of **Sara's** in the corpus, **John Powell** is referring to. For extracting these mention links, the second part of the thesis introduced a novel technique for resolving named mentions in the content of emails to entities in the network.

This third and final part of the thesis introduces a novel technique for extracting social networks from movie screenplays. Screenplays are text documents written by screenwriters for the purposes of storytelling. Unlike novels, which tell a story using free flow text, screenplays tell a story in a text format that is highly structured. For example, screenplays are segmented into scenes and each scene starts with an indicator INT. or EXT. Scenes contain dialogues between characters that are clearly marked using other textual and formatting indicators. Furthermore, unlike novels, the primary mode of storytelling in screenplays is through the interaction of characters. Characters interact with one another using dialogue. The characters are therefore mutually aware of one another and of their mutual awarenesses. Given a well-structured screenplay, creating a network of interactions of characters is trivial

– we know the position of scene boundaries, characters, and their dialogues – connect all conversing characters in a scene with interaction links. However, the screenplays found on the web are ill-structured. We show that identifying scene boundaries, characters, and their dialogues using regular expressions is not sufficient for creating an interaction network. We propose a novel machine learning approach for automatically recovering the structure of screenplays. This allows us to extract social networks, where nodes are characters and links are INR social events, from hundreds of movie screenplays. We utilize these networks for a novel NLP application: automating the Bechdel Test.

This part is organized as follows: Chapter 9 introduces the terminology regarding screenplays, their structure, and the problem definition, Chapter 10 presents our machine learning approach for recovering the structure of screenplays for extracting interaction networks, Chapter 11 uses these extracted networks for automating the *Bechdel Test*.

Chapter 9

Introduction

Screenplays are text documents written by screenwriters for the purposes of storytelling. Screenplays tell a story in a text format that is highly structured. For example, screenplays are segmented into scenes and each scene starts with an indicator INT. or EXT. Scenes contain dialogues between characters that are clearly marked using other textual and formatting indicators. The goal of this chapter is to introduce the terminology regarding screenplays (Section 9.1), present details about the their structure (Section 9.2), which we use to create a regular expression based baseline in the next chapter, provide a formal task definition for parsing screenplays for the purpose of creating movie interaction networks (Section 9.3), and review past literature related to the task (Section 9.4).

9.1 Terminology

Turetsky and Dimitrova [2004] report:

> A screenplay describes a story, characters, action, setting, and dialogue of a film. The actual content of the screenplay follows a (semi) regular format. The first line of any scene or shooting location is called a *slug line*. The slug line indicates whether the scene is to take place inside or outside (INT or EXT), the name of the location ("TRANSPORT PLANE"), and can potentially specify the time of day (e.g. DAY or NIGHT). Following the slug line is a description of the location. Additionally, the description will introduce any new characters that

> appear and any action that takes place without dialogue. Important people or objects are made easier to spot within a page by capitalizing their names. The bulk of the screenplay is the dialogue description. Dialogue is indented in the page for ease of reading and to give actors and filmmakers a place for notes.
>
> Dialogues begin with a capitalized character name and optionally a (V.O.) or (O.S.) following the name to indicate that the speaker should be off-screen (V.O. stands for "Voice-over"). Finally, the actual text of the dialogue is full-justified to a narrow band in the center of the page.

In summary, a screenplay essentially has five elements: scene boundaries (or slug lines), scene descriptions, character names, dialogues spoken by characters, and other information such as page numbers, directions to the camera, etc. We refer to scene boundaries with tag S, scene descriptions with tag N, character names with tag C, dialogues spoken by characters with tag D, and all the remaining information with tag M.

Figure 9.1 shows a snippet of a screenplay from the film *Hannah and Her Sisters*. The left column shows the tags for each line of the screenplay.[1] This snippet starts with a direction to the camera, *CUT TO:*. Following the direction to the camera is the scene boundary, *INT. MICKEY'S OFFICE – NIGHT*. The scene is being shot at night in an interior (INT.) space, MICKEY'S OFFICE. The scene boundary is followed by a scene description that describes the physical setting of the scene, *Gail, wearing her glasses, stands behind a crowded but well-ordered desk. Two assistants, a man and a woman, stand around her.* Following the scene description is a sequence of character names and dialogues spoken by these characters.

Since **Mickey** and **Gail** are having a conversation, there is an INR social event between the entities. Note that the two entities are talking about **Mickey's** doctor, *[He] didn't say you had a brain tumor.* **Mickey's** doctor, **Dr. Wilkes** is mentioned by name in an earlier scene, about 100 lines before these dialogues. Correctly resolving such pronoun mentions to the correct entity is an interesting problem but out of scope for this thesis. We therefore only extract interaction networks from screenplays.

[1] We define a line as a string of non-space characters that ends in a newline character.

```
M|                                                                CUT TO:
 |
S|               INT. MICKEY'S OFFICE - NIGHT
 |
N|               Gail, wearing her glasses, stands behind a crowded but well-
N|               ordered desk.  Two assistants, a man and a woman, stand around
N|               her.
 |
C|                                     MICKEY
M|                                (turning to Gail,
M|                                gesturing nervously)
D|                          Sssss, if I have a brain tumor, I
D|                          don't know what I'm gonna do.
M|                                (sighing)
 |
C|                                     GAIL
D|                          You don't have a brain tumor.  He
D|                          didn't say you had a brain tumor.
 |
C|                                     MICKEY
M|                                (sighing)
D|                          No, naturally
```

Figure 9.1: A scene from the movie *Hannah and Her Sisters*. The scene shows one conversation between two characters, **Mickey** and **Gail**. The line tagged with the tag S is a scene boundary, lines tagged with the tag N belong to a scene description, lines tagged with the tag C contain the names of speaking characters, lines tagged with the tag D contain the dialogue spoken by these characters, and lines containing all the remaining information are tagged using the tag M.

9.2 The Structure of Screenplays

Movie screenplays have a well defined structure:

- All scene boundaries and scene descriptions are at the lowest and fixed level of indentation; lowest relative to the indentation levels of characters and dialogues.[2]
- All speaking character names are at the highest and fixed level of indentation; highest relative to the indentation levels of scene boundaries and dialogues.
- All dialogues are at the middle and fixed level of indentation; middle relative to the indentation levels of scene boundaries and characters.

[2]By level of indentation we mean the number of spaces from the start of the line to the first non-space character.

For example, in the movie in Figure 9.1, all scene boundaries and scene descriptions are at the same level of indentation, equal to five spaces. All character names are at a different but fixed level of indentation, equal to 20 spaces. Dialogues are at an indentation level of eight spaces. These indentation levels may vary from one screenplay to the other, but are consistent within a well formatted screenplay. Furthermore, the indentation level of character names is strictly greater than the indentation level of dialogues, which is strictly greater than the indentation level of scene boundaries and scene descriptions. Apart from indentation, well structured screenplays have the following additional structural properties:

1. Scene boundaries are capitalized and usually start with one of two markers, INT. (for interior) or EXT. (for exterior). Scenes shot in a closed space are marked with INT. Scenes shot in an open space are marked with EXT.

2. Character names are capitalized with optional tags such as (V.O.) for "Voice Over" or (O.S.) for "Off-screen."

3. Scene descriptions follow scene boundaries, which are followed by character names and dialogues.

9.3 Task Definition

Our goal is to automatically extract interaction networks of characters from movie screenplays. Given a well structured screenplay, where we know exactly which line is a scene boundary and which line is a character name, extracting an interaction network is trivial; Weng *et al.* [2009] suggest connecting all pairs of characters that appear between two consecutive scene boundaries with interaction links. However, screenplays found on the web have anomalies in their structure [Gil *et al.*, 2011]: the level of indentation may be inconsistent and unexpected, character names may not be capitalized, scene boundaries may not start with INT./EXT. tags. Thus, for being able to extract interaction networks from screenplays, it is crucial to fix these anomalies. We develop a methodology for automatically fixing the anomalies in the structure of screenplays. We refer to this task as *parsing* screenplays. By parsing we mean assigning each line of the screenplay one of the following five tags: {S (scene

boundary), N (scene description), C (character), D (dialogue), M (other information)}.[3]

9.4 Literature Survey

One of the earliest works motivating the need for parsing screenplays is that of Turetsky and Dimitrova [2004]. Turetsky and Dimitrova [2004] propose a system for automatically aligning written screenplays with their videos. One of the crucial steps, they note, is to parse a screenplay into its different elements: scene boundaries, scene descriptions, character names, and dialogues. The authors propose a grammar for parsing screenplays and present results for aligning one screenplay with its video. Weng *et al.* [2009] motivate the need for parsing screenplays from a social network analysis perspective. The authors propose a set of operations on social networks extracted from movies and television shows in order to find, what they call, *hidden semantic* information. They propose techniques for identifying lead roles in *bilateral* movies (movies with two main characters) for performing community analysis, and for automating the task of story segmentation. Gil *et al.* [2011] extract character interaction networks from plays and movies. They are interested in automatically classifying plays and movies into different genres (comedy, romance, thriller, etc.) by making use of social network analysis metrics. Gil *et al.* [2011] acknowledge that the screenplays found on the internet are not in consistent formats, and propose a regular expression based system for identifying scene boundaries and character names. Walker *et al.* [2012] introduce a corpus of 862 film scripts from The Internet Movie Script Database (IMSDB).[4] In their previous work [Lin and Walker, 2011; Walker *et al.*, 2011], the authors utilize this corpus "to develop statistical models of character linguistic style and use these models to control the parameters of the PERSONAGE generator [Mairesse and Walker, 2011; 2010]." Gorinski and Lapata [2015a] use 30 movies for training and 65 movies for testing for the task of movie script summarization. Both these works, [Walker *et al.*, 2012; Gorinski and Lapata, 2015a], utilize regular expressions and rule based systems for creating their respective corpora. While there is motivation in the literature to parse screenplays, none of the aforementioned

[3]Turetsky and Dimitrova [2004] refer to this task as a screenplay parsing task.

[4]The corpus is freely available at `https://nlds.soe.ucsc.edu/software`

work addresses the task formally. The works rely on regular expressions and grammar of screenplays and do not present an evaluation of the proposed parsing techniques. As a specific example, we present details and limitations of the corpus introduced by Walker *et al.* [2012] for extracting social networks from screenplays.

The corpus introduced by Walker *et al.* [2012] has the following set of files and folders (taken from the README of the downloaded corpus):

- *all_imsdb_05_19_10/*: list of html files of IMSDB film scripts
- *output_chars/*: sorted character dialogue by number of turns
- *output_dial/*: all dialogue in original order
- *output/*: one file per movie character
- *annotated.csv*: some annotations for each movie character

The folder *all_imsdb_05_19_10/* contains the raw html files for 862 screenplays downloaded from IMSDB. The folder *output_chars/* contains 862 files, one file per screenplay, listing all the characters in the screenplay along with all their dialogues. The folder *output_dial/* contains 862 files, one file per screenplay, listing characters and their dialogues in the order of their appearance. The other files and folders in the corpus, namely *output/* and *annotated.csv*, are irrelevant to the discussion since they do not relate to parsing screenplays. Files in the folder *output_dial/* are the most relevant files to the discussion. These files contain a sequence of characters and their dialogues. Following is an excerpt from one of the files in the folder (*output_dial/Alien-3.dial*):

> RIPLEY: Wait. New...
>
> JOHN: Sits next to her. Quite asleep. Hands swathed in white bandages. Book resting on his lap.
>
> THE ALIEN: Big, black shiny-smooth head moves into the taper light. It moves towards her, cable-like arms held out at its side – moving out of sync with its feet – Ripley tries to move - to cry out – She can't.
>
> RIPLEY: AAAAAAAAAAAARGH!

This is a conversation between three characters, **Ripley**, **John**, and **The Alien**. According to this file, the character **Ripley** says "Wait. New. . . ", the character **John** replies "Sits next to her. Quite asleep. Hands swathed in white bandages. Book resting on his lap.", and so on. Note that these pieces of text that are recorded as dialogues spoken by the characters **John** and **The Alien** are not actually dialogues. These are scene descriptions. It is fair for a rule based system to tag these texts as dialogues because structurally the text "Sits next to her. . . . on his lap" seems to appear as a dialogue (see below). This is an example of the kind of anomalies present in screenplays found on the web. Our machine learning based system correctly identifies these texts as scene descriptions and not dialogues.

JOHN

Sits next to her. Quite asleep. Hands

swathed in white bandages. Book resting

on his lap.

Another limitation of the corpus under discussion is that the corpus does not contain scene boundaries. Scene boundaries are necessary for identifying scenes and scenes are necessary for identifying the set of characters that interact with each other. Without scene boundaries, identifying the set of characters that are talking to each other is hard, and thus it is hard to create social networks for movies present in this dataset.

We formalize the task and propose a machine learning based approach that is significantly more effective than the regular expression based baselines. We evaluate our models on their ability to identify scene boundaries and character names, but also on their ability to identify other important elements of a screenplay, such as scene descriptions and dialogues. We believe, these more accurately parsed screenplays have the potential to serve as a better dataset for various applications addressed in the literature that require well structured screenplays.

9.5 Conclusion

In this chapter, we introduced terminology regarding movie screenplays and their structure. We will use this terminology in the next two chapters. We also provided a formal definition for the task of parsing screenplays. Lastly, we presented a discussion of existing literature on the task. All existing techniques are regular expression and grammar based techniques with no evaluation on how well they identify various elements of a screenplay. In the next chapter, we auto-create a training set and present a supervised learning approach that outperforms rule based approaches by a large and significant margin.

Chapter 10

Machine Learning Approach

One of the main challenges in building a system for automatically parsing movie screenplays is the absence of training data. Screenplays, on average, have about 12,500 lines of text; we find a total of 12,510,372 lines in 1002 screenplays. Furthermore, different screenplays have different kinds of anomalies in their structure. Obtaining a wide variety – variety in terms of the types of anomalies – of annotated screenplays from humans is a tedious and a time consuming task. We propose a novel methodology for automatically obtaining a large and varied sample of annotated screenplays. This methodology is inspired by the distant learning paradigm. For different types of anomalies, we *perturb* the training data and train separate classifiers that are experts in handling certain combinations of possible anomalies. We combine these experts into one classifier using ensemble learning techniques. We propose a wide range of features from different levels of language abstractions (lexical, syntactic, and semantic). We also introduce hand-crafted features that incorporate domain knowledge. We show that our ensemble outperforms a regular expression baseline by a large and statistically significant margin. On an unseen test set, we report the performance of a competitive rule based system to be 0.69 F1-measure. This performance is much lower than the performance of our ensemble model, which achieves an F1-measure of 0.96 on the same test set. Apart from performing an intrinsic evaluation, we also present an extrinsic evaluation. We show that the social network extracted from a screenplay that was tagged using our ensemble method is much closer to the gold social network, as compared to the network extracted using a rule based system. The work presented in this chapter was introduced in Agarwal

et al. [2014b].

The rest of the chapter is organized as follows: Section 10.1 presents details of our data collection effort, Section 10.2 presents our regular expression based baseline, Section 10.3 provides an overview of our machine learning methodology, Section 10.4 gives details of the features we employ for training and testing our machine learning models, Section 10.5 presents the experiments and results for the task of parsing screenplays, and Section 10.6 concludes and summarizes the future direction of research.

10.1 Data

We use the Internet Movie Script Database (IMSDB) website[1] for obtaining movie screenplays in plain text format. We crawl a total of 1051 screenplays. Out of these, 49 are found to be empty. Screenplays on average have 12,500 number of lines. Obtaining manual annotations for a screenplay is a tedious and expensive task. We therefore resort to distant supervision for heuristically creating a training dataset. Before presenting our data preparation scheme, we provide a brief overview of distant supervision.

10.1.1 Distant Supervision

The article titled, Forty Seminal Distant Supervision Articles,[2] notes:

> The first acknowledged use of distant supervision was Craven *et al.* [1999] (though they used the term weak supervision); the first use of the formal term distant supervision was in Mintz *et al.* [2009]. Since then, the field has been a very active area of research.

The general idea behind distant supervision is to use heuristics for automatically creating a training dataset from a large corpus. This training set may be noisy (because no human annotation is involved), but the hope is that this heuristically annotated dataset contains useful patterns for the end classification task. For example, Go *et al.* [2009] use the following heuristic for automatically creating a training dataset for sentiment analysis of tweets:

[1] http://www.imsdb.com

[2] http://www.mkbergman.com/1833/forty-seminal-distant-supervision-articles/

- Annotated a tweet as positive if it ends with a positive emoticon such as :)
- Annotated a tweet as negative if it ends with a negative emoticon such as :(

Of course, not all tweets ending with a positive emoticon are positive. For example, the tweet *boys are dumb, plain & simple :-)* does not express a positive sentiment towards boys even though the tweet ends in a positive emoticon. However, a large proportion of the tweets that end in a positive emoticon are in fact of positive sentiment polarity. This simple heuristic allows Go *et al.* [2009] to create a large dataset with millions of training examples. The authors then use these training examples (after removing the emoticons) to train a classifier that uses a bag-of-words feature set. The classifier learns patterns of words to classify unseen tweets into the two categories (positive and negative). In a similar vein, we use heuristics to automatically create a training dataset for the task of parsing screenplays (defined in Section 9.3). We then train classifiers that learn general patterns regarding the five classes and use these classifiers to parse screenplays that contain structural anomalies. The next section presents these hueristics.

10.1.2 Heuristics for Preparing Training Data

Recall from Section 9.2, movie screenplays have a well defined structure:

- All scene boundaries and scene descriptions are at the lowest and fixed level of indentation; lowest relative to the indentation levels of characters and dialogues.
- All speaking character names are at the highest and fixed level of indentation; highest relative to the indentation levels of scene boundaries and dialogues.
- All dialogues are at the middle and fixed level of indentation; middle relative to the indentation levels of scene boundaries and characters.

Figure 10.1 shows the levels of indentation for a snippet from the screenplay *Sleepy Hollow*. Scene boundaries and scene descriptions are at five levels of indentation. Character names are at 29 levels of indentation and the dialogues they speak are at 15 levels of indentation.

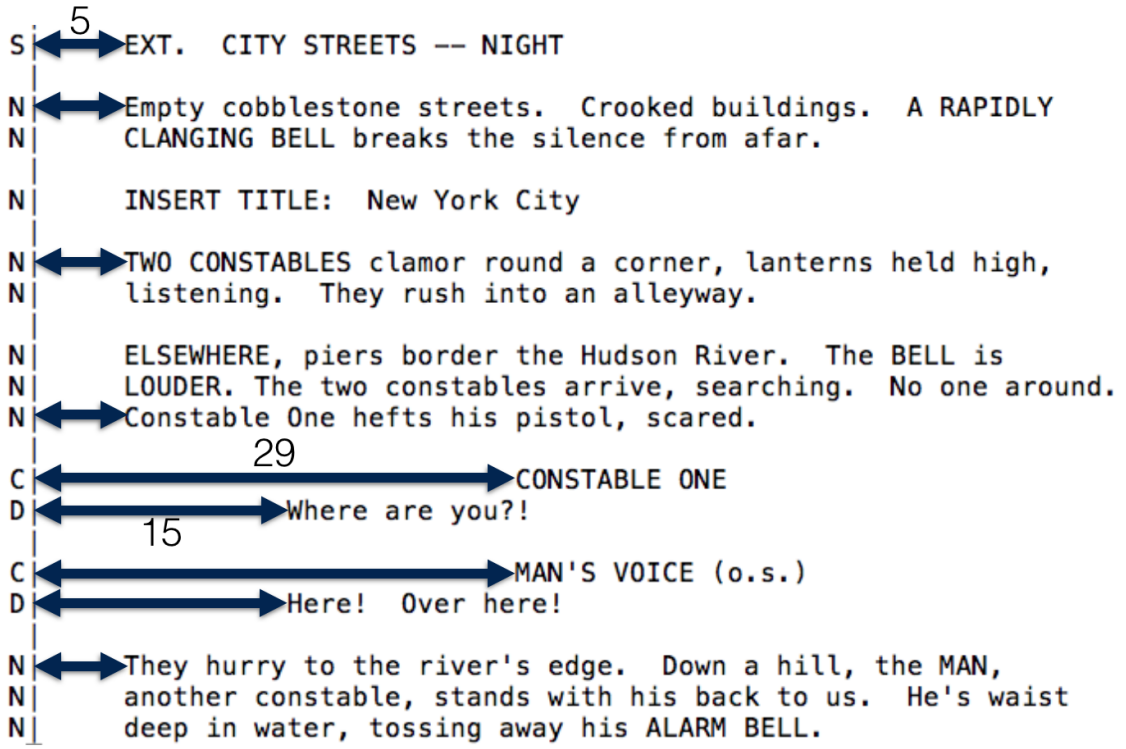


Figure 10.1: Screenplay snippet from the movie *Sleepy Hollow*. Scene boundaries and scene descriptions are at five levels of indentation. Character names are at 29 levels of indentation and the dialogues they speak are at 15 levels of indentation.

For each screenplay, we first find the frequency of all the unique levels of indentation. If the top three unique frequencies constitute 90% of the total lines in the screenplay, we flag that screenplay as well-structured. As an example, for the screenplay *Sleepy Hollow*, we find 2037 lines at five levels of indentation, 1434 lines at 15 levels of indentation, and 753 lines at 29 levels of indentation. The number of lines at these three levels of indentation constitute a majority of the screenplay, specifically 97.1% of the total lines in the screenplay.[3] Since this screenplay satisfies the 90% criteria, we label this screenplay as well-structured.

For each well-structured screenplay, we then use more heuristics to assign each line of the screenplay one out of the following five tags: {S (scene boundary), C (character), D

[3]The exact counts for all levels of indentation are as follows ([indentation_level, number_of_lines>]): {[5, 2037], [15, 1434], [29, 753], [21, 118], [14, 2], [53, 2], [22, 1], [23, 1], [6, 1], [18, 1], [19, 1]}

(dialogue), N (scene description), M (other information)}. Let L_i be the set of lines with level of indentation l_i, where i is an integer that ranges from one to the number of unique levels of indentations. For example, if a screenplay has 11 unique levels of indentations, i ranges from one through 11. Let $f_i = |L_i|$, where $|.|$ denotes the cardinality of a set. Assume that $f_1 \geq f_2 \geq \ldots \geq f_n$, where n is the number of unique levels of indentations. In the running example, $l_1 = 5$, $f_1 = 2037$, and L_1 is the set of lines at level of indentation l_1. Consider the sets L_1, L_2, L_3. For simplicity, assume that $l_1 < l_2 < l_3$.

