

ABSTRACT

Title of dissertation: **STRUCTURED APPROACHES FOR EXPLORING
INTERPERSONAL RELATIONSHIPS
IN NATURAL LANGUAGE TEXT**

Snigdha Chaturvedi, Doctor of Philosophy, 2016

Dissertation directed by: **Dr. Hal Daumé III**
Department of Computer Science

Human relationships have long been studied by scientists from domains like sociology, psychology, literature, etc. for understanding people’s desires, goals, actions and expected behaviors. In this dissertation we study inter-personal relationships as expressed in natural language text. Modeling inter-personal relationships from text finds application in general natural language understanding, as well as real-world domains such as social networks, discussion forums, intelligent virtual agents, etc.

We propose that the study of relationships should incorporate not only linguistic cues in text, but also the contexts in which these cues appear. Our investigations, backed by empirical evaluation, support this thesis, and demonstrate that the task benefits from using structured models that incorporate both types of information.

We present such structured models to address the task of modeling the nature of relationships between any two given characters from a narrative. To begin with, we assume that relationships are of two types: cooperative and non-cooperative. We first describe an approach to jointly infer relationships between all characters

in the narrative, and demonstrate how the task of characterizing the relationship between two characters can benefit from including information about their relationships with other characters in the narrative. We next formulate the relationship-modeling problem as a sequence prediction task to acknowledge the evolving nature of human relationships, and demonstrate the need to model the history of a relationship in predicting its evolution. Thereafter, we present a data-driven method to automatically discover various types of relationships such as familial, romantic, hostile, etc. Like before, we address the task of modeling evolving relationships but don't restrict ourselves to two types of relationships. We also demonstrate the need to incorporate not only local historical but also global context while solving this problem.

Lastly, we demonstrate a practical application of modeling inter-personal relationships in the domain of online educational discussion forums. Such forums offer opportunities for its users to interact and form deeper relationships. With this view, we address the task of identifying initiation of such deeper relationships between a student and the instructor. Specifically, we analyze contents of the forums to automatically suggest threads to the instructors that require their intervention. By highlighting scenarios that need direct instructor-student interactions, we alleviate the need for the instructor to manually peruse all threads of the forum and also assist students who have limited avenues for communicating with instructors. We do this by incorporating the discourse structure of the thread through latent variables that abstractly represent contents of individual posts and model the flow of information in the thread. Such latent structured models that incorporate the linguistic

cues without losing their context can be helpful in other related natural language understanding tasks as well. We demonstrate this by using the model for a very different task: identifying if a stated desire has been fulfilled by the end of a story.

STRUCTURED APPROACHES FOR EXPLORING
INTERPERSONAL RELATIONSHIPS IN
NATURAL LANGUAGE TEXT

by

Snigdha Chaturvedi

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor Hal Daumé III, Chair/Advisor
Professor Hector Corrada Bravo
Professor Marine Carpuat
Professor Chris Dyer
Professor Philip Resnik

© Copyright by
Snigdha Chaturvedi
2016

Dedication

To my wonderful mother

Acknowledgments

I would like to thank several people who have helped me in working on this dissertation. First and foremost, I would like to thank Hal, who has been an exceptional advisor, mentor and teacher. I could not have wished for a better advisor. I would also like to thank my committee members, Philip, Chris, Marine and Hector for their inputs and useful advice through the course of my dissertation work. Thanks also to Dan Goldwasser and Ben Shneiderman for their valuable mentorship and support. I am also grateful to all my friends and colleagues, who made my graduate school life enriching and enjoyable.

Finally, I would like to thank my parents, my husband, and my family for their unwavering encouragement and support throughout this process.

Table of Contents

List of Tables	vii
List of Figures	ix
1 Introduction	1
1.1 Thesis Statement	5
1.2 Contributions	5
1.3 Outline of this Dissertation	7
2 Background	11
2.1 Natural Language Understanding	11
2.1.1 Event-centric Natural Language Understanding	11
2.1.2 Character-centric Natural Language Understanding	13
2.2 Social Relationships	15
2.3 Relationships in Literature	16
2.4 Technical Background	18
2.4.1 Structured Prediction	18
2.4.2 Latent Variable Models	20
2.4.3 Expectation Maximization	22
3 Joint Relationship Identification in Movie Summaries	24
3.1 Problem Motivation and Definition	24
3.2 Role of Context	26
3.3 Model	26
3.4 Features	28
3.5 Data	30
3.6 Evaluation	31
3.6.1 Feature ablation	32
3.6.2 Structured vs. unstructured models	33
3.6.3 Qualitative Discussion	34
3.7 Conclusion	36

4	Modeling Evolving Relationships Between Literary Characters	38
4.1	Problem Motivation and Definition	38
4.2	Role of Context	40
4.3	Model	41
4.3.1	Segmentation Model	41
4.3.2	Semi-supervised Framework	43
4.4	Features	44
4.4.1	Content features	45
4.4.2	Transition features	48
4.5	Data	49
4.6	Evaluation	50
4.6.1	Evaluation on the SparkNotes dataset	52
4.6.2	Evaluation on the AMT dataset	53
4.6.3	Feature Ablation	54
4.6.4	Case Study	55
4.7	Conclusion	58
5	Unsupervised Modeling of Evolving Inter-character Relationships	59
5.1	Problem Motivation and Definition	59
5.2	Role of Context	61
5.3	Feature-vectors Extraction	62
5.4	Learning Relationship Sequences	65
5.4.1	GHMM	65
5.4.2	Penalized GHMM	66
5.4.3	Globally Aware GHMM	67
5.5	Data	71
5.6	Evaluation	71
5.6.1	Supervised Evaluation	73
5.6.2	Relationship Analogy Task	76
5.6.3	Coherence of Relationship States	78
5.7	Conclusion	80
6	Application: Identifying Existence of Instructor-student Relationship in MOOC Forums	81
6.1	Problem Motivation and Definition	83
6.2	Role of Context	85
6.3	Models	85
6.3.1	Logistic Regression (LR)	86
6.3.2	Linear Chain Markov Model (LCMM)	88
6.3.3	Global Chain Model (GCM)	93
6.4	Data	95
6.5	Evaluation	97
6.5.1	Quantitative Results	97
6.5.2	Visual Exploration of Categories	99
6.5.3	Choice of Number of Categories	101

6.5.4	How important are linguistic features?	104
6.6	Additional Application: Understanding Desire Fulfillment	105
6.6.1	Problem Motivation and Definition	105
6.6.2	Model: LCMM	106
6.6.3	Features	107
6.6.4	Evaluation	111
6.7	Conclusion	114
7	Conclusion	115
7.1	Summary	115
7.2	Future Work	116
7.2.1	Domain-specific Models	117
7.2.2	Enhancing Latent Variable Models	117
7.2.3	Applications to other NLP problems	118
7.2.4	Widening the definition of relationships	119
7.2.5	Relevance to Digital Humanities	121
7.2.6	Relevance to Psychological and Sociological theories	122
A	Frames Lexicon	124
B	Conformity Lexicon	130
	References	132