We know that in a well-structured screenplay, the top three most frequent levels of indentation belong to scene boundaries and scene descriptions, characters, and dialogues. So one of L_1, L_2, L_3 is the set of scene boundaries and scene descriptions, one of L_1, L_2, L_3 is the set of characters, and one of L_1, L_2, L_3 is the set of dialogues. We additionally know that the level of indentation of scene boundaries and scene descriptions is smaller than the levels of indentation of dialogues and characters. Using these two facts, we auto-tag the set of lines in L_1 as scene boundaries and scene descriptions (using the assumption $l_1 < l_2 < l_3$). Using the fact that characters are at a higher level of indentation than dialogues, we auto-tag the set of lines in L_3 as characters (C), and the set of lines in L_2 as dialogues (D). But we still need to divide the set of lines in L_1 into scene boundaries and scene descriptions. For doing so we use the heuristic that unlike scene descriptions, scene boundaries begin with one of two tags (INT. or EXT.) and are capitalized. We tag all the lines in the set L_1 that are capitalized and begin with INT. or EXT. as scene boundaries (S) and all the remaining lines in L_1 as scene descriptions (N). We auto-tag all the remaining lines in the sets $\{L_i | i \in \{4, 5, \ldots, n\}\}$ with tag M.

Before using an auto-tagged screenplay as a training example, we programmatically check the *sanity* of these screenplays. For checking their sanity, we utilize the fact that scene descriptions must appear after scene boundaries, character names must appear after scene descriptions, and dialogues must appear after character names. After applying the sanity checks, we obtain a set of 222 (out of 1002) auto-tagged screenplays that we use for training and developement of our machine learning models.

10.1.3 Data Distribution

Table 10.1 presents the distribution of our training, development, and test sets. We use a random subset of 14 screenplays from the set of 222 movies for training, and another random subset of 8 screenplays for development.[4] For the test set, we ask our annotator to annotate a randomly chosen screenplay (*Silver Linings Playbook*) from scratch. We choose this screenplay from the set of movies that we were unable to tag automatically, i.e. *not* from the set of 222 movies.

Data	Number of screenplays	# S	# N	# C	# D	# M
TRAIN	14	2,445	21,619	11,464	23,814	3,339
DEV1	5	714	7,495	4,431	9,378	467
DEV2	3	413	5,431	2,126	4,755	762
TEST	1	164	845	1,582	3,221	308

Table 10.1: Data distribution

10.2 Baseline Approach

Gil *et al.* [2011] mention the use of regular expressions for parsing screenplays. However, they do not specify the regular expressions or their exact methodology. We use common knowledge about the structure of the screenplays (see Section 9.2) to build a baseline system. This baseline system uses regular expressions and takes into account the grammar of screenplays.

From common knowledge we know that the scene boundaries contain one of many tokens such as DAY, NIGHT, DAWN, SUNSET, SUNRISE, INT., EXT., INTERIOR, EXTERIOR. In the first pass, our baseline system tags all the lines of a screenplay that contain one or many such tokens with the tag S (for scene boundary). In the same pass, the baseline system also tags lines that contain tokens such as V.O. (voice over), O.S. (Off stage) with the tag C

[4]Our experiments show that a set of 14 screenplays is sufficient for learning. We have five classes and each screenplay has hundreds of instances for each class. Even with 14 screenplays, we have a total of 62,681 training instances.

(for character). Lastly, the system tags all the lines that contain tokens such as CUT TO:, DISSOLVE TO, with the tag M. This exhausts the list of regular expression matches that indicate a certain tag.

In the second pass, we incorporate prior knowledge that scene boundaries and character names are capitalized. For this, the system tags all the untagged lines that are capitalized *and* that have greater than three words as scene boundaries (tag S). The system further tags all the untagged lines that are capitalized *and* have less than or equal to three words as characters (tag C). The choice of the number three is not arbitrary; upon examination of the set of 222 screenplays we found that less than two percent of the character names were of length greater than three words.

Finally, we incorporate prior knowledge about the relative positions of dialogues and scene descriptions to tag the remaining untagged lines with one of two tags: D (for dialogue) or N (for scene description). The system tags all the untagged lines between a scene boundary and the first character occurrence within that scene with tag N. Additionally, the system tags all the lines between consecutive character occurrences with tag D. The system also tags the line between the last character occurrence (in a scene) and the next scene boundary with tag D.

This is a strong baseline; it achieves a macro-F1 measure of 0.96 on the development set DEV1 (see Table 10.5 in Section 10.5).

10.3 Machine Learning Approach

Note that our baseline system is not dependent on the level of indentation (it achieves a high macro-F1 measure without using indentation information). Therefore, we have already dealt with one common anomaly found in screenplays – inconsistent and unpredictable indentation. However, there are other common anomalies, such as

1. missing scene boundary specific patterns (INT. and EXT.),
2. uncapitalized scene boundaries, and
3. uncapitalized character names.

Our experiments and results show that a rule based system is not well equipped to handle such anomalies. We propose a machine learning appraoch that is well equipped to handle a random distribution of these three anomalies. The system may easily be extended to handle other kinds of anomalies.

10.3.1 Terminology

For ease of explanation, we create and present a simple encoding scheme shown in Table 10.2. We represent each kind of anomaly with a bit (0 or 1). A 0 denotes *anomaly not present*. A 1 denotes *anomaly present*. We represent a screenplay with a bit string of length three. The least significant bit stands for the third type of anomaly, namely, uncapitalized character names. The second bit represents the second type of anomaly, namely, uncapitalized scene boundaries, and the most significant bit represents the first type of anomaly, namely, missing scene boundary specific patterns INT and EXT. For example, the bit string 000 represents the set of well structured screenplays (no amomalies), the bit string 001 represents the set of screenplays in which we lower case all character names, the bit string 011 represents the set of screenplays in which we lower case both, the scene boundaries and character names, and so on.

10.3.2 Overall Machine Learning Approach

Figure 10.2 illustrates our overall machine learning scheme. In the first step (STEP 1), we use the heuristics presented in Section 10.1 to create a set of well structured screenplays from the crawled data. All screenplays that are not deemed well structured are labeled ill-structured.

In the second step (STEP 2), we randomly sample a set of screenplays for training (call it TRAIN) and two sets for development (call them DEV1 and DEV2). Training a classifier on the set of well strucutred screenplays is bound to fail at test time – because at test time we might be confronted with a screenplay that has a random distribution of these three types of anomalies. We take motivation from ensemble learning community to first train experts for detecting a certain type of anomaly or a certain type of anomaly combination. We then combine these experts to make predictions on unseen screenplays.

Description of anomaly added to well structured screenplays	Missing INT/EXT tags in scene boundaries	Uncapitalized scene boundaries (S)	Uncapitalized character names (C)
Well structured screenplay	0	0	0
Only character names uncapitalized	0	0	1
Both character names and scene boundaries uncapitalized	0	1	1
INT/EXT tags removed from scene boundaries	1	0	0
INT/EXT tags removed from scene boundaries and character names uncapitalized	1	0	1
INT/EXT tags removed from scene boundaries and scene boundaries uncapitalized	1	1	0
INT/EXT tags removed from scene boundaries and both scene boudnaries and character names uncapitalized	1	1	1

Table 10.2: Common types of anomalies found in screenplays and our encoding scheme.

For training the experts (STEP 3), we create eight copies of the training set (TRAIN) and eight corresponding copies of the first development set (DEV1). Each copy corresponds to a unique combination of anomalies. For example, the sets TRAIN_000/DEV1_000 corresponds to the set of well structured screenplays. The sets TRAIN_001/DEV1_001 corresponds to the set of screenplays in which we lower case all character names. The sets TRAIN_011/DEV1_011 corresponds to the set of screenplays in which we lower case both,

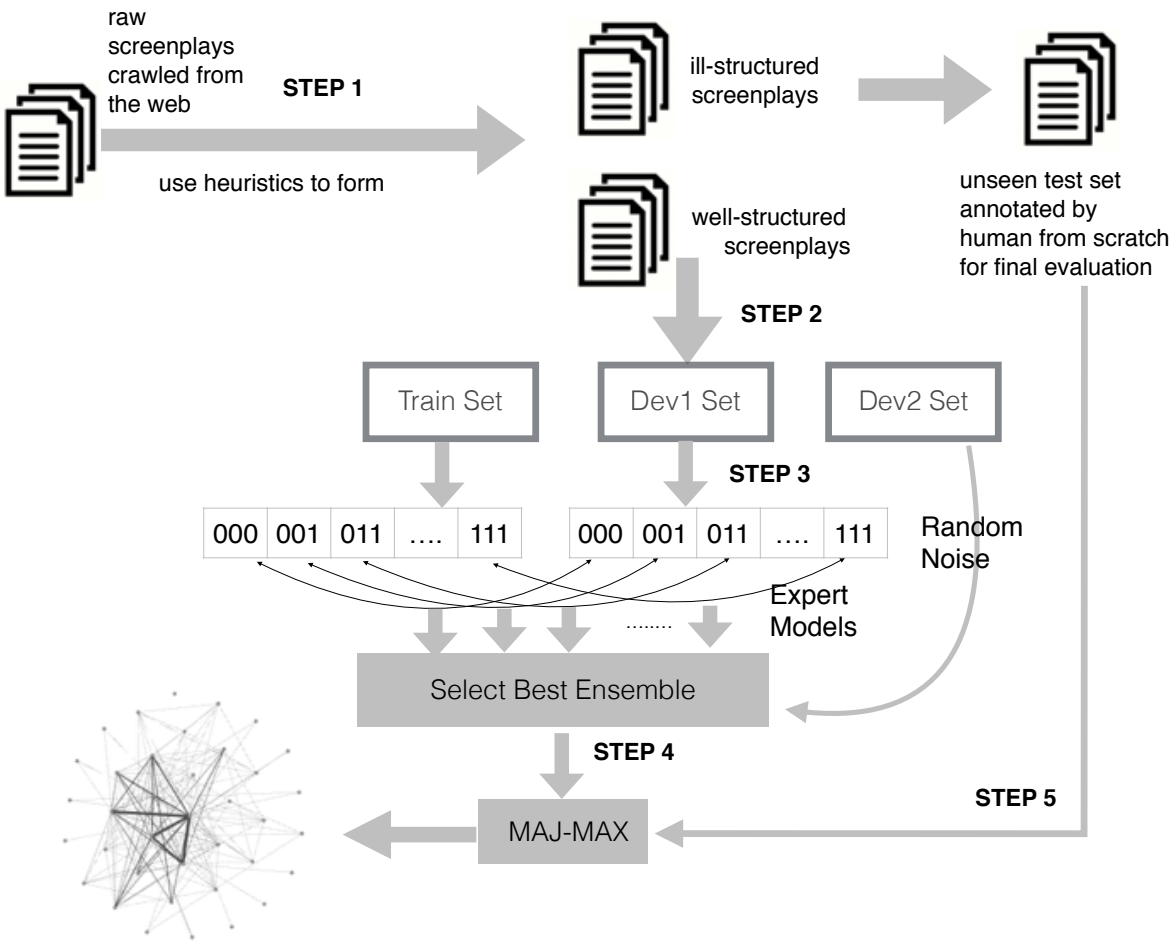


Figure 10.2: Overall machine learning approach for parsing screenplays.

the scene boundaries and the character names, and so on.

Now we have eight training and eight development sets. We train eight models, and choose the parameters for each model by tuning on the respective development set. For example, we train on TRAIN_000 and tune the parameters on DEV1_000. Each of these models acts as an expert in dealing with particular types of anomalies. However, there are two remaining issues: (1) we need one model at test time and (2) the anomalies may be distributed randomly (each expert expects a uniform distribution of anomalies). To tackle the first issue (STEP 4), we experiment with three ensemble methods. We select the ensemble that performs the best on the DEV2 set. We add all three types of anomalies randomly to the DEV2 set. Thus, the best ensemble is able to handle a random distribution of anomalies at test time, which addresses the second issue.

Finally, in STEP 5, we test the performance of our ensemble on an unseen test set,

randomly sampled from the set of ill formed screenplays.

For training the individual models, we use Support Vector Machines (SVMs), and represent data as feature vectors, discussed in the next section.

10.4 Features

We utilize six broad categories of features: bag-of-words features (BOW), bag-of-punctuation-marks features (BOP), bag-of-terminology features (BOT), bag-of-frames features (BOF), bag-of-parts-of-speech features (POS), and hand-crafted features (HAND). Table 10.3 provides a succinct description of these feature sets. We use Semafor for obtaining FrameNet frames in sentences. Note that we utilized frame semantic features for extracting social networks from unstructured text (Chapter 4 of this thesis). We convert each *line* of a screenplay into a feature vector of length 5,497: 3,946 for BOW, 22 for BOP, 2*58 for BOT, 2*45 for POS, 2*651 for BOF, and 21 for HAND. We define a line as a string of non-space characters that ends in a newline character. We also refer to a line as an **input example** that needs to be classified into one of five categories.

BOW, BOP, and BOT are binary features; we record the presence or absence of elements of each bag in the input example. The number of terminology features is multiplied by two (2*58 for BOT) because we have one binary vector for "line *contains* term", and another binary vector for "line *is* term." For instance, if the input example is "CUT TO", the binary feature "input_line_is_CUT_TO" will be set to 1. Furthermore, the binary feature "input_line_contains_CUT" will also be set to 1.

We have two sets of features for POS and two sets of features for BOF. One set is binary and similar to other binary features that record the presence or absence of parts-of-speech and frames in the input example. The other set is numeric. We record the normalized counts of each part-of-speech and frame respectively. For instance, if an example has three nouns and one verb, the normalized count for nouns is 0.75 and the normalized count for verb is 0.25. Similarly, if an example has two distinct frames, each will have a normalized count of 0.5. The impetus to design this second set of features for parts-of-speech and frames is the following: we expect some classes to have a characteristic distribution of parts-of-speech and

Feature Name	Feature Set
Bag-of-words (BOW)	All words except stop words and frequency less than 50. We use word stems obtained from Porter stemmer.
Bag-of-punctuation-marks (BOP)	' " [ ] ( ) { } < > : , - ... ! ≪ ≫ . ? ; / |
Bag-of-terminology (BOT)	AERIAL SHOT, ANGLE ON, ANGLE, END, b.g., CLOSE ON, ... A full list is presented in Appendix C.2
Bag-of-frames (BOF): displaying frame name and its count in our corpus. The frames are sorted in descending order of counts.	Locative_relation 4934, Observable_body_parts 3520, Intentionally_act 3103, Calendric_unit 2914, Arriving 2737, Quantity 2274, Cardinal_numbers 2271, Building_subparts 2061, Temporal_collocation 1808, People 1759, Buildings 1747, ... A full list is presented in Appendix C.1
Bag-of-parts-of-speech (POS)	The list of 45 parts-of-speech tag from the following website: `http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html` [Santorini, 1990].
Hand-crafted-features (HAND)	has-non-alphabetical-chars, has-digits-majority, has-alpha-majority, is-quoted, capitalization (has-all-caps, is-all-caps), scene boundary (has-INT, has-EXT), date (has-date, is-date), number (has-number, is-number), and parentheses (is-parenthesized, starts-with-parenthesis, ends-with-parenthesis, contains-parenthesis), and others presented in Section 10.4.

Table 10.3: The complete set of features used for parsing screenplays.

frames. For example, scene boundaries contain the location and time of scene. Therefore, we expect them to have a majority of nouns, and frames that are related to location and time. For example, for the scene boundary in Figure 10.3 (*EXT. FBI ACADEMY GROUNDS, QUANTICO, VIRGINIA - DAY*), we find the following distribution of parts-of-speech and frames: 100% nouns (we remove EXT/INT before running the POS tagger and semantic parser), 50% frame LOCALE (with frame evoking element *grounds*), and 50% frame CALENDRIC_UNIT (with frame evoking element *DAY*). Similarly, we expect the character names to have 100% nouns, and no frames. We use Stanford part-of-speech tagger [Toutanova *et al.*, 2003] for obtaining the part-of-speech tags and Semafor [Chen *et al.*, 2010] for obtaining the FrameNet [Baker *et al.*, 1998] frames.

We devise 21 hand-crafted features. Sixteen of these features are binary (0/1). We list these features here (the feature names are self-explanatory): has-non-alphabetical-chars, has-digits-majority, has-alpha-majority, is-quoted, capitalization (has-all-caps, is-all-caps), scene boundary (has-INT, has-EXT), date (has-date, is-date), number (has-number, is-number), and parentheses (is-parenthesized, starts-with-parenthesis, ends-with-parenthesis, contains-parenthesis). We bin the preceding number of blank lines into four bins: 0 for no preceding blank lines, 1 for one preceding blank line, 2 for two preceding blank lines, and 3 for three or more preceding blanks. We also bin the percentage of capitalized words into four bins: 0 for the percentage of capitalized words lying between [0-25%), 1 for [25-50%), 2 for [50-75%), and 3 for [75-100%]. We use three numeric features: number of non-space characters (normalized by the maximum number of non-space characters in any line in a screenplay), number of words (normalized by the maximum number of words in any line in a screenplay), and number of ASCII characters (normalized by the maximum number of ASCII characters in any line in a screenplay).

For each line, say $line_i$, we incorporate context up to x lines. Figure 10.3 shows the lines at context -2 and +3 for the line containing the text *CRAWFORD*. To do so, we extend the feature vector for $line_i$ with the feature vectors of $line_{i-1}, line_{i-2}, \ldots line_{i-x}$ and $line_{i+1}, line_{i+2}, \ldots line_{i+x}$. x is one of the parameters we tune at the time of training. We refer to this parameter as CONTEXT.

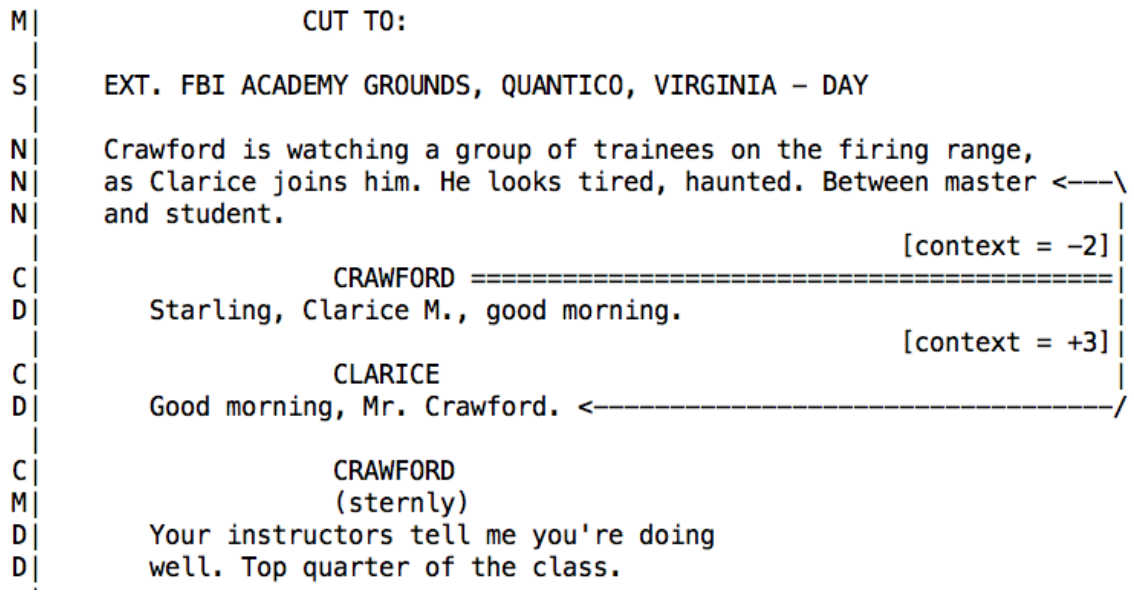


Figure 10.3: Example screenplay: first column shows the tags we assign to each line in the screenplay. M stands for "Meta-data", S stands for "Scene boundary", N stands for "Scene description", C stands for "Character name", and D stands for "Dialogue." We also show the lines that are at context -2 and +3 for the line "CRAWFORD."

10.5 Experiments and Results

In this section, we present experiments and results for the task of parsing screenplays i.e. classifying the lines of a screenplay into one of five categories: {scene boundary (S), scene description (N), character name (C), dialogue (D), other information (M)}. Table 10.4 presents the data distribution. To verify that the amount training data (14 screenplays) is sufficient for training and to verify that our feature set is rich enough to handle all possible combinations of anomalies, we experiment with eight different sets: {Train_000/Dev1_000, Train_001/Dev1_001, Train_011/Dev1_011, ..., Train_111/Dev1_111}. We present the results for training these eight experts in Section 10.5.1. In Section 10.5.2, we present strategies for combining these eight models into one model that is able to handle a random distribution of anomalies at test time. We select the best ensemble of these eight models and the best set of features for the end task by tuning on the second development set, Dev2. Section 10.5.3 presents an analysis of our features, highlighting the features that are

most important for classification. Finally, in Section 10.5.4, we present results on an unseen hand-annotated test set (TEST). For all our experiments, we use the default parameters of SVM as implemented by the SMO algorithm of Weka [Hall *et al.*, 2009]. We use a linear kernel.

Data	Number of screenplays	# S	# N	# C	# D	# M
TRAIN	14	2,445	21,619	11,464	23,814	3,339
DEV1	5	714	7,495	4,431	9,378	467
DEV2	3	413	5,431	2,126	4,755	762
TEST	1	164	845	1,582	3,221	308

Table 10.4: Data distribution

10.5.1 Training Experts

We merge training data from all 14 movies into one (TRAIN). We then randomize the data and split it into 10 pieces (maintaining the relative proportions of the five classes). We plot a learning curve by adding 10% of training data at each step.

Figure 10.4 presents the learning curves for training a model on TRAIN_000 and testing on DEV1_000.[5] There are six learning curves for six different values of the CONTEXT feature (0 through 5).

The learning curves show that the performance of our classifier without any context (CONTEXT = 0) is significantly worse than the classifiers trained on a non-zero context. In fact, training with CONTEXT size of one is at least as good as other context sizes. We therefore choose CONTEXT equal to one for our remaining experiments. Furthermore, the learning saturates early, and stabilizes at about 50% of the training data. We thus use only 50% of the entire training dataset for training. This observation also confirms the fact that 14 screenplays are sufficient for training.