List of Tables

3.1	Test set results for relation classification.	34
4.1	Frame samples used by ‘Semantic Parse’ features.	47
4.2	Cross validation performances on SparkNotes data. The second order model based framework outperforms the one that uses a first order model and the unstructured baselines.	51
4.3	Performance comparison on the AMT dataset. The second order model based framework outperforms the one that uses a first order model and the unstructured models.	53
4.4	Sample character pairs (and book titles) from the clusters obtained from the summaries of the <i>Harry Potter</i> series. Pairs in clusters 1 and 3 had a cooperative and non-cooperative relationship throughout the novel (respectively). Cluster 2 contains pairs for which the relationship became non-cooperative once in the novel but then finally became cooperative.	56
5.1	Representative words for relationship states learned by the Globally Aware GHMM.	75
6.1	Held-out test set performances of chain models, LCMM and GCM, are better than that of the unstructured models, LR and J48.	97
6.2	Representative posts from the four categories learnt by our model. Due to space and privacy concerns we omit some parts of the text, indicated by “...”.	101
6.3	Feature definitions (Section 6.6.3). F1-F3 are extracted for each example while F4-F17 are extracted for each evidence.	108
6.4	Test set performances of various models reporting averaged Precision, Recall and F measure of the two classes. Our Latent Chain Markov Model, LCMM, outperforms the unstructured models (Logistic Regression, LR and Decision Trees, DT) and the majority baseline (which always predicts the majority class).	113
A.1	Lists of negative frames.	126

A.2	Lists of positive frames.	128
A.3	Lists of ambiguous frames.	129
A.4	Lists of relationship frames.	129
B.1	Lists of conforming and dissenting phrases used by the Sustenance features.	131

List of Figures

1.1	General form of our context aware relationship modeling approaches .	5
3.1	Figure depicting importance of textual as well as contextual evidence in determining relationship polarities.	25
3.2	Triadic structural features. ‘+’ signs indicate cooperative and ‘-’ indicate adversarial relationships	30
3.3	Sample annotation for a very short summary	31
3.4	Ablation study for text feature families	32
3.5	Inferred relationships for movies ‘Titanic’ (1997) and ‘Ice Age’ (2002)	35
3.6	Two relevant subsets of the inferred relationships for movie: ‘Smokin’ Aces 2: Assassins’ Ball’ (2010). The model incorrectly learns a cooperative relationship between Lester and McTeague, and a non-cooperative relationship between Agents Baker and Redstone.	36
4.1	Sample sentences from a narrative depicting evolving relationship between characters: Tom and Becky. The relationship changes from cooperative (+) to non-cooperative (-) and then back to cooperative (+). ‘. . .’ represents text omitted for brevity.	40
4.2	Ablation results on SparkNotes dataset. All represents performance with full feature-set and rest of the bars indicate performance with incrementally removing various feature-families.	55
5.1	Excerpts from the summary of the novel ‘Harry Potter and the Prisoner of Azkaban’ depicting non-stationary nature of relationship between Harry Potter and Sirius Black. Their relationship changes from hostility to cooperative alliance.	60
5.2	Various relationship feature-sets extracted for a sample sentence depicting relationship between Jim and Maria.	64

5.3	Diagram for the Globally Aware GHMM (for 2 sentences of a sequence). Feature-vector representations of individual sentences, \vec{f}_t , and that of the complete sequence (global), \vec{F} , are observed. Relationship states, r_t and global versus local choices, c_t are hidden. γ is the model's preference towards the local component. \vec{w} represents the weights of the global component (with a logistic regression form), and ϕ and π represent the local component (transition and start-state probabilities respectively). Penalty ρ is not shown for clarity. $\vec{\mu}$ and Σ are the parameters of Normal distribution. L is the number of sequences in the dataset and R is the number of relationship states.	68
5.4	Performance comparison of various models. Previously, we obtained an F-measure of 76.76 on the same dataset using supervised models (Chapter 4).	74
5.5	Global vs local preference learned by the Globally Aware GHMM with and without penalty. The model without penalty has weaker preference for a global categorization which can result in frequent shifts in relationship states within a sequence, resulting in lower accuracy. . .	76
5.6	Screenshot from human evaluation task	77
5.7	Model Precision (MP) for the Word-intrusion detection task for relationship states learned by our model. MP is high for most states indicating semantic coherence.	79
6.1	Example posts demonstrating that in order to attract instructor's attention students have to resolve to the inefficient method of getting their posts upvoted by their classmates. Therefore, there is a need to design models that could automatically identify threads on which instructors intervene.	84
6.2	Diagrams of the Linear Chain Markov Model (LCMM) and the Global Chain Model (GCM). p_i , r and $\phi(t)$ are observed and h_i are the latent variables. p_i and h_i represent the posts of the thread and their latent categories respectively; r represents the instructor's intervention and $\phi(t)$ represent the non-structural features used by the logistic regression model.	89
6.3	Instructor's intervention decision process for the Linear Chain Markov Model.	90
6.4	Visualization of lexical contents of the categories learnt by our model from the GHC dataset. Each row is a category and each column represents a feature vector. A light color represents high values while lower values are represented by darker shades. Dark monochromatic columns are used to better separate the five feature clusters, F1-F5, which represent words that are common in thanking, logistics-related, introductory, syllabus related and miscellaneous posts respectively. Categories 1,2,3 and 4 are dominated by F2, F4, F1 and F3 respectively indicating a semantic segregation of posts by our model's categories.	98

6.5	Cross validation performances of the two models with increasing number of categories.	102
6.6	Cross validation performances of the various feature types for the two datasets.	103
6.7	Example consisting of: a Desire Expression, Evidence fragments (e1. . .e5) and a binary Desire Fulfillment Status. The Desire-subject is marked in bold fonts in the Desire-expression.	106
6.8	Artificial example indicating feature utility. The Desire-subject mentions are marked in blue, actions in bold and emotions in italics. Discourse feature is underlined.	109
7.1	Network Inferred from the Wikipedia article on 2003 Invasion of Iraq. The + and - signs indicate cooperative and non-cooperative relationships respectively.	122

Chapter 1: Introduction

Relationships and social bonds are a fundamental aspect of human nature. They serve not only as manifestations of our social temperament, but are also consequences of evolutionary design which has led to our emergence as a successful species [3]. In particular, human relationships exhibit a range of phenomenon such as family, friendship, hostility, romantic love, formality, authority, etc. Understanding human interactions and relationships is hence necessary for understanding people's actions, behavior and goals. Due to the insights offered by relationships into human conduct and behavior, scientists from diverse fields like sociology [4, 60], psychology [14] and literature [87] have been interested in studying human relationships. As early as 4th century BC, Plato suggested that love (representing human bonds) directs the worlds of Gods as well as humans [86]. In this thesis, we adopt a computational route to studying relationships, and model inter-personal relationships from digitally available natural language text. Such texts, especially narratives, are rich reflections of social relationships; and hence provide an attractive medium for their analysis.