Table 10.5 shows a comparison of our rule based baseline with the models trained using machine learning. Each column corresponds to a certain set of anomalies (terminology pre-

[5]Learning curves for all our other models were similar.

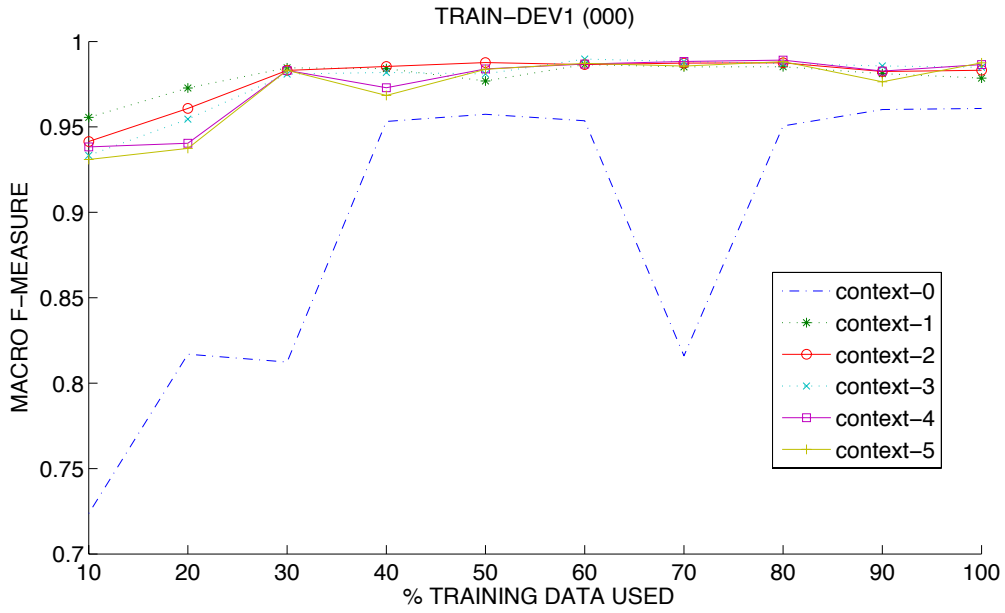


Figure 10.4: Learning curves for training on TRAIN_000 and testing on DEV1_000. X-axis is the % of training data, in steps of 10%. Y-axis is the macro-F1 measure for the five classes. Each learning curve belongs to a particular value of CONTEXT.

sented in Table 10.2). For the setting 000, when there is no anomaly in the screenplays, our rule based baseline performs well, achieving a macro-F1 measure of 0.96. However, our machine learning model outperforms the baseline by a statistically significant margin, achieving a macro-F1 measure of 0.99.[6] Results in Table 10.5 also show that while a deterministic regular expression based system is not well equipped to handle anomalies (the performance drops to as low as 0.23 F1-measure), our feature set is sufficiently rich for our machine learning models to learn any combination of the anomalies, achieving an F1-measure of 0.98 on average.

10.5.2 Finding the Right Ensemble

We have trained eight separate models; each model is an *expert* in handling a particular combination of anomalies. However, there are two remaining issues: (1) we need one model

[6]We calculate statistical significance using McNemar's significance test, with significance defined as $p < 0.05$.

	000	001	010	011	100	101	110	111
Rule based	0.96	0.49	0.70	0.23	0.93	0.46	0.70	0.24
ML model	**0.99**	**0.99**	**0.98**	**0.99**	**0.97**	**0.98**	**0.98**	**0.98**

Table 10.5: Comparison of performance (macro-F1 measure) of our rule based baseline with our machine learning based models on development sets DEV1_000, DEV1_001, ..., DEV1_111. All models are trained on 50% of the training set, with the feature space including CONTEXT equal to 1.

at test time and (2) the anomalies may be distributed randomly (each expert expects a uniform distribution of anomalies). To overcome these two issues, we explore the following three ways of combining these eight models:

1. MAJ: Given a test example, we obtain a vote from each of our eight models, and take a majority vote. At times of a clash, we pick one randomly.

2. MAX: We pick the class predicted by the model that has the highest confidence in its prediction. Since the confidence values are real numbers, we do not see any clashes.

3. MAJ-MAX: We use MAJ but at times of a clash, pick the class predicted by the classifier that has the highest confidence from among the classifiers that clash.

Movie	000	001	010	011	100	101	110	111	MAJ	MAX	MAJ-MAX
LTC	0.87	0.83	0.79	0.94	0.91	0.86	0.79	0.96	0.97	0.95	**0.98**
X-files	0.87	0.84	0.79	0.93	0.86	0.84	0.79	0.92	0.94	0.94	**0.96**
Titanic	0.87	0.87	0.81	0.94	0.86	0.83	0.82	0.93	0.94	0.95	**0.97**
Average	0.87	0.85	0.80	0.94	0.88	0.84	0.80	0.94	0.95	0.95	**0.97**

Table 10.6: Macro-F1 measure for the five classes for testing on DEV2 set. 000 refers to the model trained on data TRAIN_000, 001 refers to the model trained on data TRAIN_001, and so on. MAJ, MAX, and MAJ-MAX are the three ensembles. The first column is the movie name. LTC refers to the movie "The Last Temptation of Christ."

Row #	Feature set	LTC	X-files	Titanic
1	All	0.98	0.96	0.97
2	All - BOW	**0.94**	**0.92**	**0.94**
3	All - BOP	0.98	**0.97**	0.97
4	All - BOT	**0.97**	**0.95**	**0.96**
5	All - BOF	**0.96**	**0.93**	**0.96**
6	All - POS	0.98	0.96	**0.95**
7	All - HAND	**0.94**	**0.93**	**0.93**

Table 10.7: Performance of MAJ-MAX classifier with feature removal. Statistically significant differences are in bold.

Table 10.6 shows macro-F1 measures for the three movies in our DEV2 set. The three movies are LTC (The Last Temptation of Christ), X-files, and Titanic. Note that we add the three types of anomalies randomly to the DEV2 set. The table presents the performance of our three ensembles (last three columns) along with the performance of our eight experts that are trained to handle a uniform distribution of eight different types of anomalies.

The results show that all our ensembles (except MAX for the movie *The Last Temptation of Christ*) perform better than the individual models. Furthermore, the MAJ-MAX ensemble outperforms the other two ensembles by a statistically significant margin. We choose MAJ-MAX as our final classifier.

10.5.3 Feature Analysis

Table 10.7 shows the results for our feature ablation study. These results are for our final model, MAJ-MAX. The row "All" presents the results when we use all our features for training. The consecutive rows show the result when we remove the mentioned feature set. For example, the row "All - BOW" shows the result for our classifier trained on all but the bag-of-words feature set.

The results in rows 2 and 7 of Table 10.7 show that the performance drop is maximum due to the removal of bag-of-words (BOW) and our hand-crafted features (HAND). Clearly,

these two sets of features are the most important.

The next highest drop is due to the removal of the bag-of-frames (BOF) feature set (see row 5). Through error analysis we find that the drop was because the recall of dialogues decreases significantly. The BOF features help in disambiguating between the M category, which usually has no frames associated with them, and dialogues.

Removing bag-of-punctuation (BOP) features (row 3) results in a significant increase in the performance for the movie *X-files*, with a small increase for other two movies. We remove this feature from our final classifier.

Removing bag-of-terminology (BOT) features (row 4) results in a significant drop in the overall performance of all movies.

Removing parts-of-speech (POS) features (row 6) results in a significant drop in the overall performance for the movie *Titanic*. Through error analysis we find that the drop in performance is due the drop in the performance of detecting scene boundaries. Scene boundaries almost always consist of 100% nouns and the POS features help in capturing this characteristic distribution indicative of scene boundaries.

Our results also show that though the drop in performance for some feature sets is larger than the others, it is the conjunction of all features that helps us achieve a high F1-measure.

10.5.4 Performance on the Test Set

Table 10.8 shows a comparison of the performance of our rule based baseline with our best machine learning based model on our test set, TEST. The results show that our machine learning based models outperform the baseline by a large and significant margin on all five classes (0.96 versus 0.69 macro-F1 measure respectively). Note that the recall of the baseline is generally high, while the precision is low. Moreover, for this test set, the baseline performs relatively well on tagging character names and dialogues. However, we believe that the performance of the baseline is unpredictable. It may get lucky on screenplays that are well-structured (in one way or the other), but it is hard to comment on the robustness of its performance. In contrast, our ensemble is robust, hedging its bets on eight models, which are trained to handle different types and combinations of anomalies.

In Tables 10.9 and 10.10, we present an extrinsic evaluation on the test set. We extract

	Baseline			Maj-Max		
Tag	P	R	F1	P	R	F1
Scene boundary (S)	0.27	1.00	0.43	0.99	1.00	**0.99**
Scene description (N)	0.21	0.06	0.09	0.88	0.95	**0.91**
Character name (C)	0.89	1.00	0.94	1	0.92	**0.96**
Dialogue (D)	0.99	0.94	0.96	0.98	0.998	**0.99**
Other (M)	0.68	0.94	0.79	0.94	0.997	**0.97**
Avgerage	0.61	0.79	0.69	0.96	0.97	**0.96**

Table 10.8: A comparison of performance of our rule based baseline with our best machine learning model on the five classes.

a network from our test movie screenplay (*Silver Linings Playbook*) by using the tags of the screenplay as follows [Weng *et al.*, 2009]: we connect all characters having a dialogue with each other in a scene with links. Nodes in this network are characters, and links between two characters signal their participation in the same scene. We form three such networks: 1) based on the gold tags ($\mathcal{N}_G$), 2) based on the tags predicted by Maj-Max ($\mathcal{N}_{\text{Maj-Max}}$), and 3) based on the tags predicted by our baseline ($\mathcal{N}_B$). Table 10.9 compares the number of nodes, number of links, and graph density of the three networks. It is clear from the table that the network extracted by using the tags predicted by Maj-Max is *closer* to the gold network. For example, the number of nodes in $\mathcal{N}_{\text{Maj-Max}}$ is 37, which is closer to the number of nodes in the gold network $\mathcal{N}_G$ (41 nodes), compared to the number of nodes in the baseline network $\mathcal{N}_B$ (202 nodes). The same is the case for the number of links and graph density. The baseline incorrectly detects several scene boundaries as character names. This results in an overprediction of characters. Furthermore, since the baseline misses several scene boundaries (by virtue of labeling them as characters), several scenes appear as one big scene, and the number of pairwise links between characters explode.

Centrality measures are one of the most fundamental social network analysis metrics used by social scientists [Wasserman and Faust, 1994]. Table 10.10 presents a comparison of Pearson's correlation coefficient for various centrality measures for $\{\mathcal{N}_B, \mathcal{N}_G\}$, and

$\{\mathcal{N}_{\text{Maj-Max}}, \mathcal{N}_G\}$ for the top ten characters in the movie. The table shows that across all these measures, the statistics obtained using the network $\mathcal{N}_{\text{Maj-Max}}$ are significantly more correlated to the gold network ($\mathcal{N}_G$), as compared the the baseline network ($\mathcal{N}_B$). We conclude that the network extracted using our machine learning models is significantly more accurate than the network extracted using a regular expression based baseline.

	$\mathcal{N}_B$	$\mathcal{N}_{\text{Maj-Max}}$	$\mathcal{N}_G$
# Nodes	202	37	41
# Links	1252	331	377
Density	0.036	0.276	0.255

Table 10.9: A comparison of network statistics for the three networks extracted from the movie *Silver Linings Playbook*.

Model	Degree	Weighted Degree	Closeness	Betweenness	PageRank	Eigen
$\mathcal{N}_B$	0.919	0.986	0.913	0.964	0.953	0.806
$\mathcal{N}_{\text{Maj-Max}}$	**0.997**	**0.997**	**0.997**	**0.997**	**0.998**	**0.992**

Table 10.10: A comparison of Pearson's correlation coefficients of various centrality measures for $\mathcal{N}_B$ and $\mathcal{N}_{\text{Maj-Max}}$ with $\mathcal{N}_G$.

Figure 10.5, 10.6, and 10.7 show a visual comparison of the network plots based on the tagged screenplays, as tagged by our baseline, machine learning model, and hand annotated screenplay. Note that the second and third networks are visually similar.

10.6 Conclusion and Future Work

In this chapter, we presented a NLP and ML based approach for the task of parsing screenplays. We showed that this approach outperforms a regular expression and grammar based approach by a large and significant margin. One of the main challenges we faced early on was the absence of training and test data. We proposed a methodology for learning to handle anomalies in the structure of screenplays without requiring human annotations. We

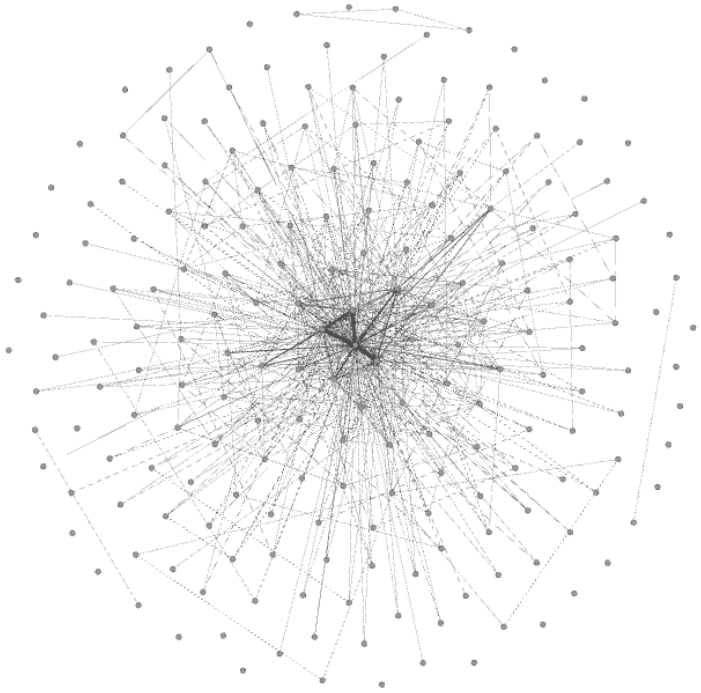


Figure 10.5: Network created from the screenplay parsed using the rule based baseline for the movie *Silver Linings Playbook*.

believe that the machine learning approach proposed in this chapter is general and may be used for parsing semi-structured documents outside of the context of movie screenplays.

In the future, we will apply our approach to parse other semi-structured sources of social networks such as television show series and theatrical plays.

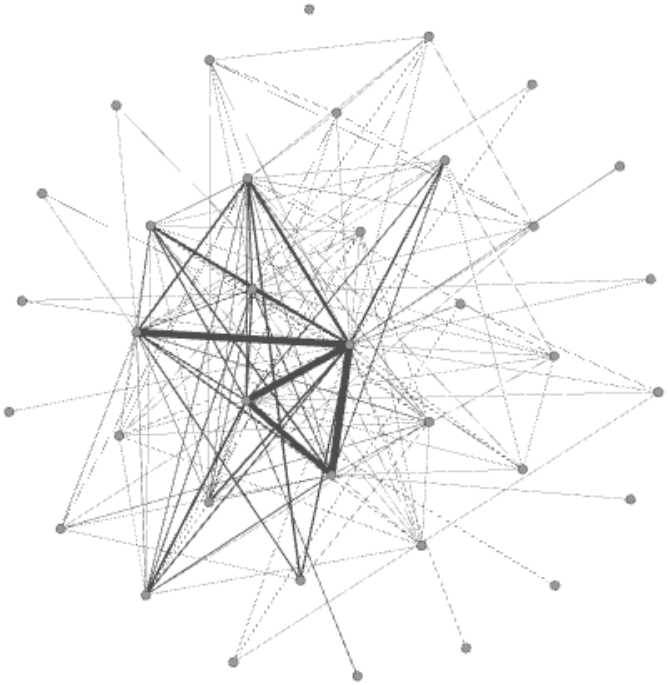


Figure 10.6: Network created from the screenplay parsed using our machine learning model for the movie *Silver Linings Playbook*.

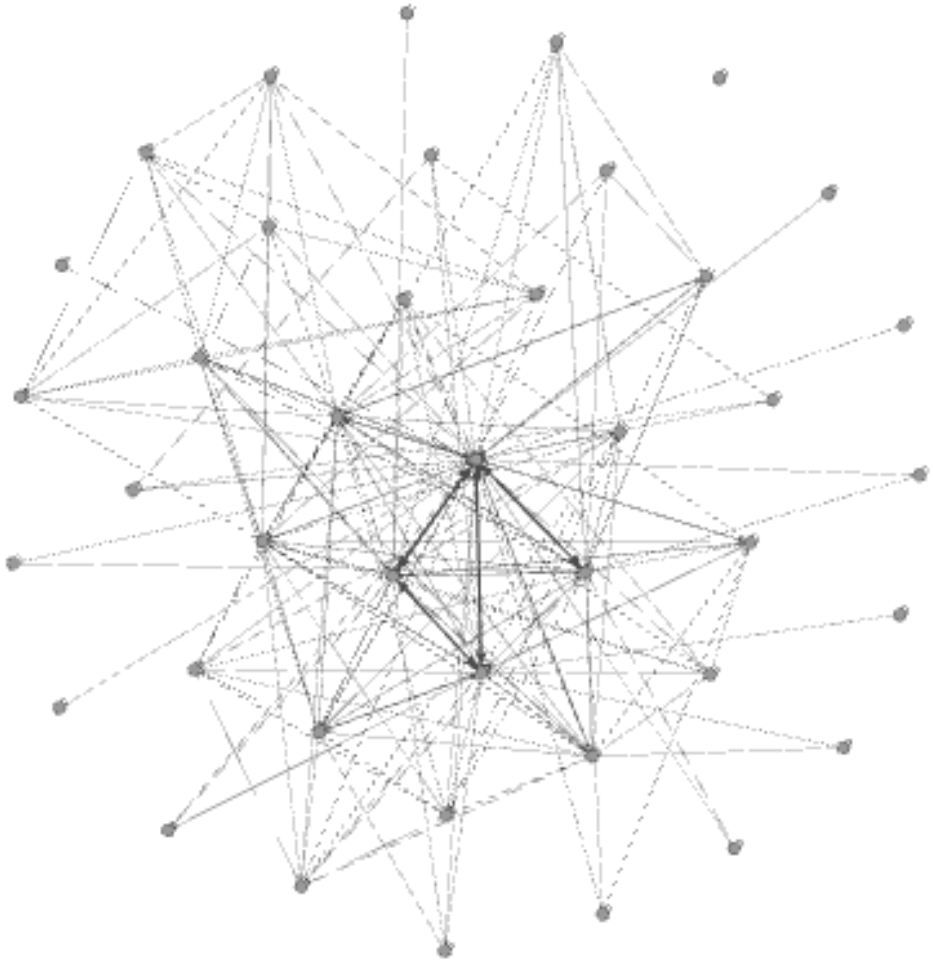


Figure 10.7: Network created from the screenplay that was manually annotated by a human for the movie *Silver Linings Playbook*.

Chapter 11

Application: Automating the Bechdel Test

The *Bechdel test* is a sequence of three questions designed to assess the presence of women in movies. Many believe that because women are seldom represented in film as strong leaders and thinkers, viewers associate weaker stereotypes with women. In this chapter, we present a computational approach to automate the task of finding whether a movie passes or fails the Bechdel test. This allows us to study the key differences in language use and in the importance of roles of women in movies that pass the test versus the movies that fail the test. Our experiments confirm that in movies that fail the test, women are in fact portrayed as less-central or less-important characters.

The work presented in this chapter was introduced in Agarwal *et al.* [2015].

11.1 Introduction

The Bechdel test is a series of three questions, which originated from Alison Bechdel's comic *Dykes to Watch Out For* [Bechdel, 1986]. The three questions (or tests) are as follows:

T1: are there at least two named women in the movie?

T2: do these women talk to each other? and

T3: do these women talk to each other about something besides a man? If after watching a movie, the viewers answer "yes" to all three questions, that movie is said to pass the Bechdel test.

The test was designed to assess the presence of women in movies. Some researchers have embraced the test as an effective primary detector for male bias [Scheiner-Fisher and Russell III, 2012]. Due to its generality, the Bechdel test has also been used to assess the presence of women in dialogues held on social media platforms such as MySpace and Twitter [Garcia *et al.*, 2014]. Several researchers have noted that gender inequality roots itself in both the subconscious of individuals and the culture of society as a whole [Žižek, 1989; Michel *et al.*, 2011; García and Tanase, 2013]. Therefore, combining the Bechdel test with computational analysis can allow for the exposure of gender inequality over a large body of films and literature, thus having the potential to alert society of the necessity to challenge the status quo of male dominance.

While the Bechdel test was originally designed to assess the presence of women in movies, it has subsequently been used to comment on the importance of roles of women in movies. But how does *talking about men* correlate with the importance of their roles? Differently put, why should it be the case that movies in which women talk about men are the movies in which their roles are less important? In an attempt to automate the Bechdel test, we provide empirical evidence for the negative correlation between *talking about men* and the importance of their roles i.e. movies in which the primary role of female characters is to talk about men are movies in which these characters are not portrayed as leaders and central characters.

In this chapter, we study the effectiveness of various linguistic and social network analysis features for automating the Bechdel test. Our results show that the features based on social network analysis metrics (such as betweenness centrality) are most effective for automating the Bechdel test. More specifically, in movies that fail the test, women are significantly less centrally connected as compared to movies that pass the test. This finding provides support for the long held belief that women are seldom portrayed as strong leaders and thinkers in popular media. Our results also show that word unigrams, topic modeling features, and features that capture mentions of men in conversations are less effective. This may appear

to be a rather surprising result since the question, *(T3) do these women talk to each other about something besides a man?* seems to be one that linguistic features should be able to answer. For example, one might expect that by simply noting the presence of mentions of men in conversations between women might be a good indicator of whether or not they talk *about* men. We found this not to be the case. To give the reader an intuition of why this may not be the case, we provide the following explanation.

Consider the screenplay excerpt in Figure 11.1 (on the next page). This excerpt is from the movie *Hannah and Her Sisters*, which passes the Bechdel test. Even though the conversation between named women **Mickey** and **Gail** mentions a man (**He**), the conversation is not *about* a man. The conversation is about *Mickey's* brain tumor. Now consider the following (contrived) conversation between the same characters:

> **Mickey**: Ssssss, if i'm in love, I don't know what I'm gonna do.
> **Gail**: You're not in love. Didn't he tell you that it was over.
> **Mickey**: No, naturally

This conversation is clearly about a man (or being in love with a man). Much like the original conversation, this conversation mentions a man only once. The linguistic phenomena that allows us to infer that this contrived conversation is about a man is quite complex; it requires a deeper semantic analysis and world knowledge. First, we need to infer that *it being over* refers to a relationship. Relationships typically have two participants. In order to identify the participants, we need to use world knowledge that relationships can end and that the person ending the relationship was once part of the relationship, and so on. Eventually, we are able to conclude that one of the main participants of the conversation or the event being discussed is a man.

As a first attempt to automate the test, we only experiment with simple linguistic features. However, we believe that the task itself offers an opportunity for the development of— and subsequent evaluation of— rich linguistic features that may be better equipped for determining the *aboutness* of conversations. More specifically, determining whether or not a conversation is about a man.

The rest of the chapter is structured as follows. Section 11.2 reviews the related literature.

Section 11.3 describes the data and gold standard used for the purposes of automating the test. Sections 11.4, 11.5, and 11.6 present our approach, evaluation, and results for the three Bechdel tests respectively. We conclude and present future direction for research in Section 11.8.

11.2 Related Work

There has been much work in the computational sciences community on studying gender differences in the way language is used by men versus women [Peersman *et al.*, 2011; Mohammad and Yang, 2011b; Bamman *et al.*, 2012; Schwartz *et al.*, 2013; Bamman *et al.*, 2014a; Prabhakaran *et al.*, 2014]. In fact, researchers have proposed linguistic features for supervised classifiers that predict the gender of authors given their written text [Koppel *et al.*, 2002; Corney *et al.*, 2002; Cheng *et al.*, 2011; Rosenthal, 2015]. There has also been a growth in research that utilizes computational techniques and big data for quantifying gender biases in society [Sugimoto *et al.*, 2013; Garcia *et al.*, 2014; Wagner *et al.*, 2015].

More closely related to our application is the ongoing work in the social sciences community regarding the study of gender biases in movie scripts and books [Weitzman *et al.*, 1972; Clark *et al.*, 2003; Gooden and Gooden, 2001; McCabe *et al.*, 2011; Chick and Corle, 2012; Smith *et al.*, 2013]. This work has largely depended on manual effort. McCabe *et al.* [2011] analyze the presence of male and female characters in titles, and their centralities, in 5,618 children's books. The authors employ multiple human coders for obtaining the relevant annotations. Smith *et al.* [2013] employ 71 research assistants to evaluate 600 films to study gender prevalence in their scripts. Our work offers computational techniques that may help reduce the manual effort involved in carrying out similar social science studies.

Recently, Garcia *et al.* [2014] use 213 movie screenplays for evaluating the correlation of two novel scores with whether or not movies passed the Bechdel test. However, the main focus of their work is not to automate the test. The focus of their work is to study gender biases in MySpace and Twitter (using these scores). Nonetheless, we experiment with these scores and in fact they provide a strong baseline for automating the task. Furthermore, we use a larger set of 457 screenplays for the study (larger than the dataset of 213 screenplays

```
M|                                                              CUT TO:
 |
S|              INT. MICKEY'S OFFICE - NIGHT
 |
N|              Gail, wearing her glasses, stands behind a crowded but well-
N|              ordered desk.  Two assistants, a man and a woman, stand around
N|              her.
 |
C|                                    MICKEY
M|                               (turning to Gail,
M|                               gesturing nervously)
D|                         Sssss, if I have a brain tumor, I
D|                         don't know what I'm gonna do.
M|                               (sighing)
 |
C|                                    GAIL
D|                         You don't have a brain tumor.  He
D|                         didn't say you had a brain tumor.
 |
C|                                    MICKEY
M|                               (sighing)
D|                         No, naturally
```

Figure 11.1: A scene from the movie *Hannah and Her Sisters*. The scene shows *one* conversation between two *named women* **Mickey** and **Gail**. Tag S denotes scene boundary, C denotes character mention, D denotes dialogue, N denotes scene description, and M denotes other information.

that Garcia *et al.* [2014] use).

Researchers in the Natural Language Processing (NLP) community have used movie screenplays for a number of different applications. Ye and Baldwin [2008] use movie screenplays for evaluating word sense disambiguation in an effort to automatically generate animated storyboards. Danescu-Niculescu-Mizil and Lee [2011] utilize movie screenplays for studying the coordination of linguistic styles in dialogues. Bamman *et al.* [2013] use movie plot summaries for finding personas of film characters. Srivastava *et al.* [2015a] In Agarwal *et al.* [2014c] we use screenplays for automatically creating the *xkcd* movie narrative charts. In this chapter, we use movie screenplays for yet another novel NLP application: automating the Bechdel test.

	Train & Dev. Set		Test Set	
	Fail	Pass	Fail	Pass
Bechdel Test 1	26	341	5	85
Bechdel Test 2	128	213	32	53
Bechdel Test 3	60	153	15	38
Overall	214	153	52	38

Table 11.1: Distribution of movies for the three tests over the training/development and test sets.

11.3 Data

The website `bechdeltest.com` has reviewed movies from as long ago as 1892 and as recent as 2015. Over the years, thousands of people have visited the website and assigned ratings to thousands of movies: movies that fail the first test are assigned a rating of 0, movies that pass the first test but fail the second test are assigned a rating of 1, movies that pass the second test but fail the third test are assigned a rating of 2, and movies that pass all three tests are assigned a rating of 3. Any visitor who adds a new movie to the list gets the opportunity to rate the movie. Subsequent visitors who disagree with the rating may leave comments stating the reason for their disagreement. The website has a *webmaster* with admin rights to update the visitor ratings. If the webmaster is unsure or the visitor comments are inconclusive, she sets a flag (called the "dubious" flag) to *true*. For example, *niel (webmaster)* updated the rating for the movie *3 Days to Kill* from 1 to 3.[1] The dubious flag does not show up on the website interface but is available as a meta-data field. Over the course of this study, we noticed that the dubious flag for the movie *Up in the Air* changed from false to true.[2] This provides evidence that the website is actively maintained and moderated by its owners.

We crawled a total of 1051 movie screenplays from the Internet Movie Script Database (IMSDB). Out of these, only 457 were assigned labels on `bechdeltest.com`. We decided

[1] `http://bechdeltest.com/view/5192/3_days_to_kill/`

[2] `http://bechdeltest.com/view/578/up_in_the_air/`

to use 367 movies for training and development and 90 movies (about 20%) for testing. The split was done randomly. Table 11.1 presents the distribution of movies that pass/fail the three tests in our training and test sets. The distribution shows that a majority of movies fail the test. In our collection, 266 fail while only 191 pass the Bechdel test.

11.4 Test 1: are there at least two named women in the movie?

A movie passes the first test if there are two or more *named women* in the movie. We experiment with several name-to-gender resources for finding the characters' gender. If, after analyzing all the characters in a movie, we find there are two or more *named women*, we say the movie passes the first test, otherwise it fails the first test.

11.4.1 Resources for Determining Gender

IMDB_GMAP: The Internet Movie Database (IMDB) provides a full list of the cast and crew for movies. This list specifies a one-to-one mapping from character names to the actors who perform that role. Actors are associated with their gender through a meta-data field. Using this information, we create an individual dictionary for each movie that maps character names to their genders.

SSA_GMAP: The Social Security Administration (SSA) of the United States has created a publicly available list of first names given to babies born in a given year, with counts, separated by gender.[3] Sugimoto *et al.* [2013] use this resource for assigning genders to authors of scientific articles. Prabhakaran *et al.* [2014] use this resource for assigning gender to sender and recipients of emails in the Enron email corpus. The authors note that a first name may appear with conflicting genders. For example, the first name *Aidyn* appears 15 times as a male and 15 times as a female. For our purposes, we remove names that appear with conflicting genders from the original list. The resulting resource has 90,000 names, 33,000 with the gender male and 57,000 with the gender female.

STAN_GMAP: In our experiments, we find both IMDB_GMAP and SSA_GMAP to be insufficient. We therefore devise a simple technique for assigning genders to named entities

[3] http://www.ssa.gov/oact/babynames/limits.html

using the context of their appearance. This technique is general (not specific to movie screenplays) and may be used for automatically assigning genders to named characters in literary texts. The technique is as follows: (1) run a named entity coreference resolution system on the text, (2) collect all third person pronouns (*she, her, herself, he, his, him, himself*) that are resolved to each entity, and (3) assign a gender based on the gender of the third person pronouns.

We use Stanford's named entity coreference resolution system [Lee *et al.*, 2013] for finding coreferences. Note that the existing coreference systems are not equipped to resolve references within a conversation. For example, in the conversation between **Mickey** and **Gail** (see Figure 11.1) "He" refers to **Mickey's** doctor, **Dr. Wilkes**, who is mentioned by name in an earlier scene (almost 100 lines before this conversation). To avoid incorrect coreferences, we run the coreference resolution system only on the scene descriptions of screenplays.

11.4.2 Results and Discussion

Our technique for predicting whether or not a movie passes the first test is unsupervised. Given a parsed screenplay – for which we have identified scene boundaries, scene descriptions, character names, and dialogues – we obtain the gender of each character (using one of many name to gender resources). We predict a movie fails the first test if we find less than two named women in the movie. Otherwise, we predict that the movie passes the first test.

Since it is important for us to perform well on both classes (fail and pass), we report the macro-F1 measure; macro-F1 measure weights the classes equally unlike micro-F1 measure [Yang, 1999]. We use training and development dataset for the first test (26 Fail and 341 Pass) presented in Table 11.1 for reporting macro-F1 measure.

Table 11.2 presents the results for using various name to gender mapping resources for the first test. We present the precision (P), recall (R), and F1-measure for each of the two classes – Fail Test 1 and Pass Test 1. The last column of the table presents the macro-F1-measure for the two classes.

The results show that SSA_Gmap performs significantly[4] worse than all the other name-to-gender resources (0.59 versus 0.71, 0.71, and 0.75 macro-F1-measure). One reason

[4]We use McNemars test with $p < 0.05$ to report significance.

	Fail Test 1			Pass Test 1			
Gender Resource	P	R	F1	P	R	F1	Macro-F1
IMDB_GMAP	0.35	0.63	0.45	0.97	0.91	0.94	0.71
SSA_GMAP	0.26	0.21	0.24	0.94	0.95	0.94	0.59
STAN_GMAP	0.22	0.96	0.36	0.996	0.74	0.85	0.71
STAN_GMAP+ IMDB_GMAP	0.52	0.55	**0.54**	0.97	0.96	**0.96**	**0.75**

Table 11.2: Results for **Test 1**: "are there at least two named women in the movie".

is that movies have several named characters whose gender is different from the common gender associated with those names. For example, the movie *Frozen* (released in 2010) has two named women: **Parker** and **Shannon**. According to SSA_GMAP, **Parker** is a male, which leads to an incorrect prediction (fail when the movie actually passes the first test).

The results show that a combination of STAN_GMAP and IMDB_GMAP outperforms all the individual resources by a significant margin (0.75 versus 0.71, 0.59, 0.71 macro-F1-measure). We combine the resources by taking their union. If a name appears in both resources with conflicting genders, we retain the gender recorded in IMDB_GMAP Recall, IMDB_GMAP is a one-to-one map from character name to actor name to gender. IMDB_GMAP has a high precision in terms of predicting gender. The following paragraphs discuss the reasons for why the combination of STAN_GMAP and IMDB_GMAP outperforms the individual resources. Both resources have limitations but when combined together, they complement each other.

Limitations of IMDB_GMAP: The precision of IMDB_GMAP is significantly lower than the precision of STAN_GMAP for the class Pass (0.97 versus 0.996). Note that this precision (precision with which IMDB_GMAP predicts if a movie passes the first test) is different from the precision with which IMDB_GMAP predicts gender. While IMDB_GMAP is precise in predicting gender, we find one problem with IMDB_GMAP that results in its low precision for predicting the Pass class. IMDB_GMAP lists non-named characters (such as **Stewardess**) along with the named characters in the credit list. So while the movie *A Space Odyssey* actually fails the test (it has only one named woman, **Elena**), IMDB_GMAP incorrectly

detects **Stewardess** as another named woman and makes an incorrect prediction. This false positive reduces the precision for the Pass class.

Note that IMDB_GMAP has a lower recall for the Pass class as compared to the combination of STAN_GMAP and IMDB_GMAP (0.91 versus 0.96). This is because certain characters are credited with a name different from the way their names appear in a screenplay. For example, in the screenplay for the movie *Up in the Air*, the character **Karen Barnes** is credited as "Terminated employee". However, in the screenplay she is referred to as **Miss Barnes**. By simply using IMDB_GMAP, we are unable to find the gender of **Karen Barnes** (because she is credited with a different name). However, by using STAN_GMAP, we are automatically able to determine the gender of **Karen Barnes**. This determination helps in predicting correctly that the movie *Up in the Air* passes the first test, thus increasing the recall for the Pass class. Following user comment from `bechdeltest.com` on the movie *Up in the Air* confirms our finding:

> Natalie refers to Karen Barnes as "Miss Barnes" when they first meet. She is also named later. Despite the fact that she's credited as "Terminated employee", she's definitely a named female character.

Limitations of STAN_GMAP: Note that the precision of IMDB_GMAP is significantly higher than the precision of STAN_GMAP for the class Fail (0.35 versus 0.22). This has to do with coverage: STAN_GMAP is not able to determine the gender of a number of characters and predicts fail when the movie actually passes the test. We expected this behavior as a result of being able to run the coreference resolution tool only on the scene descriptions. Not all characters are mentioned in scene descriptions.

The methodology used for finding named women directly impacts the performance of our classifiers on the next two tests. For instance, if a methodology under-predicts the number of named women in a movie, its chances of failing the next two tests increase. In fact, we experimented with all combinations and found the combination STAN_GMAP + IMDB_GMAP to outperform other gender resources for the next two tests. We use the lists of named women and named men generated by STAN_GMAP + IMDB_GMAP for the next two tests. We utilize the list of named men for extracting features for the third test.

11.5 Test 2: Do these women talk to each other?

So far, we have parsed screenplays for identifying character mentions, scene boundaries, and other elements of a screenplay (see Figure 11.1). We have also identified the gender of named characters. For automating the second test (*do these women talk to each other?*), all we need to do is create an interaction network of characters and investigate whether or not two named women in this network interact with one another. We experiment with two techniques for creating the interaction networks of characters: CLIQUE and CONSECUTIVE.

Consider the following sequence of tagged lines in a screenplay: {S1, C1, C2, C3, S2, ...}. S1 denotes the first scene boundary, C1 denotes the first speaking character in the first scene, C2 denotes the second speaking character in the first scene, C3 denotes the third speaking character in the first scene, and S2 denotes the second scene boundary. One way of creating an *interaction* network is to connect all the characters that appear between two scene boundaries with pair-wise links. This technique has previously been proposed by [Weng *et al.*, 2009]. In the running example, since the characters C1, C2, and C3 appear between two scene boundaries (S1 and S2), we connect all the three characters with pair-wise links. We call this the CLIQUE approach. Another way of connecting speaking characters is to connect only the ones that appear consecutively (C1 to C2 and C2 to C3, no link between C1 and C3). We call this the CONSECUTIVE approach.

	Fail Test 2			Pass Test 2			
Network	P	R	F1	P	R	F1	Macro-F1
CLIQUE	0.55	0.20	0.29	0.65	0.92	0.76	0.57
CONSECUTIVE	0.63	0.28	**0.39**	0.67	0.90	**0.77**	**0.62**

Table 11.3: Results for **Test 2**: "do these women talk to each other?"

Both these techniques are unsupervised. We use training and development dataset for the second test (128 Fail and 213 Pass) presented in Table 11.1 for reporting the performance results. Results presented in Table 11.3 show that the CONSECUTIVE approach performs significantly better than the CLIQUE approach.

We investigate the reason for an overall low performance for this test. We find that

C	NICOLE (CONT'D)
D	Uh-oh, here comes the wicked bitch of the
D	west...
N	With that, she steps away, just as Janice walks up.
N	Wesley stands to face his boss, as Janice launches into
N	him...
C	BOSS JANICE
D	Jesus H. Fucking Popscile, you don't have
D	time to get me the differential responses
D	but you got time to chitty-chatty with the intern?
D	Why do I even keep you around Wesley?...

Table 11.4: An example of a screenplay (movie *Wanted*, 2008) in which a scene description divides a long scene into two sub-scenes with a different set of conversing characters. Characters **Nicole** and **Janice** never converse with each other in the movie.

sometimes a scene description divides a scene into two scenes, while other times it does not. For example, consider the sequence of scene boundaries, characters, and scene descriptions: {S1, N1, C1, C2, N2, C3, C4, S2, ...}. While for some scenes N2 divides the scene between S1 and S2 into two scenes (S1-N2 and N2-S2), for other scenes it does not. For the screenplays in which a scene boundary (like N2) divides the scene into two scenes, our CONSECUTIVE approach incorrectly connects the characters C2 and C3 (C2 and C3 should not be connected because they belong to different scenes), which leads to an over-prediction of characters that talk to each other. This reason contributes to the low recall for the Fail class – by virtue of over-predicting who talks to whom, we over-predict the pairs of women who talk to each other and thus over-predict the movies that pass the test.

Table 11.4 provides an example of such a screenplay in which a scene description divides a long scene into two sub-scenes with a different set of characters conversing with each other. In this example, **Nicole** is having a conversation with **Wesley**. **Nicole** notices **Janice** coming and steps away. This is one sub-scene. In the next sub-scene, **Janice** and **Wesley**

have a conversation. **Nicole** and **Janice** never converse. In fact, they never converse in the entire movie. Since our CONSECUTIVE approach connects all pairs of consecutively occurring characters in a scene with a link, it incorrectly connects **Nicole** and **Janice**, leading to the incorrect prediction that this movie passes the second test, when it does not.

11.6 Test 3: Do these women talk to each other about something besides a man?

For the third Bechdel test, we experiment with machine learning models that utilize linguistic features as well as social network analysis features derived from the interaction network of characters.

11.6.1 Feature Set

We consider four broad categories of features: word unigrams (**BOW**), distribution of conversations over topics (**TOPIC**), linguistic features that capture mentions of men in dialogue (**LING**), and social network analysis features (**SNA**). We additionally experiment with the two scores proposed by Garcia *et al.* [2014].

For **BOW**, we collect all the words that appear in conversations between pairs of women and normalize the binary vector by the number of pairs of named women and by the number of conversations they have in a screenplay. **BOW** was a fixed feature vector of length 18,889.

The feature set **LING** consists of the following features: (1) the average length of conversations between each pair of named women (2) the number of conversations between each pair of named women, (3) a binary feature that records if *all* conversations between a particular pair of named women mention a man, and (4) a binary feature that records if *any* conversation between a particular pair of named women mentions a man. For detecting mentions of men, we look for nominals such as he, him, his, and named men as determined by the gender finding technique used in the first test.

Let us denote these four feature vectors by $\{v_1, v_2, v_3, v_4\}$. Note that the length of these feature vectors ($|v_i| \leq \binom{n}{2}$, where n is the number of named women in a movie) may vary from one movie to the other. We convert these variable length vectors into fixed length

vectors of length four by using a function, GET_MIN_MAX_MEAN_STD(VECTOR), that returns the minimum, maximum, mean, and standard deviation for each vector. In all, we have $4 * 4 = 16$ **LING** features for each movie.

In an attempt to capture possible correlations between general topics and conversations in which women talk about men, we experiment with features derived from topic models (**TOPIC**). Our screenplay corpus has multiple instances of conversations that are about men and around the same topic but do not explicitly mention a man. For example, both conversations, *don't we all fall for those pricks?* and *which one did you fall in love with?*, do not mention a man explicitly. However, both these conversations are around the same topic, say *love*. Our corpus also has conversations that mention a man explicitly and are around the same topic *love*: *I'm not in love with him, okay!*. The hope is that if there are certain topics that are highly correlated with "talking about a man", the **TOPIC** features would be useful. For extracting the **TOPIC** features, we train a topic model on all the conversations between named women [Blei *et al.*, 2003; McCallum, 2002]. Before training the topic model, we convert all the mentions of men to a fixed tag "MALE" and all the mentions of women to a fixed tag "FEMALE". For each conversation between every pair of women, we query the topic model for its distribution over the k topics. Since the number of pairs of women and the number of conversations between them may vary from one movie to the other, we take the average of the k-length topic distributions over all conversations of all pairs of women in one movie. We experiment with $k = 2$, 20, and 50 by simply appending the feature vector with these topic distributions. Thus the length of the **TOPIC** feature vector is 72 (2 + 20 + 50).

While the Bechdel test was originally designed to assess the presence of women, it has subsequently been used to comment on the importance of roles of women in movies. But does *talking about men* correlate with the importance of their roles? To study this correlation we experiment with the following set of **SNA** features. We create variable length feature vectors (length equal to number of women) for several social network analysis metrics [Wasserman and Faust, 1994], all appropriately normalized: (1) degree centrality, (2) closeness centrality, (3) betweenness centrality, (4) the number of men a woman is connected to, and (5) the number of other women a woman is connected to. We create two other variable length

feature vectors (length equal to the number of pairs of women) that record (6) the number of men in common between two women[5] and (7) the number of women in common between two women. Consider the sample network of characters from the movie *Up In The Air* in Figure 11.2. The network contains four women (**Julie**, **Kara**, **Alex**, and **Natalie**) and two men (**Ryan**, **Craig Gregory**). Table 11.5 present the values for feature vectors 4 and 5. 11.6 present the values for feature vectors 6 and 7.

We convert these variable length feature vectors to fixed length vectors of length four by using the GET_MIN_MAX_MEAN_STD(VECTOR) function described above. This constitutes $7 * 4 = 28$ of our **SNA** features. We additionally experiment with the following features: (8) the ratio of the number of women to the number of men in the whole movie, (9) the ratio of the number of women to the total number of characters, (10) the percentage of women that form a 3-clique with a man and another woman, (11, 12, 13) the percentage of women in the list of five *main* characters (main based on each of the three notions of centralities), (14, 15, 16) three boolean features recording whether the main character is a women, (17, 18, 19) three boolean features recording whether any woman connects another woman to the main man, and (20, 21, 23) the percentage of women that connect the main man to another woman. In all we have $28 + 15 = 43$ **SNA** features.