A narrative is an account of events or experiences, true or fictitious; and forms an indispensable part of our day-to-day lives. Ubiquitous examples of real or

fictional narratives include novels, movies, TV shows, news articles, blog articles, discussion forums, etc. Children learn and internalize societal norms and values through narratives in stories [30, 43]. As they grow older, they learn to increasingly share desires, experiences, and information through narratives [18]. In short, humans use narratives as a tool to express their (or others) interactions and relationships with others. In this sense, automatically reading and understanding narratives can serve as a platform for automatic systems to study inter-personal relationships and learn of human behavior and norms. The complexity of these narratives ranges from simple fables to abstract literary novels and news articles and interactive discussion forums, and interpreting and comprehending them can not only assist in acquiring background knowledge or common-sense but is essential for the general task of Natural Language Understanding (NLU).

The field of computational linguistics has long been interested in algorithmically understanding, representing and generating narratives [71]. In this context, the focus of much of early research has been on understanding text, including narratives, from the perspective of events mentioned in them [94, 65]. Prior research assumes that events are the core building blocks of representing text and interprets a text's meaning as a sequence of events that are directly observed or inferred from the text. This style of research focuses more on what happened (plot) and less on who did it (characters). For example, in 'John went to play basketball.' the focus would be on the act of playing basketball with John acting as an ancillary participant in the event, rather than viewing John as the protagonist who intentionally performs an act. An alternative line of research flips this attention on studying characters in a

text and what they do [38]. Recent approaches have proposed studying text from the perspective of people involved [9, 40]. E.g., reading Wikipedia to gain information related to an individual’s life [10]; incorporating attributes of authors while understanding text [80, 93]; representing documents through their readers [39]; understanding roles of users of an email system [75]; etc. Especially, while studying fictional narratives, there have been several character-centric approaches that explain the set of observed actions in the narrative using characters’ *personas* or *roles* and their expected behavior in those roles [105, 11, 12]. However, people’s roles, attributes and actions are indispensably connected to and influence others around them. So, rather than analyzing personal attributes in isolation, **this dissertation proposes studying *interactions* or *relationships* between people.**

Existing research on analysis of people’s relationships in text has been limited to simple schemes. These include constructing social networks of characters in a novel based on their co-occurrences in quoted conversations [42] and social events [2], and networks of participants of online-discussion forums [54] and email systems [75]. These methods primarily focus on identifying existence of relationships. Previous research has also focused on methods to explore certain specific aspects of relations such as power or authority in email conversations [17] and Supreme court dialogs and discussion among Wikipedia editors [32], address formality [64], and sentiment [55] in online discussion forums.

Modeling relationships has utility for several real-world tasks. For example, predicting plausible relationships between people, using their posts or messages, can help social networking platforms and discussion forums in personalizing news

feeds, predicting virality, suggesting friends or topics of interest, etc. for a particular user. In doing so they could incorporate evolving nature of social relationships by focusing on activities of current friends [7, 63, 35, 6] and paying lesser attention to activities of connections who are no longer of much interest to the user. Similarly in email systems, understanding how users interact with each other can help in making smarter email systems [50, 16]. For example, determining the level of formality of the textual content of an email being composed, and also identifying plausible recipients who are in a formal relationship with the user can help in making better recipient recommendations. Similarly arranging emails according to the nature of relationship of the recipient with the sender can help in better managing the recipient's inbox by clustering emails from friends in a different cluster than emails from office-colleagues. In general, identifying the nature of relationships between individuals can assist automatic understanding of text by explaining the actions of people mentioned in the text and building expectations of their behavior towards others [2, 42].

In particular, predictive models for inferring relationships from natural text can be especially use for researchers from the digital humanities domain. Though we make simplifying assumptions about definition of relationships to focus on how they can be inferred from text, applications based on our ideas could range from stylistic analysis and author preferences in depicting character relationships in literary works to mining geopolitical or business relations from historical archives or correspondences.

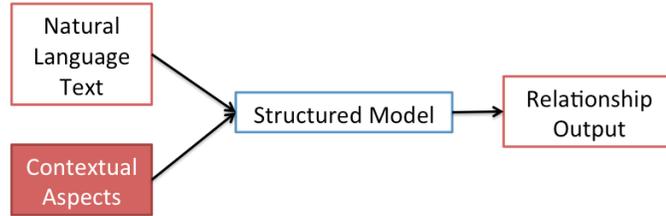


Figure 1.1: General form of our context aware relationship modeling approaches

1.1 Thesis Statement

In this dissertation, I propose that **incorporating linguistic cues as well as the context in which they appear helps in modeling the nature and the existence of inter-personal relationships in text**. Specifically, I address aspects of modeling relationships between people in narratives such as movie and novel summaries; and provide a real-world application of predicting existence of inter-personal relationships in discussion forums.

To solve these problems, I provide structured models based on hand-crafted features that depend on the input text, candidate output and the context in the text (Figure 1.1).

1.2 Contributions

The contributions of this work are summarized as:

1. We formulate the problem of modeling inter-personal relationships in various settings including fictional narratives as well as real-world domains like discussion forums.

2. We present methods to model nature of relationships between pairs of characters in a narrative summary, and demonstrate the utility of incorporating information about characters' relations with other characters in the narrative (Chapter 3).
3. We then formulate the task of modeling the evolving nature of inter-personal relationships by addressing this as a supervised sequence prediction task which is able to incorporate historical context (Chapter 4).
4. We also allow a data-driven exploration of various types of relationships while modeling their evolving nature, and simultaneously demonstrate the importance of including local as well as global context (Chapter 5).
5. We present a real-world application of our idea by addressing the problem of analyzing the contents of educational discussion forums to predict when the instructor would reply on a thread. While solving this problem we demonstrate the need for sequential models that can view individual posts in context of other posts of the thread. We additionally validate the utility of our model by applying it to another task of identifying if a desire expressed in a short paragraph was fulfilled (Chapter 6).
6. We annotate and release two datasets of inter-personal relationships. The first dataset consists of relationship annotations in about 150 movie summaries (Chapter 3), and the second dataset of novel summaries depicts evolving relationships between about 100 pairs of characters (Chapter 4).

7. We also release two other datasets of textual paragraphs containing expression of a desire and annotated with desire fulfillment (Section 6.6).

1.3 Outline of this Dissertation

With this goal of inferring relationships from text in mind, we start with fictional narratives (like movies and novels) as they are relatively well-studied and are more self-contained as compared to real narratives like news articles. We then also demonstrate a real-world application while analyzing discussion forums.

We start with a short description of previous work that has been done in the general area of understanding natural language text (Chapter 2). In this chapter we provide a brief introduction to the two major types of techniques explored in this field: event-centric and character-centric natural language understanding.

We then begin with addressing the problem of characterizing the nature of relationships between pairs of characters in movie summaries in Chapter 3. While real-life relationships have multiple facets (power, influence, friendship, animosity, romance, etc.), we make a simplifying assumption of a binary relationship status (cooperative/non-cooperative) in this chapter; and instead focus on teasing apart the connection between text and relationships. In other words, this assumption makes the problem computationally tractable and allows us to focus on analyzing the different types and sources of information for inferring relationships from text. For example, while analyzing a typical romantic movie, we want to identify that the protagonist is in a non-cooperative relationship with the villain because they

try to engage in actions that harm each other, while he is in a cooperative relationship with his lover because they assist each other. While solving this problem, we demonstrate the importance of understanding the relationships between the two characters *in context of their relationships with other characters* in the narrative. In other words, rather than looking at relationship between pairs of characters in isolation, we also incorporate structural cues from the social community in the narrative by incorporating transitive phenomena like ‘friend of a friend is a friend’, ‘common enemy’, etc. Specifically, if two people have a common friend or a common enemy, they are more likely to be in a cooperative relationship.

However in real-world, relationships between people can change with time. In Chapter 4, we model the evolution of inter-personal relationships in novel summaries. Like the previous chapter we assume that relationships are only of two types: cooperative/non-cooperative. With this intuition, we present the relationship-modeling problem as a structured prediction task that can incorporate *historical context* while modeling the evolving nature of relationships between characters. For example, in the novel ‘Adventures of Tom Sawyer’, Tom is in love with Becky Thatcher. However there is a period in the novel when their relationship sours (see Figure 4.1 in Chapter 4). In this work, we want to identify their relationship as not just a single (most-likely) cooperative state but a sequence of cooperative, non-cooperative and then back to cooperative states depicting their relationships at various points in the novel. We empirically show how this problem can benefit from structured models that apart from including linguistic cues about a relationship’s nature can also remember a history of the characters’ relationship in the novel.

The previous chapters make a limiting assumption about types of relationships—they are binary in nature (cooperative/non-cooperative). In the next chapter, Chapter 5, we present an unsupervised method that relaxes this constraint, and allows a data-driven inference of these inter-character relationship types. We assume that the textual excerpts, that serve as our evidences for modeling relationships, belong to discrete latent clusters. Each cluster represents a single relationship-type identifiable by the frequent words in the cluster. For example, a cluster containing words like father, mother, brother is likely to represent familial relationships. We focus on the problem of modeling evolving inter-personal relationships like before, and while addressing this problem we demonstrate the importance of incorporating a local as well as a *global context* of inter-character relationships.

Thereafter in Chapter 6, we present real-world applications of our work on interaction modeling in the domain of MOOCs (Massive Open Online Courses) forums. These forums serve as platforms for their users to interact or develop relationships. In this work, we focus on the student-instructor relationship, and viewing their textual exchanges on the forum as cues of development of their relationship we address the task of predicting which threads are likely to be in need of an instructor’s intervention. This implicitly models the intervention of an instructor in a discussion thread as the initiation of a stronger relationship with the student who started the thread. By doing this, we alleviate the need for MOOC’s instructional staff to manually inspect all threads in forums, which is prohibitively time consuming owing to typically large sizes of MOOC forums. We propose two models that include *context in form of the linear chain structure of the threads* and empirically demonstrate that

they outperform structure-unaware models. We also apply our model to a different problem of identifying if a desire expressed in a short narrative subtext is fulfilled. In this case, the model extracts cues about the likely desire fulfillment status from individual sentences of the paragraph while viewing them *in context of each other*.

Lastly in Chapter 7, we summarize our contributions and present avenues for future work in this field.

Chapter 2: Background

2.1 Natural Language Understanding

This section provides a brief overview of prominent event-centric and character-centric approaches to understanding narratives.

2.1.1 Event-centric Natural Language Understanding

Understanding natural language text has attracted linguistics for a long time. Some of the earliest works have focused on understanding text by designing knowledge structures of the events described in the text. These methods have focused on representing text using structured sequences of prototypical events, their participants and causal relationships describing them called scripts [94] or Fillmorean frames. For example, these methods argue that understanding a text document about eating at a restaurant involves understanding the ‘restaurant script’ which describes the usual events in the process (sitting, ordering, bringing food, eating, etc.), their relative orders (ordering happens before eating, etc.), participants (customer, waiters, etc.), preconditions and results. Frames [8] are also ‘script-like’ structures that are hierarchically arranged and represent mental concepts (e.g. commercial

transaction) and their participants (e.g. buyer, seller).

Another one of the earliest representations of narratives include plot-units [65]. This representation attempted to summarize a narrative using several ‘plot-units’, each of which consists of a graphical structure involving *affect states*. An affect state occurs with respect to single character and indicates a positive reaction (+); or a negative reaction (-) to a situation; or a neutral affect ‘Mental’ state (M) indicating a combination of desires and intentions. The events in the narrative are described using causal links between affect states of the various characters involved. Lehnert uses four types of links describing causalities behind mental states (Motivation), intentions behind events (Actualization), displaced impact of an event terminating another state (Termination), and equivalence behind multiple perspectives of a single state (Equivalence). Using this structure, she represents common plot-units describing success or failure in fulfilling an intention, an instance of honored promise, an act of kindness, etc. For example, a success (or failure) can be represented using an actualization link between an M state and a + (or -) state. The narrative is then described using these plot-units.

However, the representational structures used in these methods are hand-coded and also domain-dependent in some cases. Later works have focused on automatically learning these structures. Mooney and DeJong [79] presented a non-statistical method of automatically inducing scripts from unstructured text. Chambers and Jurafsky [23] present a three-step statistical process to learn partially ordered set of events related to a protagonist, which they refer to as the ‘narrative chains’. The directed edges between the events in a chain represent the temporal relationships

between them. They use a novel application of pairwise mutual information and temporal relation learning to learn these chains. Later, they build upon this idea to represent events related to all characters in the narrative by building ‘narrative schemas’ [22]. The arguments of events in the schemas are filled by participants’ semantic roles. They also present methods to learn these schemas probabilistically albeit without the temporal ordering [21]. Another approach that focuses on learning scripts as a bag of related events was proposed by Cheung et al. [29]. Regneri et al. [91] propose a graph learning approach based on Multiple Sequence Alignment methods popular in bioinformatics, and a semantic similarity function to cluster event sequences describing a scenario (e.g. eating at a restaurant) into a directed graph. Recent methods have also used Recurrent Neural Networks [84] and Hidden Markov Models [83] to learn scripts from unstructured text.

Similarly, statistical methods have also been employed to produce plot-units representations of narrative text [53], and to learn plot-units from folktales using Bayesian model merging, a grammar induction technique [47].

2.1.2 Character-centric Natural Language Understanding

An alternate perspective attempts to understand text, especially narratives, from the viewpoint of characters. Most approaches in this field are based on Vladimir Propp’s Structuralist narrative theory [89] on character-roles in folktales. According to Propp’s theory, each role (e.g. a hero, a villain, etc.) has a *sphere of actions*, which defines the core actions of the character fulfilling that role. For example, irrespective

of what character fulfills the role of the villain in a story, his/her actions will be centered around malicious acts, violence, cheating, etc. Based on this theory, Valls-Vargas et al. [105] have studied actions of various characters of Russian folktales to assign roles to them. Similarly, Bamman et al. [11] and [12] present probabilistic methods to learn roles or *personas* of the various characters described in movie plots and novels (respectively). However, unlike previous methods, they do not pre-define the meanings of roles, but instead learn the definitions of *personas* in a data-driven manner.