11.6.2 Baseline

As a baseline, we experiment with the features proposed by Garcia *et al.* [2014]. The authors propose two scores: B_F and B_M. B_F is the ratio of {dialogues between female characters that did not contain mentions of men} over {the total number of dialogues in a movie}. B_M is the ratio of {dialogues between male characters that did not contain mentions of women} over {the total number of dialogues in a movie}. Garcia *et al.* [2014] do not present an evaluation of exactly with what accuracy they are able to predict whether a movie passes of fails the Bechdel test. However, they report Wilcoxon tests numbers and "show that movies that pass the test have higher B_F by 0.026, (p < 10?9) and lower B_M by 0.051 (p < 10?3)."

[5]Lets say **Sara** has a conversation with three men: **John**, **Paul**, and **Adam**. Lets say **Mary** has a conversation with two men: **John** and *Michael*. Then there is one man in common between the two women, namely **John**.

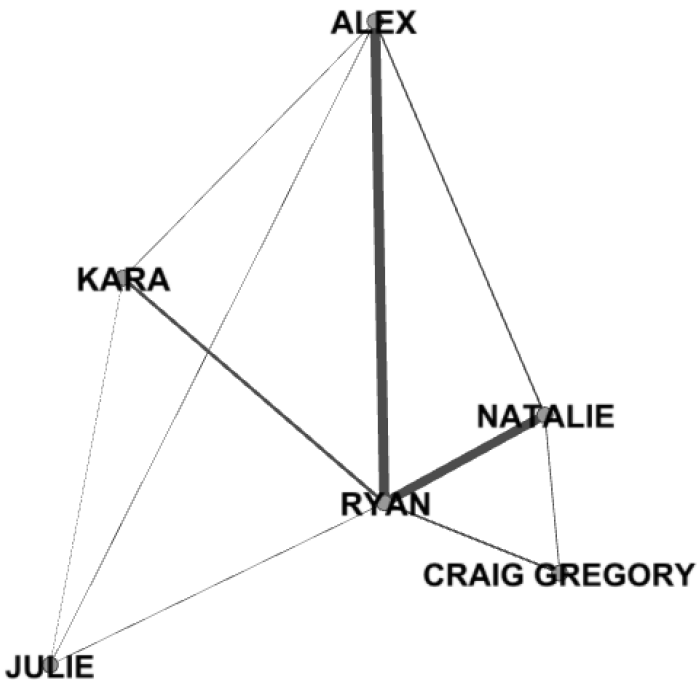


Figure 11.2: The network of the main characters from the movie *Up In The Air*. The set of women = {Julie, kara, Alex, Natalie}. The set of men = {Ryan, Craig Gregory}.

Feature	Julie	Kara	Alex	Natalie
(4) the number of men a woman is connected with	1	1	1	2
(5) the number of women a woman is connected with	2	2	3	1

Table 11.5: Feature values for **SNA** feature vectors (4) and (5).

11.6.3 Evaluation and Results

For evaluation, we use the training and development set for Bechdel Test 3 (see Table 11.1). There are 60 movies that fail and 153 movies that pass the third test. We experiment with Logistic Regression and Support Vector Machines (SVM) with the linear and RBF kernels. Out of these SVM with linear and RBF kernels perform the best. Table 11.7 reports the averaged 5-fold cross-validation F1-measures for the best combinations of classifiers and feature sets. For each fold, we penalize a mistake on the minority class by a factor of 2.55 (153/60), while penalizing a mistake on the majority class by a factor of 1. This is an

Feature	Julie - Kara	Julie - Alex	Julie - Natalie	Kara - Alex	Kara - Natalie	Alex - Natalie
(6) the number of men in common between two women	1 (Ryan)	1 (Ryan)	1 (Ryan)	1 (Ryan)	1 (Ryan)	1 (Ryan)
(7) the number of women in common between two women	1 (Alex)	1 (Kara)	1 (Alex)	1 (Julie)	1 (Alex)	0

Table 11.6: Feature values for **SNA** feature vectors (6) and (7).

important step and as expected has a significant impact on the results. A binary classifier that uses a 0-1 loss function optimizes for accuracy. In a skewed data distribution scenario where F1-measure is a better measure to report, classifiers optimizing for accuracy tend to learn a trivial function that classifies all examples into the same class as the majority class. By penalizing mistakes on the minority class more heavily, we force the classifier to learn a non-trivial function that is capable of achieving a higher F1-measure.

			Fail Test 3			Pass Test 3			Macro
Row #	Kernel	Feature Set	P	R	F1	P	R	F1	F1
1	Linear	Garcia *et al.* [2014]	0.39	0.70	0.50	0.84	0.57	0.67	0.62
2	Linear	**BOW**	0.40	0.37	0.38	0.74	0.76	**0.75**	0.57
3	Linear	**LING**	0.39	0.37	0.37	0.75	0.76	**0.75**	0.57
4	Linear	**TOPIC**	0.28	0.29	0.27	0.71	0.70	0.70	0.50
5	RBF	**SNA**	0.42	0.84	**0.56**	0.90	0.55	0.68	**0.68**

Table 11.7: Results for **Test 3**: "do these women talk to each other about something besides a man?" Column two specifies the kernel used with the SVM classifier.

Table 11.7 presents the results for using various kernels and feature combinations for the third test. We present the precision (P), recall (R), and F1-measure for each of the two classes – Fail Test 3 and Pass Test 3. The last column of the table presents the macro-F1-

measure for the two classes.

Results in Table 11.7 show that the features derived from social network analysis metrics (**SNA**) outperform linguistic features (**BOW**, **LING**, and **TOPIC**) by a significant margin. **SNA** features also outperform the features proposed by Garcia *et al.* [2014] by a significant margin (0.68 versus 0.62). Various feature combinations did not outperform the **SNA** features. In fact, all the top feature combinations that perform almost as well as the **SNA** features include **SNA** as one of the feature sets.

11.6.4 Discussion

In this section, we present a correlation analysis of our **SNA** features and of the features proposed by Garcia *et al.* [2014] with the gold class on the set of 183 movies that pass or fail the third test in our training set. The most correlated **SNA** feature is the one that calculates the percentage of women who form a 3-clique with a man and another woman ($r = 0.34$). Another highly correlated **SNA** feature is the binary feature that is true when the main character is a woman in terms of betweenness centrality ($r = 0.32$). Several other **SNA** features regarding the different notions of centralities of women are among the top. The feature suggested by Garcia *et al.* [2014], B_F and B_M, are also significantly correlated, with $r = 0.27$ and $r = -0.23$ respectively.

Figure 11.3 shows the distribution of three of our **SNA** features: mean degree centrality, mean closeness centrality, and mean betweenness centrality of named women. The x-axis is the feature value and the y-axis is the number of examples that have a particular feature value. The blue histogram is for movies that pass the third test and the red histogram is for movies that fail the third test. As the distributions show, most of the mass for movies that fail the test (red histogram) is towards the left of the plot, while most of the mass for movies that pass (blue histogram) is towards the right. So movies that fail the test tend to have lower centrality measures as compared to movies that pass the test. Using our classification results, correlation analysis, and visualizations of the distributions of the **SNA** features, we conclude that, in fact, movies that fail the test are likely to have less centrally connected women.

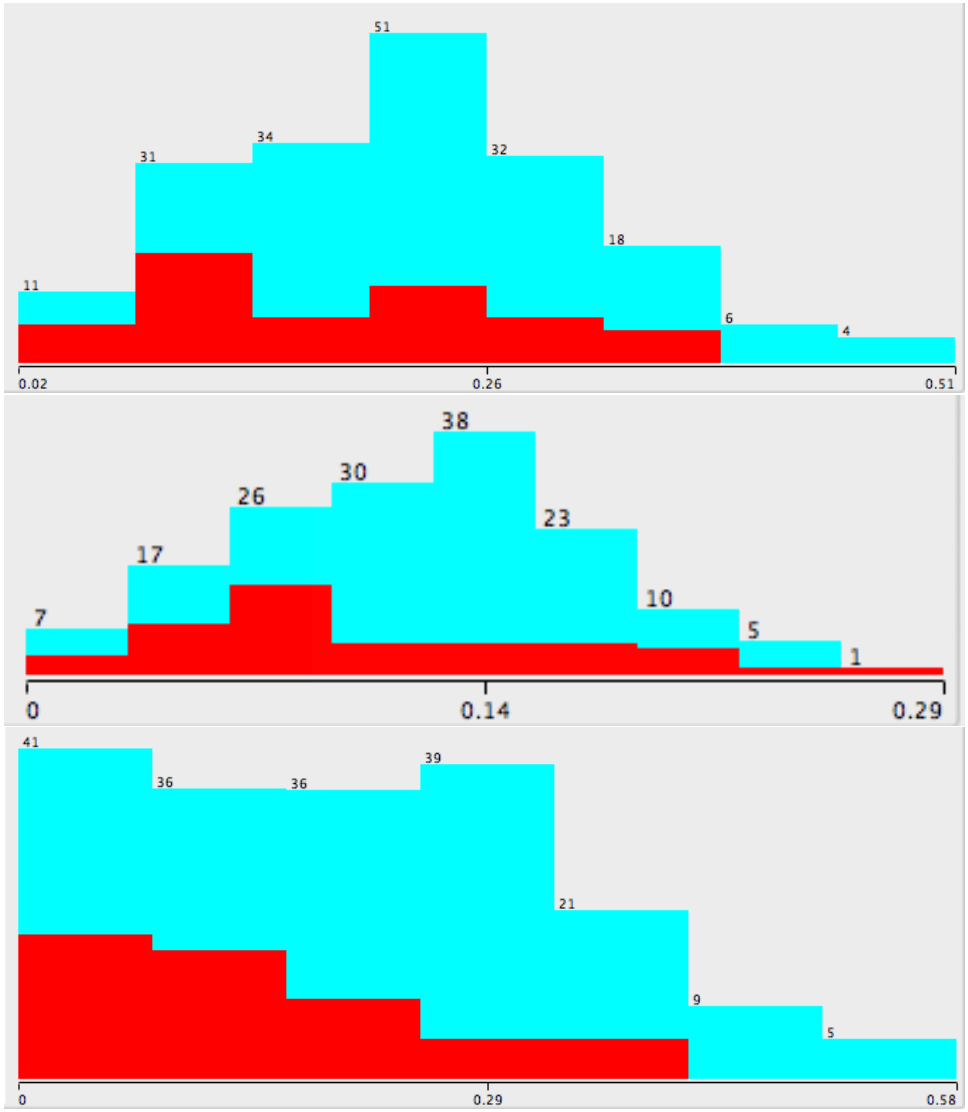


Figure 11.3: Distribution of three **SNA** features (top to bottom): mean degree centrality, mean closeness centrality, and mean betweenness centrality of named women. Red histogram is for movies that fail and the Blue histogram is for movies that pass the third Bechdel Test. The histograms show that the average centralities of women are higher for movies that pass the Bechdel test.

11.7 Evaluation on the End Task

We use the IMDB_GMAP + STAN_GMAP gender resource for the first test, the CONSECUTIVE approach for creating an interaction network for the second test, and the machine learning

model trained on **SNA** features for the third test. We compare the performance of our system with the baseline features suggested by Garcia *et al.* [2014].

		Fail Test 3			Pass Test 3			Macro
Kernel	Feature	P	R	F1	P	R	F1	Macro-F1
Linear	Garcia *et al.* [2014]	0.72	0.93	0.81	0.81	0.47	0.60	0.73
RBF	**SNA**	0.80	0.91	**0.85**	0.83	0.66	**0.73**	**0.80**

Table 11.8: Results on the unseen test set on the end task: does a movie passes the Bechdel Test?

Table 11.8 presents the results for the evaluation on our unseen test set of 52 movies that fail and 38 movies that pass the Bechdel test (see Table 11.1). As the results show, our best feature and classifier combination outperforms the baseline by a significant margin (0.73 versus 0.80). Note that the end evaluation is easier than the evaluation of each individual test. Consider a movie that fails the first test (and thus fails the Bechdel test). At test time, lets say, the movie is mis-predicted and passes the first two tests. However, the classifier for the third test correctly predicts the movie to fail the Bechdel test. Even though the errors propagate all the way to the third level, these errors do not affect the evaluation metric on the end task.

11.8 Conclusion and Future Work

In this chapter, we introduced a novel NLP task of automating the Bechdel test. We utilized and studied the effectiveness of a wide range of linguistic features and features derived from social network analysis metrics for the task. Our results revealed that the question, *do women talk to each other about something other than a man*, is best answered by network analysis features derived from the interaction networks of characters in screenplays. We were thus able to show a significant correlation between the importance of roles of women in movies with the Bechdel test. Indeed, movies that fail the test tend to portray women as less-important and peripheral characters.

To the best of our knowledge, there is no large scale empirical study on quantifying the

percentage of children's books and novels that fail the Bechdel test. In the future, we hope to combine the ideas from this work with our work on social network extraction from literary texts (Chapter 4) for presenting a large scale study on children's book and novels.

Part V

Conclusions

In the pre-digital age, when electronically stored information was non-existent, the only ways of creating and studying social networks were by hand through surveys, interviews, and observations. In this digital age of the internet, numerous indications of social interactions and associations are available electronically in an easy to access manner as structured meta-data. This lessens our dependence on manual surveys and interviews for creating and studying social networks. However, there are sources of networks that remain untouched simply because they are not associated with any meta-data. Primary examples of such sources include the vast amounts of literary texts, news articles, content of emails, and other forms of unstructured and semi-structured texts.

The main contribution of this thesis is the introduction of natural language processing and applied machine learning techniques for uncovering social networks in such sources of unstructured and semi-structured texts. Specifically, we proposed three novel techniques for mining social networks from three types of texts: unstructured texts (such as literary texts), emails, and movie screenplays. For each of these types of texts, we demonstrated the utility of the extracted networks on three applications (one for each type of text). Following is a summary of the main contributions in more detail.

11.9 Summary of Contributions

- In Chapter 2, we introduced a new kind of social network – a network in which nodes are people and links were *social events*. Two entities (of type person) were said to participate in a social event if at least one of the entities is cognitively aware of the other. We defined two broad categories of social events: observations (OBS) and interactions (INR). The OBS social event has two subcategories: OBS.Near and OBS.Far (see Section 2.3 for precise definitions). The INR social event has four subcategories: INR.Verbal.Near, INR.Verbal.Far, INR.Non-Verbal.Near, and INR.Non-Verbal.Far. One of the main contributions of this thesis is the introduction of the notion of social events and its taxonomy. Our notion of social events grounds the definition of social networks in the most basic building blocks of relationships – cognition. We claim that social events are the smallest possible, the most

rudimentary building blocks for more complex social relationships such as friendships. People have to be cognitively aware of each other for building and maintaining complex social relations. We hope that our nomenclature serves as a unifying definitional platform for other types social networks.

- In Chapter 4, we introduced a supervised methodology for automatically extracting social event networks from unstructured texts such as news articles and literary texts (SINNET). We took motivation from the relation extraction community and used subsequence and tree convolution kernels in conjunction with Support Vector Machines for training our models. We created several baselines, also motivated from past literature, and showed that convolution kernels are well-equipped at adapting to a new task. In fact, SINNET is now being used in the DEFT project at Columbia University for an entirely new task of source-and-target belief and sentiment detection.

- Elson *et al.* [2010] previously introduced the task of computationally validating literary theories that assumed a structural difference between the social worlds of rural and urban novels using *conversational* networks extracted from nineteenth-century British novels. In Chapter 5, we employed SINNET for extracting interactional links (a conceptual generalization of conversational links) and a new class of links called observational links. This allowed us to examine a wider set of literary hypotheses and provide deeper insights into some of the long standing literary theories.

- In Chapter 7, we introduced a novel unsupervised technique for resolving named mentions in emails to real people in the organization. This allowed us to extract the mention network – a new kind of network that has not been explored for any applications in the past.

- In Chapter 8, we utilized the mention network for predicting dominance relations between employees and showed that it performs better than the more commonly used email network. Through a comprehensive set of experiments, we provided evidence for a new finding about the Enron corpus – *you're the boss if people get mentioned to you*. We found that people who receive emails that contain a lot of mentions to other people are the boss. We believe this finding may be attributed to the corporate

reporting culture in which managers report to their superiors about the performance of their team (thus mentioning a high volume of people in the emails to their superiors).

- We introduced the first NLP and ML based system for extracting social networks from movie screenplays. Our method outperforms the previously proposed regular expression and grammar based methods by large and significant margins. There has been a growth in the number of applications that make use of parsed screenplays and film summaries [Turetsky and Dimitrova, 2004; Smith *et al.*, 2013; Bamman *et al.*, 2014b; Gorinski and Lapata, 2015b; Groza and Corde, 2015; Srivastava *et al.*, 2015b]. We believe that the availability of well-parsed screenplays may have a positive impact on these applications. Furthermore, the models we proposed may be applied for extracting networks from other types of screenplays such as drama and theatrical play scripts.

- One of the main challenges in building a system for automatically parsing screenplays (which is required for extracting a social network) was the absence of training data. We proposed a methodology for automatically obtaining a large and varied sample of annotated screenplays that required minimal human intervention. For different types of anomalies (in the structure of screenplays), we *perturbed* the training data and trained separate classifiers that were experts in handling certain combinations of anomalies. We combined these experts into one classifier using ensemble learning techniques. We believe that our general technique may be applied for automatically parsing other types of documents that are supposed to be well-structured but are not, for example, emails that are converted to text using optical character recognition techniques.

- The Bechdel Test is a sequence of three questions designed to assess the presence of women in movies. Many believe that because women are seldom represented in film as strong leaders and thinkers, viewers associate weaker stereotypes with women. We presented the first computational approach for automating the task of finding whether or not a movie passes the Bechdel Test. This automation allowed us to study the key differences in the importance of roles of women in movies that pass the test versus the

movies that fail the test. Our experiments confirmed that in movies that fail the test, women are in fact portrayed as less-central or less-important characters.

11.10 New Datasets

- ACE-2005 Social Event Annotations: For training and testing our models for SINNET, we annotated a well-known and widely distributed corpus by the Linguistic Data Consortium (LDC) called the Automatic Content Extraction (ACE) 2005 Multilingual Training Corpus.[6] The ACE-2005 corpus contains annotations for entities, entity mentions, ACE relations, and ACE events. The data sources in the corpus come from weblogs, broadcast news, newsgroups, broadcast conversation. We overlay our social event annotations onto the dataset and make it available for download in LDC's standard offset annotation format.

- Enron Organizational Hierarchy Corpus: Through this work, we introduce the largest known gold standard for both dominance and hierarchy relations of Enron employees. Previously used gold standards contained dominance relations of only 158 Enron employees. The gold standard we introduce contains dominance and hierarchy relations of 1518 Enron employees.[7].

11.11 Limitations and Future Work

- As mentioned earlier, we annotated the ACE-2005 corpus for social events. There were several advantages of annotating this corpus: (1) the corpus is widely used and distributed by a well-know data consortium, the LDC, (2) the corpus already contained annotations for entity mentions (social event annotations require that the entity mentions have already been annotated), and (3) the corpus had several sources of journalistic texts such as weblogs, broadcast news, newsgroups, broadcast conversation. However, we found one limitation of annotating this corpus – subcategories of some of

[6] `https://catalog.ldc.upenn.edu/LDC2006T06`. LDC Catalog number: LDC2006T06

[7] The corpus may be downloaded from `http://www1.ccls.columbia.edu/~rambow/enron/`

the social events were in significant minority. Specifically, we found only two instances of the OBS.NEAR social event as compared to 110 instances of the OBS.FAR social event. Similarly, we found only 17 instances of the INR.NON-VERBAL social event as compared to 83 instances of the INR.VERBAL social event.[8] It is conceivable that journalistic text does not contain many instances of social events such as *glimpsing* at someone (an OBS.NEAR social event) or *gazing into someone's eyes* (an INR.NON-VERBAL.NEAR social event). Due to this limitation, we were unable to build models for classifying social events into these subcategories. In the future, we will use our annotation manual for annotating genres of texts that may contain a higher frequency of such social events. Since our machine learning approach is task independent, we believe that once we have the annotations, we would be able to re-train SINNET to identify these subcategories rather easily.

- We primarily used a convolution kernel based approach for automating the detection and classification of social events. This approach was motivated by the approaches used in the relation extraction community. We proposed novel data representations that incorporated frame semantics. Conceptually, we found the use of frame semantics appealing (see Section 4.2.5 for details). While frame semantic kernels helped in achieving the best performance for the social event detection task, we did not see an overall gain in performance for the end task (social network extraction). There could be two reasons for this: (1) we did not come up with the right data representations and/or (2) Semafor's performance is hampered by the sparsity of the training data it uses from FrameNet. The sparsity of FrameNet has been reported to be a problem by other researchers [Shi and Mihalcea, 2005; Johansson and Nugues, 2007]. In the future, we will explore and incorporate the advancements in distributional semantics to overcome the sparsity of FrameNet.

- In Chapter 5, we used SINNET for extracting interaction and observation networks from nineteenth century British literature. This allowed us to study a set of literary hypotheses that was broader than the set of hypotheses that were studied by Elson *et*

[8] We reported these numbers on a sample of 62 news articles from the ACE corpus.

al. [2010] using conversational networks. Even though we succeeded in broadening the scope of the theories that could be validated using computational techniques, there remain aspects of these theories that we could not validate. For example, Elson *et al.* [2010] cite Eagleton:

> the city is where "most of our encounters consist of seeing rather than speaking, glimpsing each other as objects rather than conversing as fellow subjects" [Eagleton, 2005, p. 145]

This suggests that a version of SINNET that is able to differentiate between verbal and non-verbal interactions may be used to validate more aspects of these theories.

- In Chapter 7, we proposed a novel unsupervised algorithm for resolving named mentions in the content of emails to real people in the corpus. Our approach was based on minimizing the joint distance between the sender and the recipients to the candidates of the mentioned person. In Section 7.3.4 we showed that 92.6% of the total errors that our name disambiguation technique made were due to the inability of the algorithm to find one winning candidate; the algorithm found multiple candidates at the same joint distance from the sender and the recipients. In the same section, we also showed that about 43% of the times, there were many candidates at the same distance because of the entity normalization problem. For the remaining 57% cases, we believe that factors other than shortest paths may be helpful. In the future, we will explore incorporating factors such as recency of communications and volume of communications between the sender, the recipients, and the candidates of a mention. We will also consider resolving all the mentions in the email jointly rather than independently. For example, if *John*, *Sara*, and *Zhane* (an uncommon first name) are mentioned in the same email, and if there is a community of people consisting people with these three first names, we may want to resolve the mention of *John* to the **John** in this community.