Wilenskys PAM (Plan Applier Mechanism) system [109] is also related to this perspective of understanding narratives. PAM understands a narrative by treating characters as intentional agents. It interprets every action of a character as instantiating some goal and the character's plan to achieve that goal. Such methods that treat characters as intentional agents can be useful in explaining characters' motivations and actions, and also help text generation systems that program interesting behaviors for agents [72].

This character-centric understanding of text motivates the problems and methods described in this dissertation. However, instead of viewing characters as assuming specific roles, we study relationships between them. In the next two sections we discuss relationships in detail.

2.2 Social Relationships

Max Weber [106] uses social actions to describe social relationships. He defines actions as human activities to which the actor attaches a subjective meaning. A social action is one whose subjective meaning takes account of the behavior of others and is oriented in its course.

Social relationships are then defined on the basis of patterns of social actions [106, 62]. They denote behavior of multiple actors such that the action of each takes account of that of the others and is oriented in these terms. A social relationship essentially assumes mutual orientation of the actors and the content of mutual orientation could be characterized by friendship, anger, sexual attraction, competition, conflict, or economic exchange. It is possible that the concerned parties attach different ‘meanings’ to their actions in which case the relationship will be asymmetrical from the points of view of the two parties. A social relationship can be of a fleeting character or permanent. In case of a permanent relationship, there is a probability of repetition of the behavior, which corresponds to its subjective meaning. This regularly repeated behavior eventually come to be expected, and forms patterns that we may call institutions which include formal institutions like school and workplace as well as voluntary ones like friendship, or peer-group. The content of these regularized relationships can become ‘maxims’ - forms of actions that are adhered to or are expected to be adhered to. When these maxims become more regularized they develop into customs, or even laws. Lastly, subjective meanings of relationships can change due to mutual agreement, thus a political relationship once

based on solidarity may develop into a conflict of interests.

The work presented in this dissertation is guided by this definition of relationships. We use character's or people's actions (apart from other indicators) as features that help us determine the nature of relationships. This is especially true of our data driven method presented in Chapter 5 which, in some sense, exploits repeated actions by individuals to discover norms or expected behavior that constitute the meaning of a relationship. In Chapters 4 and 5, we also allow relationships to be fleeting or change with the progress of the narrative. However, in this work we do not study asymmetric relationships and leave that as possible future work.

2.3 Relationships in Literature

In this section we present some previous works that study social relationships in narrative texts. While this domain is large, we provide representative examples of various types of work to familiarize the reader with this domain.

Polhemus [87] focused on erotic love and studied works by several authors including Jane Austen, Walter Scott, the Brontës, Dickens, George Eliot, Trollope, Thomas Hardy, Joyce, D. H. Lawrence, Virginia Woolf, and Samuel Beckett. Polhemus draws parallels between religious faith and erotic love described by these authors which ranges from decorous to bold portrayal of love.

Several works have also studied romantic relationships in works of specific authors. For example, White [108] studied evolution and history of romantic dramas that constitute early works of Shakespeare; previous works [100, 57] studied the love-

hate relationships as depicted in Jane Austen’s novels; etc.

Apart from romantic relationships, literary work has also focused on other types of relationships. Abel [1] used object relations theory to study dynamics of female friendships in various novels. Frith [49] provided an in-depth study of relationships as depicted in works by female novelists. They analyzed female friendship as a formative experience and its role in shaping gendered identity. Ford [48] analyzed father-daughter relationships, especially in context of a suitor for the daughter, in works of several writers. Previous work have also studied the implications of such fictional father-daughter relations on the family unit as well as the extent to which an author’s work mirrors the social structure at that time [15].

There has also been work on closely studying and analyzing specific types of relationships as depicted by certain novelists or relationships between specific characters in certain novels. These include study of female friendship between characters of Matilde Serao [44]; the complicated relationship between Catherine and Heathcliff in *Wuthering Heights* [52]; etc.

Our goal in this dissertation is not to provide such ‘close reading’ or in-depth analysis of specific novels or authors; but to provide methods that could lead to ‘distant reading’ of several narratives simultaneously, and could infer relationship-types automatically. As pointed out before,¹ specific assumptions made by computational models in the domain of social sciences might limit their direct utility. However, such methods can be extended for use in digital humanities and social sciences.

¹<http://languagelog.ldc.upenn.edu/n11/?p=6968>

2.4 Technical Background

We now describe the prominent Machine Learning tools that motivate the methods described in this dissertation.

2.4.1 Structured Prediction

Structured prediction refers to a class of machine learning methods in which the output of the algorithm is structured in nature rather than being a scalar binary, categorical or real value. The input to the algorithm can be structured or unstructured. Structured prediction finds numerous applications in Natural Language Processing, Computer Vision, Computational Biology etc.

Owing to the interaction between the various parts of the structured output, the number of its possible configurations is often very large, making the task of inference (or finding the best output) non-trivial. For this reason, most structured prediction models are often more complicated than their unstructured counterparts, employing dynamic programming or approximate methods for inference. These methods often assume joint features that depend on the input as well as the candidate output, and most commonly maximize a linear scoring function based on the dot product of features and their weights. The approaches proposed in this dissertation are based on two major structured prediction models: Structured Perceptron and Structured SVM.

Structured Perceptron [31] learns the weights of the joint features using an iterative method based on the classic perceptron algorithm for learning linear boundaries.

In particular, given an input, \mathbf{x} , and a candidate output, \mathbf{y} , it assigns a ‘score’ to this candidate output as follows:

$$\text{score} = \mathbf{w} \cdot \phi(\mathbf{x}, \mathbf{y})$$

where $\phi(\mathbf{x}, \mathbf{y})$ represents joint features that can depend on the input as well as the candidate output, and \mathbf{w} are the candidate weights. Algorithm 1 shows the procedure for learning the feature weights. The inference procedure usually employs a dynamic programming based method (such as Viterbi) or an approximate method (such as belief propagation).

Algorithm 1 Training algorithm for Structured Perceptron

- 1: **Inputs:** Training examples $(\mathbf{x}_i, \mathbf{y}_i)$
 - 2: **Output:** Feature Weights \mathbf{w}
 - 3: **Initialization:** Set $\mathbf{w} = 0$
 - 4: **for** $t : 1$ to T , $i : 1$ to N **do**
 - 5: $\hat{\mathbf{y}}_i = \arg \max_{\mathbf{y} \in \mathcal{Y}} [\mathbf{w}_t \cdot \phi(\mathbf{x}_i, \mathbf{y})]$
 - 6: If $(\hat{\mathbf{y}}_i \neq \mathbf{y}_i)$ $\mathbf{w}_{t+1} = \mathbf{w}_t + \phi(\mathbf{x}_i, \mathbf{y}_i) - \phi(\mathbf{x}_i, \hat{\mathbf{y}}_i)$
 - 7: **end for**
 - 8: return \mathbf{w}
-

The structured perceptron algorithm has convergence guarantees similar to the traditional perceptron method. Collins [31] showed that if the training set is separable with margin δ , then the above mentioned algorithm (Algorithm 1) makes at most R^2/δ^2 mistakes before convergence, where R is a constant such that

$$\forall i, \forall \mathbf{y} \in \mathcal{Y}, \|\phi(\mathbf{x}_i, \mathbf{y}_i) - \phi(\mathbf{x}_i, \mathbf{y})\| \leq R.$$

Structured SVM is a method that incorporates the idea of margin-based learning for structured prediction. Like before, the goal is to learn a linear prediction rule of the form:

$$f_w(\mathbf{x}) = \arg \max_{\mathbf{y} \in \mathcal{Y}} [\mathbf{w} \cdot \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})]$$

Training a Structural SVMs involves solving the following optimization [104, 61]:

$$\min_w \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N l(y_i, \hat{y}_i(w))$$

where, $\hat{y}_i(w) = \arg \max_{y \in \mathcal{Y}} w \cdot \phi(x_i, y)$, and l is a user supplied loss function.