- In Chapter 8, we introduced the largest known gold standard for hierarchy prediction of Enron employees. One of the limitations of our work, along with all of existing work, is that we are concerned with predicting organizational dominance and not the

hierarchy itself. Predicting hierarchy has two sub-tasks: (1) predicting if two people are in the same managerial lineage and (2) predicting who is the boss of whom. To the best of our knowledge, the work to date (including ours) on the Enron email corpus tackles the second sub-task but not the first. In the future, we will attempt to predict the organizational hierarchy.

- One of our main findings from Chapter 8 was that the number of people being mentioned to a person is a good predictor of dominance relations (*you're the boss if people get mentioned to you*). However, we do not know if this is a characteristic of the Enron email corpus or if this finding generalizes to other corpora. In the future, we will apply our techniques on more recently released email datasets such as the Avocado Research Email Collection [Oard *et al.*, 2015] to validate the generalizability of the finding.

- In Chapter 10, we proposed a novel technique for automatically parsing movie screenplays. Even though our approach was able to handle the most common types of anomalies, it is not trained to handle one other type of anomaly – missing newline characters. Our current approach works at the line level. It assumes that different elements of the screenplay are on different lines. However, it may be the case that different elements appear on the same line. For example, character name appears on the same line as the character dialogue ("GAIL: You don't have a brain tumor. He didn't say you had a brain tumor."). For handling such cases, we will experiment with a sequence labeling classifier such as Conditional Random Fields (CRFs, [Lafferty *et al.*, 2001]). Our general methodology for preparing training data for CRFs and our general framework for combining experts at dealing with a combination of anomalies would still be applicable.

- In Chapter 11, we introduced a technique for automating the Bechdel Test. One of the core elements of finding whether or not a movie passes the test is to find if two named women talk *about* a man. As a first attempt to automate the test, we only experimented with simple linguistic features. We showed that recording the presence of male mentions in a conversation was insufficient for determining whether or not a conversation was about a man. We also showed that topic models were insufficient for

determining if the topic of a conversation was about a man. We believe that the task of finding aboutness of conversations offers an opportunity for the development of— and subsequent evaluation of— rich linguistic features that may be better equipped for determining the *aboutness* of texts. There has been some recent work on determining the aboutness of web documents [Gamon *et al.*, 2013]. However, to the best of our knowledge, there has been no work on finding the aboutness of conversations or other kinds of unstructured texts.

11.12 Other Future Work

- Large scale study of gender bias in literature: To the best of our knowledge, there has been no large scale empirical study on quantifying the percentage of children's books and novels that pass the Bechdel Test. In the future, we will try to combine the ideas from our work on automating the Bechdel Test with SINNET. We will use SINNET for extracting an interaction network of characters from literature and use techniques developed in Chapter 11 for determining the percentage of texts that pass the Bechdel Test.

- SINNET + Enron: We ran SINNET on the content of emails to mine social event links between people mentioned in the content of emails. These additional links, however, did not contribute to the performance of our method on the task of dominance prediction. In the future, we will try to utilize these links for other applications. One potential application on our radar is the automatic determination of insider trading traces. While delivering insider information to an outsider, people may not interact directly or through obvious means. They, however, might describe their social interactions in the content of electronic media (intentionally for some other reason or unintentionally).

Part VI

Bibliography

Bibliography

[Agarwal and Rambow, 2010] Apoorv Agarwal and Owen Rambow. Automatic detection and classification of social events. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1024–1034, Cambridge, MA, October 2010. Association for Computational Linguistics.

[Agarwal *et al.*, 2010] Apoorv Agarwal, Owen C. Rambow, and Rebecca J. Passonneau. Annotation scheme for social network extraction from text. In *Proceedings of the Fourth Linguistic Annotation Workshop*, 2010.

[Agarwal *et al.*, 2012] Apoorv Agarwal, Adinoyi Omuya, Aaron Harnly, and Owen Rambow. A comprehensive gold standard for the enron organizational hierarchy. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–165, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[Agarwal *et al.*, 2013] Apoorv Agarwal, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. Sinnet: Social interaction network extractor from text. In *Sixth International Joint Conference on Natural Language Processing*, page 33, 2013.

[Agarwal *et al.*, 2014a] Apoorv Agarwal, Sriramkumar Balasubramanian, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. Frame semantic tree kernels for social network extraction from text. *14th Conference of the European Chapter of the Association for Computational Linguistics*, 2014.

[Agarwal *et al.*, 2014b] Apoorv Agarwal, Sriramkumar Balasubramanian, Jiehan Zheng, and Sarthak Dash. Parsing screenplays for extracting social networks from movies. *EACL-CLFL 2014*, pages 50–58, 2014.

[Agarwal *et al.*, 2014c] Apoorv Agarwal, Daniel Bauer, and Owen Rambow. Using frame semantics in natural language processing. *ACL 2014*, 1929:30–33, 2014.

[Agarwal *et al.*, 2014d] Apoorv Agarwal, Adinoyi Omuya, Jingwei Zhang, and Owen Rambow. Enron corporation: YouâĂŹre the boss if people get mentioned to you. *In Proceedings of The Seventh ASE International Conference on Social Computing (SocialCom)*, 2014.

[Agarwal *et al.*, 2015] Apoorv Agarwal, Jiehan Zheng, Shruti Vasanth Kamath, Sriram Balasubramanian, and Shirin Ann Dey. Key female characters in film have more to talk about besides men: Automating the bechdel test. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, pages 830–840, 2015.

[Agarwal, 2011] Apoorv Agarwal. Social network extraction from texts: A thesis proposal. In *Proceedings of the ACL 2011 Student Session*, pages 111–116, Portland, OR, USA, June 2011. Association for Computational Linguistics.

[Agichtein and Gravano, 2000] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM, 2000.

[Almack, 1922] John C. Almack. The influence of intelligence on the selection of associates. *School and Society*, 16:529–530, 1922.

[Baker *et al.*, 1998] C. Baker, C. Fillmore, and J. Lowe. The berkeley framenet project. *Proceedings of the 17th international conference on Computational linguistics*, 1, 1998.

[Bakhtin, 1937] Mikhail M Bakhtin. Forms of time and of the chronotope in the novel: Notes toward a historical poetics. *Narrative dynamics: Essays on time, plot, closure, and frames*, pages 15–24, 1937.

[Bamman *et al.*, 2012] David Bamman, Jacob Eisenstein, and Tyler Schnoebelen. Gender in twitter: Styles, stances, and social networks. *arXiv preprint arXiv:1210.4567*, 2012.

[Bamman *et al.*, 2013] David Bamman, Brendan O'Connor, and Noah A. Smith. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 352–361, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[Bamman *et al.*, 2014a] David Bamman, Jacob Eisenstein, and Tyler Schnoebelen. Gender identity and lexical variation in social media. *Journal of Sociolinguistics*, 18(2):135–160, 2014.

[Bamman *et al.*, 2014b] David Bamman, Brendan O'Connor, and Noah A Smith. Learning latent personas of film characters. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, page 352, 2014.

[Banko *et al.*, 2007] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[Bechdel, 1986] Alison Bechdel. *Dykes to watch out for*. Firebrand Books, 1986.

[Bekkerman and McCallum, 2005] Ron Bekkerman and Andrew McCallum. Disambiguating web appearances of people in a social network. In *Proceedings of the 14th international conference on World Wide Web*, pages 463–470. ACM, 2005.

[Bekkerman *et al.*, 2005] Ron Bekkerman, Ran El-Yaniv, and Andrew McCallum. Multi-way distributional clustering via pairwise interactions. In *Proceedings of the 22nd international conference on Machine learning*, pages 41–48. ACM, 2005.

[Bhattacharya and Getoor, 2007] Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):5, 2007.

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[Borgatti and Cross, 2003] Stephen P. Borgatti and Rob Cross. A relational view of information seeking and learning in social networks. *Management science*, 2003.

[Boschee *et al.*, 2005] Elizabeth Boschee, Ralph Weischedel, and Alex Zamanian. Automatic information extraction. In *Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, VA*, pages 2–4, 2005.

[Bramsen *et al.*, 2011] Philip Bramsen, Martha Escobar-Molano, Ami Patel, and Rafael Alonso. Extracting social power relationships from natural language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 773–782, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[Brandes *et al.*, 2001] U. Brandes, J. Raab, and D. Wagner. Exploratory network visualization: Simultaneous display of actor status and connections. *Journal of Social Structure*, 2001.

[Brin, 1999] Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. Springer, 1999.

[Bunescu and Mooney, 2005] Razvan C Bunescu and Raymond J 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.

[Bunescu and Mooney, 2007] Razvan Bunescu and Raymond Mooney. Learning to extract relations from the web using minimal supervision. In *Annual meeting of the Association for Computational Linguistics*, volume 45, page 576, 2007.

[Burges, 1998] Chris Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 1998.

[Carenini *et al.*, 2007] Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, pages 91–100. ACM, 2007.

[Chan and Roth, 2010] Yee Seng Chan and Dan Roth. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 152–160. Association for Computational Linguistics, 2010.

[Chan and Roth, 2011] Yee Seng Chan and Dan Roth. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[Chen and Martin, 2007] Ying Chen and James Martin. Towards robust unsupervised personal name disambiguation. In *EMNLP-CoNLL*, pages 190–198. Citeseer, 2007.

[Chen *et al.*, 2010] Desai Chen, Nathan Schneider, Dipanjan Das, and Noah A. Smith. Semafor: Frame argument resolution with log-linear models. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 264–267, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[Cheng *et al.*, 2011] Na Cheng, Rajarathnam Chandramouli, and KP Subbalakshmi. Author gender identification from text. *Digital Investigation*, 8(1):78–88, 2011.

[Chick and Corle, 2012] Kay Chick and Stacey Corle. A gender analysis of ncss notable trade books for the intermediate grades. *Social Studies Research and Practice*, 7(2):1–14, 2012.

[Clark *et al.*, 2003] Roger Clark, Jessica Guilmain, Paul Khalil Saucier, and Jocelyn Tavarez. Two steps forward, one step back: The presence of female characters and gender stereotyping in award-winning picture books between the 1930s and the 1960s. *Sex roles*, 49(9-10):439–449, 2003.

[Cohen, 1960] Jacob Cohen. A coeffiecient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.

[Collins and Duffy, 2002] Michael Collins and Nigel Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings*

of the 40th annual meeting on association for computational linguistics, pages 263–270. Association for Computational Linguistics, 2002.

[Cormack and Lynam, 2005] Gordon V Cormack and Thomas R Lynam. Trec 2005 spam track overview. In *TREC*, 2005.

[Corney *et al.*, 2002] Malcolm Corney, Olivier de Vel, Alison Anderson, and George Mohay. Gender-preferential text mining of e-mail discourse. In *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 282–289. IEEE, 2002.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[Craven *et al.*, 1999] Mark Craven, Johan Kumlien, et al. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, volume 1999, pages 77–86, 1999.

[Creamer *et al.*, 2009] Germán Creamer, Ryan Rowe, Shlomo Hershkop, and Salvatore J Stolfo. Segmentation and automated social hierarchy detection through email network analysis. In *Advances in Web Mining and Web Usage Analysis*, pages 40–58. Springer, 2009.

[Cross *et al.*, 2001] Rob Cross, Andrew Parker, and Laurence Prusak. Knowing what we know:-supporting knowledge creation and sharing in social networks. *Organizational Dynamics*, 2001.

[Cucerzan, 2007] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL*, volume 7, pages 708–716, 2007.

[Culotta and Sorensen, 2004] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 423–429, Barcelona, Spain, July 2004.

[Danescu-Niculescu-Mizil and Lee, 2011] Cristian Danescu-Niculescu-Mizil and Lillian Lee. Chameleons in imagined conversations: A new approach to understanding coordination

of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87. Association for Computational Linguistics, 2011.

[Day *et al.*, 2004] David Day, Chad McHenry, Robyn Kozierok, and Laurel Riek. Callisto: A configurable annotation workbench. *International Conference on Language Resources and Evaluation*, 2004.

[Diehl *et al.*, 2006] C. Diehl, L. Getoor, and G. Namata. Name reference resolution in organizational email archives. *SIAM International Conference on Data Mining*, 2006.

[Diehl *et al.*, 2007] Christopher Diehl, Galileo Mark Namata, and Lise Getoor. Relationship identification for social network discovery. *AAAI '07: Proceedings of the 22nd National Conference on Artificial Intelligence*, 2007.

[Diesner *et al.*, 2005] Jana Diesner, Terrill L Frantz, and Kathleen M Carley. Communication networks from the enron email corpus âĂIJit's always about the people. enron is no differentâĂİ. *Computational & Mathematical Organization Theory*, 11(3):201–228, 2005.

[Ding *et al.*, 2002] Jing Ding, Daniel Berleant, Dan Nettleton, and Eve Syrkin Wurtele. Mining medline: abstracts, sentences, or phrases? In *Pacific Symposium on Biocomputing*, volume 7, pages 326–337. World Scientific, 2002.

[Do and Roth, 2010] Quang Do and Dan Roth. On-the-fly constraint based taxonomic relation identification. Technical report, University of Illinois, 2010.

[Doddington *et al.*, 2004] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The automatic content extraction (ace) program–tasks, data, and evaluation. *LREC*, pages 837–840, 2004.

[Downey *et al.*, 2006] Doug Downey, Oren Etzioni, and Stephen Soderland. A probabilistic model of redundancy in information extraction. Technical report, DTIC Document, 2006.

[Dredze *et al.*, 2010] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd*

International Conference on Computational Linguistics, pages 277–285. Association for Computational Linguistics, 2010.

[Eagleton, 1996] Terry Eagleton. *Literary theory: An introduction*. U of Minnesota Press, 1996.

[Eagleton, 2005] Terry Eagleton. *The English novel: an introduction*. John Wiley & Sons, 2005.

[Eagleton, 2013] Terry Eagleton. *The English novel: an introduction*. John Wiley & Sons, 2013.

[Elson and McKeown, 2010] David K Elson and Kathleen McKeown. Automatic attribution of quoted speech in literary narrative. In *AAAI*, 2010.

[Elson *et al.*, 2010] David K. Elson, Nicholas Dames, and Kathleen R. McKeown. Extracting social networks from literary fiction. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, 2010.

[Elson, 2012] David Elson. *Modeling Narrative Discourse*. PhD thesis, Columbia University, 2012.

[Etzioni *et al.*, 2005] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[Fan *et al.*, 2011] Xiaoming Fan, Jianyong Wang, Xu Pu, Lizhu Zhou, and Bing Lv. On graph-based name disambiguation. *Journal of Data and Information Quality (JDIQ)*, 2(2):10, 2011.

[Fan *et al.*, 2014] Miao Fan, Deli Zhao, Qiang Zhou, Zhiyuan Liu, Thomas Fang Zheng, and Edward Y Chang. Distant supervision for relation extraction with matrix completion. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 839–849, 2014.

[Fellegi and Sunter, 1969] Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[Finkel *et al.*, 2005] Jenny Rose Finkel, Trond Grenager, and Christopher 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, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[Freeman, 2004] Linton Freeman. The development of social network analysis. *A Study in the Sociology of Science*, 2004.

[Fundel *et al.*, 2007] Katrin Fundel, Robert Küffner, and Ralf Zimmer. RelexâĂŤrelation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371, 2007.

[Gamon *et al.*, 2013] Michael Gamon, Tae Yano, Xinying Song, Johnson Apacible, and Patrick Pantel. Understanding document aboutness-step one: Identifying salient entities. Technical report, Technical Report MSR-TR-2013-73, Microsoft Research, 2013.

[García and Tanase, 2013] David García and Dorian Tanase. Measuring cultural dynamics through the eurovision song contest. *Advances in Complex Systems*, 16(08), 2013.

[Garcia *et al.*, 2014] David Garcia, Ingmar Weber, and Venkata Rama Kiran Garimella. Gender asymmetries in reality and fiction: The bechdel test of social media. *International Conference on Weblogs and Social Media (ICWSM)*, 2014.

[Gil *et al.*, 2011] Sebastian Gil, Laney Kuenzel, and Suen Caroline. Extraction and analysis of character interaction networks from plays and movies. Technical report, Stanford University, 2011.

[Gilbert, 2012] Eric Gilbert. Phrases that signal workplace hierarchy. *In Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (CSCW)*, pages 1037–1046, 2012.

[Go *et al.*, 2009] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. Technical report, Stanford, 2009.

[Gooden and Gooden, 2001] Angela M Gooden and Mark A Gooden. Gender representation in notable children's picture books: 1995–1999. *Sex Roles*, 45(1-2):89–101, 2001.

[Gorinski and Lapata, 2015a] Philip John Gorinski and Mirella Lapata. Movie script summarization as graph-based scene extraction. *In the Proceedings of The 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, 2015.

[Gorinski and Lapata, 2015b] Philip John Gorinski and Mirella Lapata. Movie script summarization as graph-based scene extraction. *In Proceedings of HLT-NAACL*, 2015.

[Grishman and Sundheim, 1996] Ralph Grishman and Beth Sundheim. Message understanding conference-6: A brief history. In *COLING*, volume 96, pages 466–471, 1996.

[Groza and Corde, 2015] Adrian Groza and Lidia Corde. Information retrieval in falktales using natural language processing. In *Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on*, pages 59–66. IEEE, 2015.

[GuoDong *et al.*, 2005] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of 43th Annual Meeting of the Association for Computational Linguistics*, 2005.

[Hage and Harary, 1983] P. Hage and F. Harary. *Structural models in anthropology*. Cambridge University Press, 1983.

[Hall *et al.*, 2009] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11, 2009.

[Han and Zhao, 2009] Xianpei Han and Jun Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 215–224. ACM, 2009.

[Harabagiu *et al.*, 2005] Sanda Harabagiu, Cosmin Adrian Bejan, and Paul Morarescu. Shallow semantics for relation extraction. In *International Joint Conference On Artificial Intelligence*, 2005.

[Hasegawa *et al.*, 2004] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 415–422, Barcelona, Spain, July 2004.

[Haussler, 1999] David Haussler. Convolution kernels on discrete structures. Technical report, University of California at Santa Cruz, 1999.

[Hernández and Stolfo, 1995] Mauricio A Hernández and Salvatore J Stolfo. The merge/purge problem for large databases. In *ACM SIGMOD Record*, volume 24, pages 127–138. ACM, 1995.

[Hershkop, 2006] Shlomo Hershkop. *Behavior-based email analysis with application to spam detection.* PhD thesis, Columbia University, 2006.

[Hobson, 1884] John A. Hobson. *The Evolution of Modern Capitalism; A Study of Machine Production.* London, New York: Allen & Unwin, Macmillan., 1884.

[Hoffmann *et al.*, 2010] Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. Learning 5000 relational extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 286–295, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[Hoffmann *et al.*, 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics, 2011.

[Jayannavar *et al.*, 2015] Prashant Arun Jayannavar, Apoorv Agarwal, Melody Ju, and Owen Rambow. Validating literary theories using automatic social network extraction. *on Computational Linguistics for Literature*, page 32, 2015.

[Jelier *et al.*, 2005] Rob Jelier, Guido Jenster, Lambert CJ Dorssers, C Christiaan van der Eijk, Erik M van Mulligen, Barend Mons, and Jan A Kors. Co-occurrence based meta-

analysis of scientific texts: retrieving biological relationships between genes. *Bioinformatics*, 21(9):2049–2058, 2005.

[Jenssen *et al.*, 2001] Tor-Kristian Jenssen, Astrid Lægreid, Jan Komorowski, and Eivind Hovig. A literature network of human genes for high-throughput analysis of gene expression. *Nature genetics*, 28(1):21–28, 2001.

[Jiang and Zhai, 2007] Jing Jiang and ChengXiang 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; Proceedings of the Main Conference*, pages 113–120, Rochester, New York, April 2007. Association for Computational Linguistics.

[Joachims, 1999] Thorsten Joachims. *Advances in Kernel Methods - Support Vector Learning*, chapter Making Large-Scale SVM Learning Practical. MIT Press, 1999.

[Johansson and Nugues, 2007] Richard Johansson and Pierre Nugues. Using wordnet to extend framenet coverage. In *Building Frame Semantics Resources for Scandinavian and Baltic Languages*, pages 27–30. Department of Computer Science, Lund University, 2007.

[Johnson, 1994] J. C. Johnson. Anthropological contributions to the study of social networks: A review. *Advances in social network analysis: Research in the social and behavioral sciences*, 1994.

[Kalashnikov and Mehrotra, 2006] Dmitri V Kalashnikov and Sharad Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems (TODS)*, 31(2):716–767, 2006.

[Kambhatla, 2004] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22. Association for Computational Linguistics, 2004.

[Keila and Skillicorn, 2005] P S Keila and D B Skillicorn. Structure in the enron email dataset. *Computational & Mathematical Organization Theory*, 11 (3):183–199, 2005.

[Klimt and Yang, 2004] Bryan Klimt and Yiming Yang. Introducing the enron corpus. *In proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004.

[Knoke, 1990] D. Knoke. *Political Networks: The structural perspective*. Cambridge University Press, 1990.

[Koehly and Shivy, 1998] Laura M. Koehly and Victoria A. Shivy. Social network analysis: A new methodology for counseling research. *Journal of Counseling Psychology*, 1998.

[Koppel *et al.*, 2002] Moshe Koppel, Shlomo Argamon, and Anat Rachel Shimoni. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412, 2002.

[Krippendorff, 1980] Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, 1980.

[Kulkarni *et al.*, 2009] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466. ACM, 2009.

[Lafferty *et al.*, 2001] J Lafferty, A McCallum, and F Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proc. 18th International Conf. on Machine Learning*, page 282âĂŞ289, 2001.

[Lee *et al.*, 2005] Dongwon Lee, Byung-Won On, Jaewoo Kang, and Sanghyun Park. Effective and scalable solutions for mixed and split citation problems in digital libraries. In *Proceedings of the 2nd international workshop on Information quality in information systems*, pages 69–76. ACM, 2005.

[Lee *et al.*, 2011] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics, 2011.

[Lee *et al.*, 2013] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *MIT Press*, 2013.

[Leskovec and Krevl, 2014] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[Lin and Walker, 2011] Grace I Lin and Marilyn A Walker. All the world's a stage: Learning character models from film. In *AIIDE*, 2011.