However, solving this problem is difficult due to the non-convex and discontinuous nature of typical forms for l . Tsochantaridis et al. [104] overcome these difficulties by replacing the loss function l with a piecewise linear convex upper bound (margin rescaling). Thus training this method now requires solving the following convex optimization problem:

$$\min_w \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \left[\max_{\hat{y} \in \mathcal{Y}} [l(y_i, \hat{y}) + w \cdot \phi(x_i, \hat{y})] - w \cdot \phi(x_i, y_i) \right]$$

2.4.2 Latent Variable Models

Latent Variable Models are a broad class of methods that assume that apart from the observed input and output variables, there exist another set of variables which are hidden or latent and so are not directly observed. Hence instead of directly generating/predicting the output from the input, these methods additionally assume one or more layers of latent variables that are directly related to the input and the output. Assuming their existence may not only help in predicting better outputs, but also in explaining the data.

There have been several approaches that couple latent variables with structured predictions. Examples of such methods include Structured Perceptron [102], and Latent Structural SVMs [113]. Both these methods assume a set of (structured) latent variables, h . The features for the model, $\phi(x, y, h)$, are extracted not only from the input, x , and a candidate output, y but also on this latent variables h . Methods like these help in capturing latent dependencies and hidden sub-structure [103]. The formulation for Latent Structured SVM is similar to that of Structured SVM. The goal is to now learn linear prediction rule of the form

$$f_w(\mathbf{x}) = \arg \max_{(\mathbf{y}, \mathbf{h}) \in \mathcal{Y} \times \mathcal{H}} [\mathbf{w} \cdot \phi(\mathbf{x}, \mathbf{y}, \mathbf{h})]$$

The objective function is now written as:

$$\min_w \left[\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \max_{(\hat{y}, \hat{h}) \in \mathcal{Y} \times \mathcal{H}} [l(y_i, \hat{y}, \hat{h}) + w \cdot \phi(x_i, \hat{y}, \hat{h})] - C \sum_{i=1}^N \left[\max_{h \in \mathcal{H}} w \cdot \phi(x_i, y_i, h) \right] \right]$$

Hidden Markov Model (HMM) [13] is another latent structured prediction method typically used for sequential or temporal input (chain of observations), $\mathbf{x} = \langle x^1, x^2, x^3, \dots, x^T \rangle$, and output (a chain of states), $\mathbf{y} = \langle y^1, y^2, y^3, \dots, y^T \rangle$. An example of such a problem includes Part-of-speech (POS) tagging where the input is a sentence (viewed as a chain of words) and the goal is to label each word with a POS tag. Thus the structured output is a chain of POS tags. The model makes Markov assumption which means that at any point in the process, the current output state depends on previous few states. In a simple Markov Model, the states are observed while in a Hidden Markov Model the output states are hidden or latent. Assuming a first order Markov assumption for simplicity, HMM defines the joint distribution

over input and output sequences as:

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^T p(y^t|y^{t-1})p(x^t|y^t)$$

where, $p(y^t|y^{t-1})$ and $p(x^t|y^t)$ are the state transition and observation emission probabilities respectively. The parameters of an HMM, the emission and the transition probabilities, and are usually learned using the Baum-Welch algorithm.

2.4.3 Expectation Maximization

Expectation-Maximization (EM) [36] is an optimization algorithm used to learn parameters of a model which depends on latent variables. It is applicable when the objective function resembles a likelihood in presence of the missing data. It is an iterative process consisting of two steps: the E step in which the current parameters are used to obtain an expectation of the sufficient statistics of the missing variables; and the M step in which these expectations are used to update parameters. This method is guaranteed to converge to a local optimum of the likelihood.

A variant of the above mentioned method is hard-EM. In this method, instead of computing expectations, the hidden variables are assigned their ‘most likely’ value. Another variation of EM, Generalized (Incomplete) EM, is used in cases when finding the maximum likelihood parameters in the M-step might be difficult even with the given expectations (or completed data). In such a case, M-step can proceed by improving the likelihood slightly (for example by using a gradient step).

In this dissertation we use structured latent variables to help us model the underlying structure of the textual data while inferring relationships. We jointly

learn/assign these latent variables along with the rest of the data. However, the learning method depends on the nature of the latent variable and the problem setting and will be discussed the corresponding chapters.

Chapter 3: Joint Relationship Identification in Movie Summaries

Work described in this chapter was published in AAI 2016 [97]

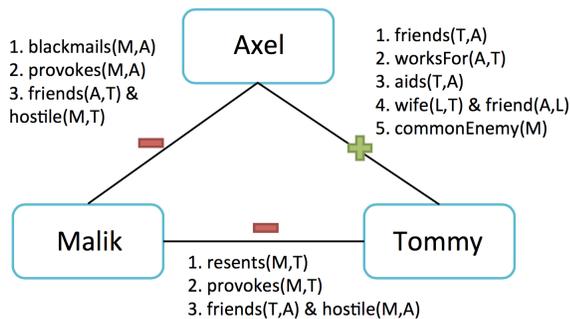
In this chapter we start with the problem of identifying relationships between characters in a narrative. Our domain is movie summaries and we make a simplifying assumption that relationships can be of only two types: cooperative vs non-cooperative. We propose to include structural cues from the social community in the narrative to characterize the relationships between individual character pairs in *context of their relationships with other characters*. We demonstrate that inference of relationship polarities can be significantly improved by considering their relationships with other characters.

3.1 Problem Motivation and Definition

The character-centric approaches for understanding text believe that comprehending a narrative requires the ability to interpret character roles and their expected behavior towards other characters in those roles. While existing approaches can identify characters types or personas [11, 12, 105], they do not model *relationships* between characters in a narrative. We present methods to model inter-character relationships instead of characters' roles.

Young drifter Axel Nordmann goes to work in a gang of stevedores headed by Charlie Malik, a vicious bully, and is befriended by Tommy Tyler, who also supervises a stevedore gang. Malik resents blacks in positions of authority, and is antagonized when Axel goes to work for Tommy. Axel moves into Tommy's neighborhood and becomes friends with Tommy's wife Lucy. Axel is hiding something, and it emerges that he is a deserter from the United States Army. Malik is aware of that, and is extorting money from him. Malik frequently tries to provoke Tommy and Axel into fights, with Tommy coming to Axel's aid ...

(a) Sample summary extract for the 1957 movie 'Edge of the City'.



(b) Inferred relationship polarities with supporting evidences.

Figure 3.1: Figure depicting importance of textual as well as contextual evidence in determining relationship polarities.

Specifically, given two characters from a movie summary, we address the problem of automatically identifying binary (cooperative and non-cooperative) relationships between them based on an analysis of the content of the narrative summary.

3.2 Role of Context

Following previous works, we incorporate several linguistic cues for solving this problem (Section 3.4). However, apart from linguistic knowledge, there is a lot of contextual information to be exploited in this problem. For example, let us consider the plot summary in Figure 3.1a (condensed here for brevity). In this passage, the relations between the principal characters are explicated through a combination of cues, as seen in Figure 3.1b. For instance, one can infer that Alex (A) and Tommy (T) have a cooperative relationship through a combination of the following observations (among others): (1) T initially ‘befriends’ A, (2) A works for T, and its connotation that A is likely to cooperate with T, (3) T aids A in fights, (4) A is a friend of T’s wife, (5) A and T have a common adversary. In particular, we note that cues (4) and (5) cannot be extracted from looking at the relation between A and T in isolation, but depend on their relations with others. In this work, we show that such indirect structural cues can be very significant for inference of inter-character relationships. In other words, instead of independently identifying the nature of relationship between two characters, it is important to view their relationship in context of their relations with other characters in the narrative.

3.3 Model

The example presented above suggests that a joint inference model that incorporates both context and text would perform better than one that considers pairwise

relations in isolation. Therefore, we formulate the problem of relation classification as a structured prediction task, where we attempt to *jointly* infer the collective assignment of relation-labels for all pairs of characters in a narrative (movie summary).

Let \mathbf{x} denote a narrative document for which we want to infer relationship structure \mathbf{y} . We could think of \mathbf{x} as a graph with characters as nodes, and relationships between characters corresponding to edge-labels. We assume a supervised setting where we have labeled training set and we consider linear structured classifier of the form:

$$h_{\mathbf{w}} = \arg \max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \mathbf{w}^T \phi(\mathbf{x}, \mathbf{y}) \quad (3.1)$$

Here, $\phi(\mathbf{x}, \mathbf{y})$ is a feature representation for the narrative and a relation-assignment pairing and \mathbf{w} is a vector of feature-weights which can be learnt using a voted structured perceptron training algorithm [31]. Finding the best assignment corresponds to the decoding problem, i.e. finding the highest scoring assignment under a given model.

Structured Perceptrons have been conventionally used for simple structured inputs, which are amenable to efficient dynamic programming inference algorithms. This is because updates require exact inference over an exponentially large space (solving the decoding problem in Equation 3.1), and updates from inexact search can violate convergence properties. In a problem like ours where the output is a network, finding the best scoring assignment (exact inference) can be computationally prohibitive if the network is large. However, Huang et al. [59] show that exact inference

is not necessary for convergence as long as we can guarantee perceptron updates only on ‘violations’, i.e. when the score of an assignment from the inference procedure is higher than the score for the ground truth labeling. This essentially means that for larger networks, it is not necessary to perform exact inference. Instead, any assignment whose score is greater than that of the ground-truth assignment can be used to update the model-parameters. Secondly, our structural features are based on triads of relationships but for most narratives, the character graphs are sparse (not fully connected) because summaries do not report the relationships/interactions between all pairs of characters. This means that for the vast majority of narratives in the dataset, the inference problem can be exactly solved. While training our structured perceptron model we utilize these two observations.

3.4 Features

Our feature set, $\phi(\mathbf{x}, \mathbf{y})$, consists of two types of features: linguistic and structural. Our linguistic features identified the actions by the characters in the narrative and the narrator’s bias while describing those actions. We also obtained connotation [46], sentiment [69] and prior-polarity [110] of these words (identifying the features) when needed. The feature-extraction process is described in detail in Section 4.4.

On the other hand, our structural features focused on configurations of relationships among triads of characters. We briefly characterize the triadic features with our informal appellations for them in Figure 3.2. These features based on the

Structural Balance Theory [56, 20]. This theory, originally proposed in Social psychology in 1940s, analyzes signed triads of individuals and posits that certain triads are more likely and hence more prevalent in real networks than others. Specifically, triads with three mutual friends (‘Clique’ in Figure 3.2) or with two friends and a common enemy (‘Common Enemy’ in Figure 3.2) are more prevalent than the other two triads (‘Love triangle’ and ‘Tuel’ in Figure 3.2). Davis [34] later proposed a variant of this theory, the *Weak Structural Balance* theory, which posits that only the triad with exactly two friends (‘Love triangle’ in Figure 3.2) is unlikely in a real network. The other three triads are plausible. These triads have been previously used to empirically analyze signed triads in social networks [66, 107].

In some domains, observed relations between entities can directly imply unknown relations among others due to natural orderings. For example, temporal relations among events yield natural transitive constraints. For the current task, such constraints do not apply. While structural regularities like ‘a friend of a friend is a friend’ might be prevalent, these configurations are not logically entailed; and affinities for such structural regularities must be directly learnt from the observed data. Values of our structural features (Figure 3.2) consist of the number of such configurations in any assignment. The empirical affinities for such configurations, as reflected in corresponding weights can then be learnt from the data.

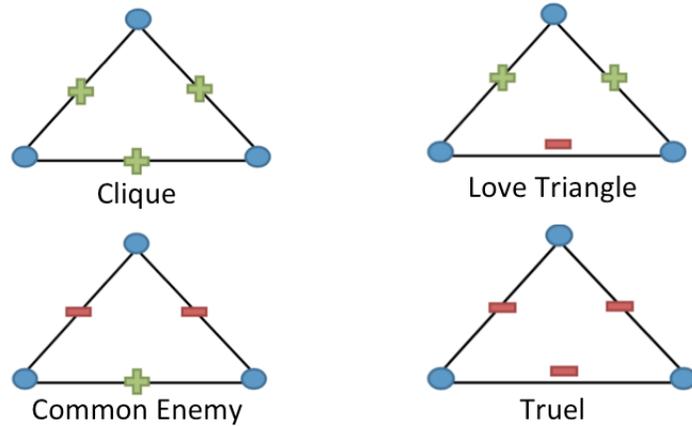


Figure 3.2: Triadic structural features. ‘+’ signs indicate cooperative and ‘-’ indicate adversarial relationships

3.5 Data

We processed the CMU Movie Summary corpus, a collection of movie plot summaries from Wikipedia, along with aligned meta-data [11]; and set up an online annotation task using BRAT [98]. We use Stanford Core NLP pipeline to identify named entities (in particular person names) and coreferent mentions of those entities.

Annotators could choose pairs of characters in a summary (screenshot of a short sample from the BRAT annotation interface shown in Fig 3.3), and characterize a relationship between them on an ordinal five-point Likert scale labeled as 1.‘Hostile’, 2.‘Adversarial’, 3.‘Neutral’, 4.‘Cooperative’ or 5.‘Friendly’. Annotators were asked to base their judgment of relationship characterization on the presented movie summary only. This resulted in a dataset of 153 movie summaries, consisting of 1044 character relationship annotations.