[Lodhi *et al.*, 2002] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Christianini, and Chris Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002.

[Mairesse and Walker, 2010] François Mairesse and Marilyn A Walker. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, 20(3):227–278, 2010.

[Mairesse and Walker, 2011] François Mairesse and Marilyn A Walker. Controlling user perceptions of linguistic style: Trainable generation of personality traits. *Computational Linguistics*, 37(3):455–488, 2011.

[Martin *et al.*, 2005] Steve Martin, Blaine Nelson, Anil Sewani, Karl Chen, and Anthony D Joseph. Analyzing behavioral features for email classification. In *CEAS*, 2005.

[McCabe *et al.*, 2011] Janice McCabe, Emily Fairchild, Liz Grauerholz, Bernice A Pescosolido, and Daniel Tope. Gender in twentieth-century childrenâĂŹs books patterns of disparity in titles and central characters. *Gender & Society*, 25(2):197–226, 2011.

[McCallum *et al.*, 2007] Andrew McCallum, Xuerui Wang, and Andres Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30 (1):249–272, 2007.

[McCallum, 2002] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[Michel *et al.*, 2011] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182, 2011.

[Mihalcea and Csomai, 2007] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.

[Miller, 1995] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[Min *et al.*, 2013] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *HLT-NAACL*, pages 777–782, 2013.

[Minkov *et al.*, 2006] Einat Minkov, William W. Cohen, and Andrew Y. Ng. Contextual search and name disambiguation in email using graphs. *In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 27–34, 2006.

[Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. 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: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

[Mohammad and Yang, 2011a] Saif M Mohammad and Tony Wenda Yang. Tracking sentiment in mail: how genders differ on emotional axes. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (ACL-HLT 2011*, pages 70–79, 2011.

[Mohammad and Yang, 2011b] Saif M Mohammad and Tony Wenda Yang. Tracking sentiment in mail: how genders differ on emotional axes. In *Proceedings of the 2nd Workshop*

on Computational Approaches to Subjectivity and Sentiment Analysis (ACL-HLT 2011), pages 70–79, 2011.

[Mooney and Bunescu, 2005] Raymond J Mooney and Razvan C Bunescu. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178, 2005.

[Moretti, 1999] Franco Moretti. *Atlas of the European novel, 1800-1900*. Verso, 1999.

[Moretti, 2005] Franco Moretti. *Graphs, Maps, Trees: Abstract Models for a Literary History*. Verso, 2005.

[Moretti, 2011] Franco Moretti. Network theory, plot analysis. *New Left Review*, 2011.

[Moretti, 2013] Franco Moretti. *Distant reading*. Verso Books, 2013.

[Moschitti, 2004] Alessandro Moschitti. A study on convolution kernels for shallow semantic parsing. In *Proceedings of the 42nd Conference on Association for Computational Linguistic*, 2004.

[Murray and Carenini, 2008] Gabriel Murray and Giuseppe Carenini. Summarizing spoken and written conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 773–782. Association for Computational Linguistics, 2008.

[Newman, 2004] Mark EJ Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004.

[Nguyen and Moschitti, 2011] Truc-Vien T Nguyen and Alessandro Moschitti. End-to-end relation extraction using distant supervision from external semantic repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 277–282. Association for Computational Linguistics, 2011.

[Nguyen *et al.*, 2009] Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. Convolution kernels on constituent, dependency and sequential structures for relation extraction. *Conference on Empirical Methods in Natural Language Processing*, 2009.

[Oard *et al.*, 2015] Douglas W. Oard, William Webber, David A. Kirsch, and Sergey Golitsynskiy. Avocado research email collection ldc2015t03. DVD, 2015.

[On and Lee, 2007] Byung-Won On and Dongwon Lee. Scalable name disambiguation using multi-level graph partition. In *SDM*, pages 575–580. SIAM, 2007.

[Otte and Rousseau, 2002] Evelien Otte and Ronald Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of information Science*, 28(6):441–453, 2002.

[Palus *et al.*, 2013] Sebastian Palus, Piotr Brodka, and Przemysław Kazienko. Evaluation of organization structure based on email interactions. *Governance, Communication, and Innovation in a Knowledge Intensive Society*, page 117, 2013.

[Paşca *et al.*, 2006] Marius Paşca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. Names and similarities on the web: fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 809–816. Association for Computational Linguistics, 2006.

[Pedersen *et al.*, 2005] Ted Pedersen, Amruta Purandare, and Anagha Kulkarni. Name discrimination by clustering similar contexts. In *Computational Linguistics and Intelligent Text Processing*, pages 226–237. Springer, 2005.

[Peersman *et al.*, 2011] Claudia Peersman, Walter Daelemans, and Leona Van Vaerenbergh. Predicting age and gender in online social networks. In *Proceedings of the 3rd international workshop on Search and mining user-generated contents*, pages 37–44. ACM, 2011.

[Peterson *et al.*, 2011] Kelly Peterson, Matt Hohensee, and Fei Xia. Email formality in the workplace: A case study on the enron corpus. In *Proceedings of the Workshop on Languages in Social Media*, pages 86–95. Association for Computational Linguistics, 2011.

[Plank and Moschitti, 2013] Barbara Plank and Alessandro Moschitti. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *ACL (1)*, pages 1498–1507, 2013.

[Prabhakaran and Rambow, 2014] Vinodkumar Prabhakaran and Owen Rambow. Predicting power relations between participants in written dialog from a single thread. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 339–344, 2014.

[Prabhakaran *et al.*, 2014] Vinodkumar Prabhakaran, Emily E Reid, and Owen Rambow. Gender and power: How gender and gender environment affect manifestations of power. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, October. Association for Computational Linguistics*, 2014.

[Ratinov *et al.*, 2010] Lev Ratinov, Doug Downey, and Dan Roth. Wikification for information retrieval. Technical report, University of Illinois, 2010.

[Ratinov *et al.*, 2011] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics, 2011.

[Riedel *et al.*, 2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.

[Riedel *et al.*, 2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL-HLT*, pages 74–84, 2013.

[Riloff *et al.*, 1999] Ellen Riloff, Rosie Jones, et al. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479, 1999.

[Rosenthal, 2015] Sara Rosenthal. *Detecting Influence in Social Media Discussions*. PhD thesis, Columbia University, 2015.

[Rowe *et al.*, 2007] Ryan Rowe, German Creamer, Shlomo Hershkop, and Salvatore J Stolfo. Automated social hierarchy detection through email network analysis. *Proceedings of*

the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, pages 109–117, 2007.

[Sanjek, 1974] R. Sanjek. What is social network analysis, and what it is good for? *Reviews in Anthropology*, 1974.

[Santorini, 1990] Beatrice Santorini. Part-of-speech tagging guidelines for the penn treebank project (3rd revision). Technical report, University of Pennsylvania, 1990.

[Scheiner-Fisher and Russell III, 2012] Cicely Scheiner-Fisher and William B Russell III. Using historical films to promote gender equity in the history curriculum. *The Social Studies*, 103(6):221–225, 2012.

[Schwartz *et al.*, 2013] H. Andrew Schwartz, Johannes C. Eichstaedt, Margaret L. Kern, Lukasz Dziurzynski, Stephanie M. Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin E. P. Seligman, and Lyle H. Ungar. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PLoS ONE*, 2013.

[Seidman, 1985] S. B. Seidman. Structural consequences of individual position in nondyadic social networks. *Journal of Mathematical Psychology*, 1985.

[Sekine, 2006] Satoshi Sekine. On-demand information extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 731–738. Association for Computational Linguistics, 2006.

[Shetty and Adibi, 2004] J. Shetty and J. Adibi. Ex employee status report. Technical report, 2004.

[Shetty and Adibi, 2005] Jitesh Shetty and Jafar Adibi. Discovering important nodes through graph entropy the case of enron email database. In *Proceedings of the 3rd international workshop on Link discovery*, pages 74–81. ACM, 2005.

[Shi and Mihalcea, 2005] Lei Shi and Rada Mihalcea. Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In *Computational linguistics and intelligent text processing*, pages 100–111. Springer, 2005.

[Shinyama and Sekine, 2006] Yusuke Shinyama and Satoshi Sekine. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 304–311. Association for Computational Linguistics, 2006.

[Smith *et al.*, 2013] S.L. Smith, M. Choueiti, E. Scofield, and K. Pieper. Gender inequality in 500 popular films: Examining onâĂŘscreen portrayals and behindâĂŘtheâĂŘscenes employment patterns in motion pictures released between 2007 and 2012. *Media, Diversity, and Social Change Initiative: Annenberg School for Communication and Journalism, USC*, 2013.

[Sparrow, 1991] Malcolm K. Sparrow. The application of network analysis to criminal intelligence: An assessment of the prospects. *Social networks*, 1991.

[Srivastava *et al.*, 2015a] Shashank Srivastava, Snigdha Chaturvedi, and Tom Mitchell. Inferring interpersonal relations in narrative summaries. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, Arizona*, 2015.

[Srivastava *et al.*, 2015b] Shashank Srivastava, Snigdha Chaturvedi, and Tom Mitchell. Inferring interpersonal relations in narrative summaries. *arXiv preprint arXiv:1512.00112*, 2015.

[Suchanek *et al.*, 2007] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

[Sugimoto *et al.*, 2013] Cassidy R Sugimoto, Vincent Lariviere, CQ Ni, Yves Gingras, and Blaise Cronin. Global gender disparities in science. *Nature*, 504(7479):211–213, 2013.

[Sun *et al.*, 2011] Ang Sun, Ralph Grishman, and Satoshi 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-Volume 1*, pages 521–529. Association for Computational Linguistics, 2011.

[Surdeanu *et al.*, 2011] Mihai Surdeanu, Sonal Gupta, John Bauer, David McClosky, Angel X Chang, Valentin I Spitkovsky, and Christopher D Manning. StanfordâĂŹs distantly-supervised slot-filling system. In *Proceedings of the Text Analytics Conference*, 2011.

[Tichy *et al.*, 1979] Noel M. Tichy, Michael L. Tushman, and Charles Fombrun. Social network analysis for organizations. *Academy of Management Review*, pages 507–519, 1979.

[Toutanova *et al.*, 2003] Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. *In Proceedings of HLT-NAACL*, 2003.

[Turetsky and Dimitrova, 2004] Robert Turetsky and Nevenka Dimitrova. Screenplay alignment for closed-system speaker identification and analysis of feature films. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, pages 1659–1662. IEEE, 2004.

[Wagner *et al.*, 2015] Claudia Wagner, David Garcia, Mohsen Jadidi, and Markus Strohmaier. It's a man's wikipedia? assessing gender inequality in an online encyclopedia. Arxiv preprintarXiv:1501.06307, 2015.

[Walker *et al.*, 2011] Marilyn A Walker, Ricky Grant, Jennifer Sawyer, Grace I Lin, Noah Wardrip-Fruin, and Michael Buell. Perceived or not perceived: Film character models for expressive nlg. In *Interactive Storytelling*, pages 109–121. Springer, 2011.

[Walker *et al.*, 2012] Marilyn A Walker, Grace I Lin, and Jennifer Sawyer. An annotated corpus of film dialogue for learning and characterizing character style. In *LREC*, pages 1373–1378, 2012.

[Wang *et al.*, 2011] Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. Relation extraction with relation topics. *Empirical Methods in Natural Language Processing*, 2011.

[Wang *et al.*, 2013] Yi Wang, Marios Iliofotou, Michalis Faloutsos, and Bin Wu. Analyzing communication interaction networks (cins) in enterprises and inferring hierarchies. *Computer Networks*, 57(10):2147–2158, 2013.

[Wasserman and Faust, 1994] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. New York: Cambridge University Press, 1994.

[Weitzman *et al.*, 1972] Lenore J Weitzman, Deborah Eifler, Elizabeth Hokada, and Catherine Ross. Sex-role socialization in picture books for preschool children. *American journal of Sociology*, pages 1125–1150, 1972.

[Wellman, 1926] Beth Wellman. The school childâĂŹs choice of companions. *Journal of Educational Research*, 14:21–423, 1926.

[Weng *et al.*, 2009] Chung-Yi Weng, Wei-Ta Chu, and Ja-Ling Wu. Rolenet: Movie analysis from the perspective of social networks. *Multimedia, IEEE Transactions on*, 11(2):256–271, 2009.

[Williams, 1975] Raymond Williams. *The country and the city*. Oxford University Press, 1975.

[Winkler, 1999] William E Winkler. The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau*. Citeseer, 1999.

[Wu and Weld, 2007] Fei Wu and Daniel S Weld. Autonomously semantifying wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50. ACM, 2007.

[Wu and Weld, 2010] Fei Wu and Daniel S. Weld. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 118–127, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[Yang, 1999] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information retrieval*, 1(1-2):69–90, 1999.

[Ye and Baldwin, 2008] Patrick Ye and Timothy Baldwin. Towards automatic animated storyboarding. In *AAAI*, pages 578–583, 2008.

[Yeh and Harnly, 2006] Jen-Yuan Yeh and Aaron Harnly. Email thread reassembly using similarity matching. *In Proceedings of the Third Conference on Email and Anti-Spam (CEAS)*, 2006.

[Yin *et al.*, 2007] Xiaoxin Yin, Jiawei Han, and Philip S Yu. Object distinction: Distinguishing objects with identical names. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 1242–1246. IEEE, 2007.

[Zajic *et al.*, 2008] David M Zajic, Bonnie J Dorr, and Jimmy Lin. Single-document and multi-document summarization techniques for email threads using sentence compression. *Information Processing & Management*, 44(4):1600–1610, 2008.

[Zelenko *et al.*, 2002] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 71–78. Association for Computational Linguistics, July 2002.

[Zhang *et al.*, 2006] Min Zhang, Jie Zhang, Jian Su, and Guodong Zhou. A composite kernel to extract relations between entities with both flat and structured features. In *Proceedings of COLING-ACL*, 2006.

[Zhang *et al.*, 2008] Min Zhang, GuoDong Zhou, and Aiti Aw. Exploring syntactic structured features over parse trees for relation extraction using kernel methods. *Information processing & management*, 44(2):687–701, 2008.

[Zhao and Grishman, 2005] Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd Meeting of the ACL*, 2005.

[Zhou *et al.*, 2010] Guodong Zhou, Longhua Qian, and Jianxi Fan. Tree kernel-based semantic relation extraction with rich syntactic and semantic information. *Information Sciences*, 180(8):1313–1325, 2010.

[Žižek, 1989] Slavoj Žižek. *The sublime object of ideology.* Verso, 1989.

Part VII

Appendices

Appendix A

Support Vector Machines and Convolution Kernels

A.1 Support Vector Machines

Support Vector Machines (SVMs) are supervised, binary, maximum margin classifiers introduced by Cortes and Vapnik [1995]. Given a training data-set, $\{(\vec{x_i}, y_i) \mid i = 1, 2, \ldots, l, y_i \in \{1, -1\}\}$, SVMs learn a maximum margin hyperplane that separates the data into two classes. The hyperplane is given by $f(\vec{x}) = \vec{w} \cdot \vec{x} + b$, where $\vec{w}$ is the normal to the hyperplane and b is its intercept at $\vec{x} = 0$. The gradient, $\vec{w} = \Sigma_{i=1}^{l} \mu_i y_i \vec{x_i}$, where $\mu_i \geq 0$ is the Lagrange multiplier associated with each point $\vec{x_i}$. Points with $\mu_i > 0$ are called support vectors. As the above formula shows, the model learned by an SVM is a linear combination of training data points ($\vec{w}$), and a *bias* term (b). Given a test point, SVM use the following rule to classify the point into one of two categories:

$$y = \begin{cases} 1 & \text{if } sign(f(\vec{x})) \geq 0 \\ -1 & \text{if } sign(f(\vec{x})) < 0 \end{cases}$$

So far, we have described the way SVMs learn a maximum margin hyperplane separating the two classes of data points. One important question is, how do we represent our data i.e. the set $\{\vec{x}_i\}_{i=1}^{l}$. A popular methodology of converting unstructured data into structured representation is by using feature extraction. Using feature extraction, unstructured data

may be mapped into a finite dimensional space. In this case $\vec{x} = (x_1, x_2, \ldots, x_d)$, where x_j is the j^{th} component of vector $\vec{x}$, and $d \in \mathbb{N}$ is the dimensionality of the space (or the number of features). Data may also be represented as abstract structures such as strings, trees, etc. For the latter representation, it becomes crucial that SVMs be used in their *dual* form. In the dual form, the optimization function that SVMs solve is as follows [Burges, 1998]:

$$max \;\; \Sigma_i \mu_i - \Sigma_{i,j} \mu_i \mu_j y_i y_j K(\vec{x_i}, \vec{x_j})$$

$$\text{s.t.} \quad \Sigma_i \mu_i y_i = 0$$

$$\mu_i \geq 0 \quad \forall i = 1, 2, \ldots, l$$

Here, K is called the kernel function that assigns every pair of objects a real number. This function is required to satisfy Mercer's condition. More formally, $K : X \times X \rightarrow \mathbb{R}$, where X is the set of objects. For all square integrable functions $g(\vec{x})$, $\int K(\vec{x}, \vec{y}) g(\vec{x}) g(\vec{y}) d\vec{x} d\vec{y} \geq 0$. For example, if we represent our input examples as feature vectors, the set X would be the set of feature vectors. For feature vectors, if we use a linear kernel, then $K(\vec{x_i}, \vec{x_j}) = \vec{x_i} \cdot \vec{x_j}$ (dot product of the two vectors). But if X is the set of abstract objects, such as, trees, then K must be a convolution kernel, first introduced by Haussler [1999].

As is clear from the definition of a kernel, to use an SVM in its dual form, the machine needs to go through the data twice i.e. the complexity of the SVM in its dual form is quadratic in the number of examples. With growing data-set sizes, this quadratic complexity can become unacceptable. Neural networks, which are online in nature, are useful to overcome this limitation.

Appendix B

Frame Semantic Rules

B.1 Semantic Rules

S.No.	Frame	First frame element	Second frame element	Type of social event
1	Abandonment	Agent	Theme	OBS
2	Abusing	Abuser	Victim	INR
3	Activity_prepare	Agent	Beneficiary	OBS
4	Activity_ready_state	Protagonist	Salient_entity	OBS
5	Activity_start	Agent	Purpose	OBS
6	Activity_start	Agent	Activity	OBS
7	Activity_stop	Agent	Purpose	OBS
8	Activity_stop	Agent	Activity	OBS
9	Adducing	Speaker	Specified_entity	OBS
10	Aggregate	Individuals	Individuals	OBS
11	Agree_or_refuse_to_act	Speaker	Interlocutor	INR
12	Agree_or_refuse_to_act	Speaker	Proposed _action	OBS
13	Agree_or_refuse_to_act	Interlocutor	Proposed _action	OBS

14	Alliance	Members	Members	INR
15	Alliance	Member_1	Member_2	INR
16	Amalgamation	Parts	Parts	INR
17	Amalgamation	Part_1	Part_2	INR
18	Appeal	Convict	Representative	INR
19	Appointing	Selector	Official	INR
20	Arrest	Authorities	Suspect	INR
21	Arriving	Theme	Cotheme	INR
22	Assessing	Assessor	Phenomenon	OBS
23	Assistance	Helper	Benefited_party	INR
24	Attaching	Agent	Item	INR
25	Attack	Assailant	Victim	INR
26	Attempt_suasion	Speaker	Addresse	INR
27	Attempt_suasion	Speaker	Content	OBS
28	Attempt_suasion	Addressee	Content	OBS
29	Attention	Perceiver	Figure	OBS
30	Attributed_information	Speaker	Proposition	OBS
31	Avoiding	Agent	Undesirable _situation	OBS
32	Awareness	Cognizer	Content	OBS
33	Bail_decision	Judge	Accused	INR
34	Be_in_control	Controlling _entity	Dependent _entity	OBS
35	Beat_opponent	Winner	Loser	INR
36	Becoming_a_member	New_member	Group	INR
37	Becoming_aware	Cognizer	Phenomenon	OBS
38	Being_at_risk	Asset	Dangerous _entity	OBS
39	Being_employed	Employer	Employee	INR
40	Biological_urge	Experiencer	Reason	OBS

41	Birth_scenario	Mother	Offspring	OBS
42	Birth_scenario	Mother	Father	INR
43	Board_vehicle	Traveller	Cotheme	OBS
44	Body_movement	Agent	Addresse	OBS
45	Candidness	Speaker	Addresse	INR
46	Candidness	Speaker	Message	OBS
47	Candidness	Addressee	Message	OBS
48	Categorization	Cognizer	Item	OBS
49	Cause_change	Entity	Agent	OBS
50	Cause_change_of_strength	Agent	Patient	INR
51	Cause_emotion	Experiencer	Agent	OBS
52	Cause_harm	Agent	Victim	INR
53	Cause_motion	Agent	Theme	INR
54	Cause_to_amalgamate	Agent	Parts	OBS
55	Change_of_leadership	Selector	New_leader	INR
56	Change_of_leadership	Selector	Old_leader	INR
57	Chatting	Interlocutors	Interlocutors	INR
58	Chatting	Interlocutor_1	Interlocutor_2	INR
59	Clemency	Executive _authority	Offender	INR
60	Collaboration	Partner_1	Partner_2	INR
61	Collaboration	Partners	Partners	INR
62	Come_together	Individuals	Individuals	INR
63	Come_together	Party_1	Party_2	INR
64	Commerce_buy	Buyer	Seller	INR
65	Commerce_pay	Buyer	Seller	INR
66	Commerce_scenario	Buyer	Seller	INR
67	Commerce_sell	Buyer	Seller	INR
68	Commitment	Speaker	Addresse	INR
69	Commitment	Speaker	Message	OBS

70	Commitment	Addressee	Message	OBS
71	Communication	Speaker	Addresse	INR
72	Communication	Speaker	Message	OBS
73	Communication	Addressee	Message	OBS
74	Communication_noise	Speaker	Addresse	INR
75	Communication_noise	Speaker	Message	OBS
76	Communication_noise	Addressee	Message	OBS
77	Communication_response	Speaker	Addresse	INR
78	Communication_response	Speaker	Message	OBS
79	Communication_response	Addressee	Message	OBS
80	Competition	Participant_1	Participant_2	INR
81	Competition	Participants	Participants	INR
82	Conquering	Theme	Conqueror	INR
83	Contacting	Communicator	Addressee	INR
84	Convey_importance	Speaker	Addresse	INR
85	Convey_importance	Speaker	Message	OBS
86	Convey_importance	Addressee	Message	OBS
87	Cotheme	Theme	Cotheme	INR
88	Court_examination	Questioner	Witness	INR
89	Create_representation	Creator	Represented	OBS
90	Cure	Healer	Patient	INR
91	Defending	Defender	Assailant	INR
92	Defending	Defender	Victim	OBS
93	Defending	Assailant	Victim	INR
94	Delivery	Deliverer	Recipient	INR
95	Deny_permission	Authority	Protagonist	INR
96	Discussion	Interlocutors	Interlocutors	INR
97	Discussion	Interlocutor_1	Interlocutor_2	INR
98	Education_teaching	Teacher	Student	INR
99	Employing	Employer	Employee	INR

100	Exchange	Exchangers	Exchangers	INR
101	Exchange	Exchanger_1	Exchanger_1	INR
102	Exchange	Exchanger_2	Exchanger_2	INR
103	Excreting	Excreter	Goal	INR
104	Execution	Executioner	Executed	INR
105	Expressing_publicly	Communicator	Addressee	INR
106	Expressing_publicly	Communicator	Content	OBS
107	Expressing_publicly	Addressee	Content	OBS
108	Fairness_evaluation	Actor	Affected_party	OBS
109	Forgiveness	Judge	Evaluee	INR
110	Forgiveness	Judge	Offense	OBS
111	Forgiveness	Evaluee	Offense	OBS
112	Forming_relationships	Partners	Partners	INR
113	Forming_relationships	Partner_1	Partner_2	INR
114	Gathering_up	Agent	Individuals	INR
115	Getting	Source	Recipient	INR
116	Giving	Donor	Recipient	INR
117	Grant_permission	Grantor	Grantee	INR
118	Grant_permission	Grantor	Action	OBS
119	Grant_permission	Grantee	Action	OBS
120	Hear	Speaker	Hearer	INR
121	Hear	Speaker	Message	OBS
122	Hear	Hearer	Message	OBS
123	Heralding	Communicator	Individual	INR
124	Hostile_encounter	Sides	Sides	INR
125	Hostile_encounter	Side_1	Side_2	INR
126	Intentionally_affect	Agent	Patient	OBS
127	Judgment	Cognizer	Evaluee	OBS
128	Judgment	Cognizer	Reason	OBS
129	Judgment_communication	Communicator	Addressee	INR

130	Judgment_communication	Communicator	Evaluee	OBS
131	Judgment_communication	Addressee	Evaluee	OBS
132	Judgment_communication	Communicator	Reason	OBS
133	Judgment_communication	Addressee	Reason	OBS
134	Judgment_direct_address	Communicator	Addressee	INR
135	Judgment_direct_address	Communicator	Reason	OBS
136	Judgment_direct_address	Addressee	Reason	OBS
137	Justifying	Agent	Judge	INR
138	Justifying	Agent	Act	OBS
139	Justifying	Judge	Act	OBS
140	Kidnapping	Perpetrator	Victim	INR
141	Killing	Killer	Victim	INR
142	Kinship	Ego	Alter	INR
143	Labeling	Speaker	Entity	OBS
144	Make_agreement_on_action	Parties	Parties	INR
145	Make_agreement_on_action	Party_1	Party_2	INR
146	Meet_with	Party_1	Party_2	INR
147	Meet_with	Party_1	Party_1	INR
148	Morality_evaluation	Judge	Evaluee	INR
149	Notification_of_charges	Arraign _authority	Accused	INR
150	Offering	Offerer	Potential _recipient	INR
151	Pardon	Authority	Offender	INR
152	Participation	Participant_1	Participant_2	INR
153	Participation	Participant_1	Participant_1	INR
154	Participation	Participants	Participants	INR
155	Perception_active	Perceiver _agentive	Phenomenon	OBS

156	Perception_experience	Perceiver _passive	Phenomenon	OBS
157	Personal_relationship	Partner_1	Partner_2	INR
158	Personal_relationship	Partner_1	Partner_1	INR
159	Personal_relationship	Partners	Partners	INR
160	Prevarication	Speaker	Addresse	INR
161	Prevarication	Speaker	Topic	OBS
162	Prevarication	Addressee	Topic	OBS
163	Quarreling	Arguer_1	Arguer_2	INR
164	Quarreling	Arguers	Arguers	INR
165	Questioning	Speaker	Addresse	INR
166	Questioning	Speaker	Message	OBS
167	Questioning	Addressee	Message	OBS
168	Quitting	Employee	Employer	INR
169	Reasoning	Arguer	Addresse	INR
170	Reasoning	Arguer	Content	OBS
171	Reasoning	Arguer	Support	OBS
172	Receiving	Donor	Recipient	INR
173	Request	Speaker	Adressee	INR
174	Request	Speaker	Message	OBS
175	Respond_to_proposal	Speaker	Interlocutor	INR
176	Respond_to_proposal	Speaker	Proposal	OBS
177	Respond_to_proposal	Interlocutor	Proposal	OBS
178	Reveal_secret	Speaker	Addresse	INR
179	Reveal_secret	Speaker	Information	OBS
180	Reveal_secret	Addressee	Information	OBS
181	Revenge	Avenger	Offender	INR
182	Revenge	Injured_Party	Offender	INR
183	Rewards_and_punishments	Agent	Evaluee	INR
184	Rewards_and_punishments	Agent	Reason	OBS

185	Rewards_and_punishments	Evaluee	Reason	OBS
186	Sending	Sender	Recipient	INR
187	Sentencing	Court	Convict	INR
188	Sentencing	Convict	Offense	INR
189	Sentencing	Court	Offense	OBS
190	Silencing	Agent	Speaker	INR
191	Sociability	Judge	Protagonist	OBS
192	Sociability	Judge	Company	OBS
193	Sociability	Protagonist	Company	INR
194	Social_connection	Individual_1	Individual_2	INR
195	Social_connection	Individuals	Individuals	INR
196	Social_event	Attendee	Attendee	INR
197	Social_event	Attendee	Honoree	OBS
198	Social_event_collective	Attendees	Attendees	INR
199	Social_interaction_evaluation	Judge	Evaluee	INR
200	Social_interaction_evaluation	Judge	Affected_party	INR
201	Social_interaction_evaluation	Evaluee	Affected_party	INR
202	Speak_on_topic	Speaker	Audience	INR
203	Speak_on_topic	Speaker	Topic	OBS
204	Speak_on_topic	Audience	Topic	OBS
205	Statement	Speaker	Message	OBS
206	Statement	Speaker	Topic	OBS
207	Statement	Speaker	Medium	OBS
208	Suasion	Speaker	Addressee	INR
209	Suasion	Speaker	Topic	OBS
210	Suasion	Speaker	Content	OBS
211	Suasion	Addressee	Content	OBS
212	Suasion	Speaker	Text	OBS
213	Submitting_documents	Submittor	Authority	INR
214	Subordinates_and_superiors	Subordinate	Superior	INR

215	Subversion	Counter_actor	Agent	INR
216	Supply	Supplier	Recipient	INR
217	Surrendering	Fugitive	Authorities	INR
218	Telling	Speaker	Addresse	INR
219	Telling	Speaker	Message	OBS
220	Telling	Addressee	Message	OBS
221	Terrorism	Terrorist	Victim	OBS
222	Transfer	Donor	Recipient	INR
223	Trial	Defendant	Judge	INR
224	Trial	Defendant	Charges	OBS
225	Trial	Defense	Defendant	INR
226	Trial	Judge	Defense	INR
227	Trial	Prosecution	Judge	INR
228	Trial	Prosecution	Defense	INR
229	Trial	Prosecution	Defendant	INR
230	Trial	Prosecution	Charges	OBS
231	Trial	Jury	Judge	OBS
232	Trial	Jury	Prosecution	OBS
233	Trial	Jury	Charges	OBS
234	Trial	Jury	Defendant	OBS
235	Trial	Jury	Defense	OBS
236	Verdict	Defendant	Charges	OBS
237	Verdict	Judge	Finding	OBS
238	Verdict	Defendant	Finding	OBS
239	Verdict	Judge	Defendant	INR
240	Volubility	Judge	Speaker	OBS

Appendix C

List of Features for Bechdel Test

C.1 Frame Features and their Counts for the Bechdel Test

Locative _relation 4934, Observable _body _parts 3520, Intentionally _act 3103, Calendric _unit 2914, Arriving 2737, Quantity 2274, Cardinal _numbers 2271, Building _subparts 2061, Temporal _collocation 1808, People 1759, Buildings 1747, Scrutiny 1594, Leadership 1473, Vehicle 1437, Placing 1412, Statement 1363, Self _motion 1332, Motion 1236, Causation 1205, Roadways 1157, Capability 1156, Increment 1145, Connecting _architecture 1084, Becoming 1055, Being _obligated 1045, Removing 1025, Perception _active 1010, Ingestion 1002, Relational _quantity 940, Change _position _on _a _scale 925, Cause _motion 862, Emotion _directed 830, Desirability 830, Dimension 813, Perception _experience 786, Clothing 757, Weapon 754, Awareness 746, Grasp 734, Body _movement 719, Aggregate 703, Part _orientational 697, Measure _duration 683, Natural _features 675, Food 675, Cause _harm 674, Desiring 673, Experiencer _focus 649, Certainty 639, Kinship 636, Architectural _part 629, Containers 599, Contacting 590, Frequency 588, Locale _by _use 587, Age 574, Existence 569, Manipulation 556, Giving 548, Relative _time 547, Make _noise 546, Being _located 522, People _by _vocation 498, Stimulus _focus 497, Taking _sides 485, Possession 480, Opinion 472, Personal _relationship 460, Likelihood 458, Request 449, Political _locales 441, Locale 436, Morality _evaluation 420, Killing 417, Contingency 416, Attempt 415, Time _vector 409, Experiencer _obj 409, Sufficiency 404, Location _of _light 403, Correctness 403, Theft 402, Making _faces 402, Process _start 396, Substance 394, People

_by _age 394, Posture 383, Measure _linear _extent 383, Impact 380, Size 369, Sounds 354, Continued _state _of _affairs 351, Identicality 339, Accoutrements 337, Required _event 330, Ordinal _numbers 327, Reason 326, Grant _permission 322, Text 318, Secrecy _status 317, Departing 312, Containing 306, Degree 304, Color 300, Taking _time 298, Means 294, Type 291, Gesture 286, Part _inner _outer 284, Temporal _subregion 275, Wearing 270, Clothing _parts 270, Process _continue 268, Choosing 267, Activity _start 266, Have _as _requirement 261, Physical _artworks 257, Activity _ongoing 254, Beat _opponent 250, Sole _instance 243, Bringing 239, Becoming _aware 231, Being _named 226, Using 223, Sensation 219, Money 217, Importance 212, Speed 211, Evidence 209, Undergo _change 202, Abounding _with 201, Judgment _communication 198, Gizmo 194, Businesses 194, Difficulty 193, Education _teaching 192, Cause _change _of _position _on _a _scale 190, Reasoning 189, Filling 185, Assistance 185, Fluidic _motion 184, Activity _stop 183, Telling 181, Candidness 180, Attaching 178, Inclusion 176, Front _for 176, Expertise 176, Measure _volume 174, Reading 173, Shapes 172, Locating 170, Hostile _encounter 170, Assessing 168, Working _on 166, Social _connection 166, Performers _and _roles 166, Coming _to _be 165, Similarity 163, Waking _up 160, Duration _attribute 160, Waiting 159, Breathing 158, Event 156, Dead _or _alive 156, Purpose 154, Inspecting 154, Mental _property 152, Part _whole 150, Part _piece 150, Behind _the _scenes 150, Residence 149, Cotheme 148, Shoot _projectiles 146, Medical _professionals 143, Topi142, Conquering 142, Preventing 141, Feeling 140, Intoxicants 138, Communication _response 137, Judgment _direct _address 131, Grinding 130, Death 128, Origin 126, Collaboration 126, Connectors 123, Biological _urge 123, Being _up _to _it 122, Position _on _a _scale 121, Change _posture 120, Traversing 119, Sending 119, Predicament 119, Appearance 119, Ingest _substance 117, Getting 117, Expressing _publicly 117, Success _or _failure 113, Precipitation 113, Cause _to _fragment 113, Prison 112, Communication _noise 112, Come _together 111, Cause _change 111, Temporal _pattern 110, Social _event 108, Performing _arts 108, Hunting _success _or _failure 108, Commerce _pay 108, Usefulness 107, Sign _agreement 107, Intentionally _create 107, Ammunition 107, Compliance 106, Hair _configuration 105, Discussion 104, Adducing 104, Cause _to _make _noise 103, Military 101, Communication _manner 100, Reveal _secret 98, Finish _competition 98, Coming _to _believe 98,

Contrition 97, Categorization 96, Being _employed 96, Excreting 95, Defend 95, Fullness 94, Arrest 91, Instance 90, Eclipse 90, Simple _name 89, Obviousness 89, Being _at _risk 89, Range 88, Direction 88, Memory 86, First _rank 85, Aesthetics 84, Apply _heat 83, Organization 82, Operate _vehicle 82, Activity _ready _state 82, Active _substance 81, People _along _political _spectrum 80, Commitment 80, Participation 79, Biological _area 79, Withdraw _from _participation 78, Ranked _expectation 76, Process _end 76, Expensiveness 76, Emptying 76, Commerce _buy 75, Building 74, Resolve _problem 73, Ambient _temperature 73, Noise _makers 72, Moving _in _place 72, Manufacturing 72, Text _creation 71, Halt 71, Chatting 71, Being _attached 71, Medical _conditions 70, Storing 69, Judgment 69, Response 67, Attack 67, Communicate _categorization 66, Attention _getting 65, Volubility 64, Body _description _holistic 64, People _by _origin 63, Expansion 63, Sleep 62, Path _shape 60, Control 60, Avoiding 60, Membership 59, Law 59, Judicial _body 59, Cogitation 59, Estimating 58, Questioning 57, Quarreling 57, Protecting 57, Terrorism 56, Measurable _attributes 56, Inhibit _movement 56, Supply 55, Earnings _and _losses 55, Documents 55, Setting _fire 54, Encoding 54, Attempt _suasion 54, Risky _situation 53, Undressing 51, Sound _level 51, Field51, Recording 50, Becoming _a _member 50, Adorning 49, Partitive 48, Closure 48, Locale _by _event 47, Confronting _problem 47, Travel 46, Timespan 46, Make _acquaintance 46, Remembering _information 45, Change _event _time 45, Being _wet 45, Artifact 45, Social _interaction _evaluation44, Offering 44, Duplication 44, Verdict 43, Trust 43, Activity _pause 43, State _continue 42, Process _completed _state 42, Point _of _dispute 42, Mass _motion 42, Emotion _active 42, Thwarting 41, Remembering _experience 41, Evaluative _comparison 41, Employing 41, Commerce _sell41, Attention 40, Lively _place 39, Sequence 38, Dressing 38, Practice 37, Operational _testing 37, Dispersal 37, Rite 36, Reshaping 36, Objective _influence 36, Motion _noise 35, Motion _directional 35, Legality 35, Verification 34, Operating _a _system 34, Occupy _rank 34, Expectation 34, Electricity 34, Destroying 34, Quitting _a _place 33, Indigenous _origin 32, Forming _relationships 32, Concessive 32, Change _direction 32, Bungling 32, Rank 31, Public _services 31, Conduct 31, Activity _finish 31, Used _up 30, Undergoing 30, Individual _history 30, Hit _target 30, Exertive _force 30, Deciding 30, Commerce _scenario 29, Measure _mass 28, Facial _expression 28, Exchange 28, Diversity

28, Create _physical _artwork 28, Cause _expansion 28, Board _vehicle 28, Separating 27, Seeking 27, Make _agreement _on _action 27, Experience _bodily _harm 27, Cause _to _wake 27, Appointing 27, Adopt _selection 27, Temperature 26, Quitting 26, Make _cognitive _connection 26, Information 26, Suitability 25, Receiving 25, Project 25, Presence 25, Event _instance 25, Cure 25, Creating 25, Become _silent 25, Trial 24, Seeking _to _achieve 24, Releasing 24, Place _weight _on 24, Part _ordered _segments 24, Forgiveness 24, Coming _up _with 24, Cause _to _make _progress 24, Sign 23, Gathering _up 23, Evading 23, Craft 23, Completeness 23, Committing _crime 23, Colonization 23, Suspiciousness 22, Fame 22, Cause _to _end 22, Body _decoration 22, Activity _prepare 22, Sharpness 21, Ride _vehicle 21, Revenge 21, Hear 21, Firing 21, Emotion _heat 21, Cause _fluidic _motion 21, Weather 20, Taking 20, Remainder 20, Offenses 20, Deserving 20, Court _examination 20, Communication 20, Cognitive _connection 20, Cause _to _amalgamate 20, Catastrophe 20, Aiming 20, Abandonment 20, Linguistic _meaning 19, Interrupt _process 19, Hiding _objects 19, Delivery 19, Change _tool 19, Wealthiness 18, Submitting _documents 18, Store 18, Research 18, Progress 18, Change _of _phase 18, Bearing _arms 18, Altered _phase 18, Adjusting 18, Temporary _stay 17, Create _representation 17, Compatibility 17, Cause _temperature _change 17, Artificiality 17, System 16, Sound _movement 16, Prevarication 16, Measure _by _action 16, Just _found _out 16, Ineffability 16, Chemical-sense _description 16, Being _necessary 16, Arranging 16, Addiction 16, Luck 15, Intentionally _affect 15, Imitating 15, Hiring 15, Feigning 15, Damaging 15, Make _possible _to _do 14, Agree _or _refuse _to _act 14, Speak _on _topic 13, Relational _natural _features 13, Labeling 13, Idiosyncrasy 13, History 13, Emitting 13, Change _of _quantity _of _possession 13, Cause _to _start 13, Cause _to _be _dry 13, Being _operational 13, Becoming _separated 13, Take _place _of 12, Network 12, Being _detached 12, Accompaniment 12, Scouring 11, Rejuvenation 11, Possibilities11, Hunting 11, Hit _or _miss 11, Distinctiveness 11, Differentiation 11, Cutting 11, Coincidence 11, Version _sequence 10, Sent _items 10, Reliance 10, Misdeed 10, Medical _instruments 10, Fleeing 10, Entity 10, Custom 10, Typicality 9, Transfer 9, State _of _entity 9, Respond _to _proposal 9, Regard 9, Punctual _perception 9, People _by _jurisdiction 9, Patter9, Partiality 9, Kidnapping 9, Health _response 9, Foreign _or _domestic _country 9, Execu-

tion 9, Evoking 9, Dominate _situation 9, Complaining 9, Competition 9, Activity _resume 9, Replacing 8, Processing _materials 8, People _by _residence 8, Manipulate _into _doing 8, Ingredients 8, Getting _underway 8, Emotions _by _stimulus 8, Change _operational _state 8, Cause _emotion 8, Vocalizations 7, Summarizing 7, Shopping 7, Scope 7, Run _risk 7, Robbery 7, Prohibiting 7, Omen 7, Medical _specialties 7, Having _or _lacking _access 7, Destiny 7, Change _of _leadership 7, Body _mark 7, Body _description _part 7, Being _born 7, Be _in _agreement _on _action 7, Accomplishment 7, Surviving 6, Subordinates _and _superiors 6, Silencing 6, Rotting 6, Reporting 6, Render _nonfunctional 6, Relational _political _locales 6, Recovery 6, Isolated _places 6, Grooming 6, Change _resistance 6, Amounting _to 6, Willingness 5, Visiting 5, Unattributed _information 5, Sociability 5, Sentencing 5, Rewards _and _punishments 5, Preserving 5, Needing 5, Losing _it 5, Launch _process 5, Intercepting 5, Frugality 5, Degree _of _processing 5, Daring5, Becoming _dry 5, Bail _decision 5, Agriculture 5, Tolerating 4, Suasion 4, Stage _of _progress 4, Source _of _getting 4, Setting _out 4, Reliance _on _expectation 4, Process _stop 4, Prevent _from _having 4, Piracy 4, Invading 4, Institutions 4, Guilt _or _innocence 4, Extreme _value 4, Criminal _investigation 4, Cause _impact 4, Boundary 4, Attributed _information 4, Attending 4, Attempt _means 4, Arraignment 4, Achieving _first 4, Absorb _heat 4, Toxic _substance 3, Rope _manipulation 3, Proper _reference 3, Nuclear _process 3, Notification _of _charges 3, Installing 3, Inclination 3, Importing 3, Growing _food 3, Enforcing 3, Economy 3, Detaining 3, Corroding 3, Cause _to _continue 3, Cause _to _be _wet 3, Birth 3, Actually _occurring _entity 3, Accuracy 3, Successful _action 2, Stinginess 2, Spelling _and _pronouncing 2, Smuggling 2, Resurrection 2, Renting 2, Relation 2, Rashness 2, Rape 2, Proliferating _in _number 2, Preliminaries 2, Posing _as 2, Permitting 2, Perception _body 2, People _by _religion 2, Pardon 2, Openness 2, Meet _with 2, Import _export 2, Ground _up 2, Forging 2, Food _gathering 2, Fear 2, Familiarity 2, Explaining _the _facts 2, Drop _in _on 2, Delimitation _of _diversity 2, Cooking _creation 2, Change _of _consistency 2, Cause _to _resume 2, Carry _goods 2, Bragging 2, Be _subset _of 2, Annoyance 2, Alliance 2, Abusing 2,

C.2 Bag of Terminology used for the Bechdel Test

CLOSE, CLOSER ANGLE, CONTINUOUS, CRAWL, CAMERA, CROSSFADE, CUT TO, CUT, DISSOLVE TO, DISSOLVE, DOLLYING, ESTABLISHING SHOT, EXT., EXTERIOR, EXTREMELY LONG SHOT, XLS, FADE TO, FAVOR ON, FREEZE FRAME, INSERT, INT., INTERIOR, FRAME, INTERCUT BETWEEN, INTO FRAME, INTO VIEW, IRIS OUT, IRIS FADE OUT, IRIS FADE IN, JUMP CUT TO, JUMP, LAP DISSOLVE, DISSOLVE, MATCH CUT TO, MATCH DISSOLVE TO, MOS, O.S., OMIT, OMITTED, O.C., PUSH IN, REVERSE ANGLE, ROLL, SMASH CUT TO, SPLIT SCREEN SHOT, STOCK SHOT, SUPER, TIGHT ON, TIME CUT, V.O., WIPE TO, ZOOM