For evaluation, we aggregated ‘hostile’ and ‘adversarial’ edge-labels, and ‘friendly’

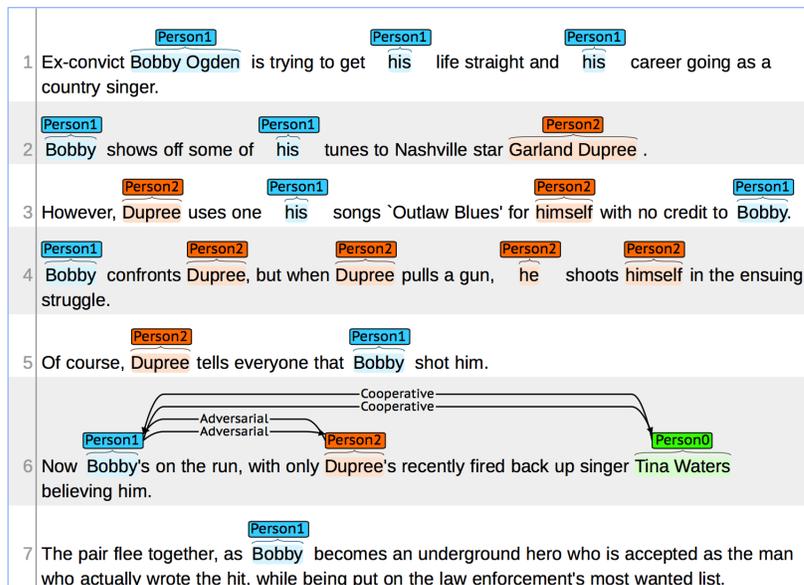


Figure 3.3: Sample annotation for a very short summary

and ‘cooperative’ edge-labels to have two classes (neutral annotations were sparse, and ignored). Of these, 58% of the relations were classified as cooperative or friendly, while 42% were hostile or adversarial. The estimated annotator agreement (Cohen’s Kappa) for the collapsed classes on a small subset of the data was 0.89.

3.6 Evaluation

In this section, we discuss quantitative and qualitative evaluation of our methods. First, we make an ablation study to assess the relative importance of families of text-based features. We then make a comparative evaluation of our methods in recovering gold-standard annotations on a held-out test set of movie summaries. Finally, we qualitatively analyze the performance of the model, and briefly discuss common sources of errors.

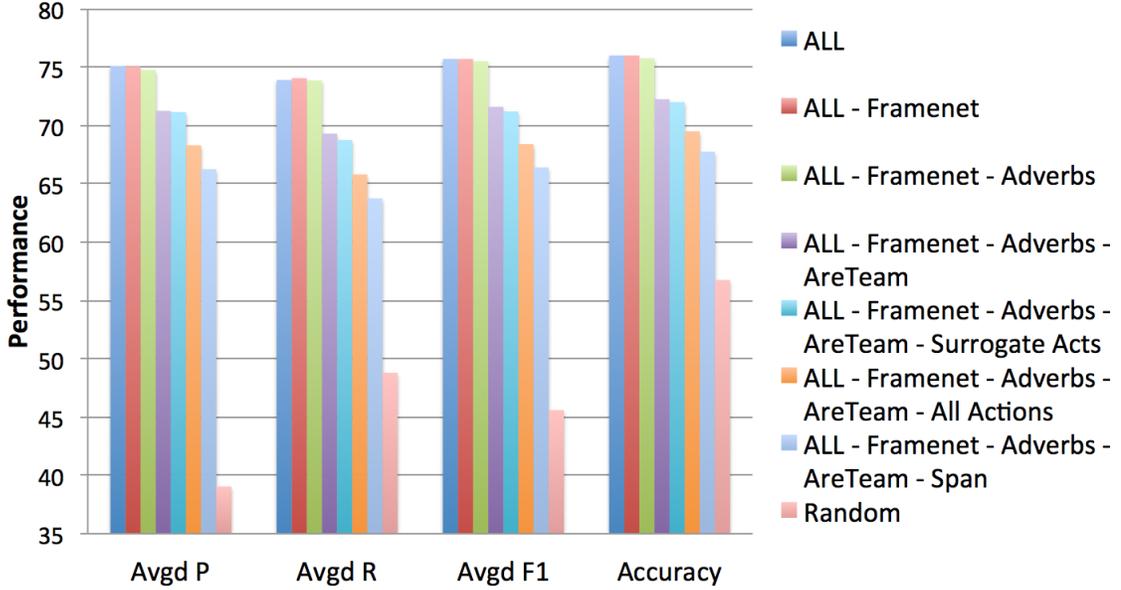


Figure 3.4: Ablation study for text feature families

3.6.1 Feature ablation

Figure 3.4 shows the 5-fold cross-validation performance of major feature families of text features on the training set. We note that Frame-semantic features and adverbial connotations of character actions do not add significantly to model performance. This is perhaps because both these families of features were sparse. Additionally, frame-semantic parses were observed to have frequent errors in frame evocation, and frame element assignment. On the other hand, we observe that joint participation in actions (as agent or patient) is a strong indicator of cooperative relationships. In particular, incorporating these (*‘Are Team’*) features was seen to improve both precision and recall for the cooperative class; while not degrading recall for the non-cooperative class. Further, while ignoring sentiment and connotation features for surrogate action features results in marginal degradation in

performance; the most significant features are seen to be sentiment and connotation features for actions where both characters occur in agent/patient roles (*‘Acts Together’* features); and overall sentiment characterizations for words and phrases in spans of text between character mentions (span based *‘Lexical sentiment’* features).

3.6.2 Structured vs. unstructured models

We now analyze the performance of our proposed models; and evaluate the significance of adding structural features to our models. In our experiments, we found the structured models to consistently outperform text-based models. We tune values of hyper-parameters, i.e. number of training epochs for the structured perceptron (10), through cross validation on training data. Table 3.1 compares the performance of the models on our held-out test set of 27 movie summaries (comprising about 20% of the all annotated character relations) using accuracy and averaged precision, recall and F-measure over the two classes. For the structured models, reported results are averages over 10 initializations.

We observe that the structured perceptron model (Structured Model) outperforms the text-only model trained with logistic regression (Unstructured Model). These results are consistent with our cross-validation findings, and demonstrate that structural features can substantially improve inference of character relations.

Let us consider the affinities for structural features learnt by the model. Over 10 runs of the structured model, the average weights were: $w_{\text{clique}} = -2.79$, $w_{\text{lovetriangle}} = -0.84$, $w_{\text{commonenemy}} = 10.26$ and $w_{\text{truel}} = -5.49$. From the perspective of structural

Model	Avgd P	Avgd R	Avgd F	Acc
Unstructured (Logistic Regression)	70.2	69.7	69.9	70.1
Structured	79.4	79.3	79.3	79.2

Table 3.1: Test set results for relation classification.

balance theory (described in Sec. 3.4), the social configurations `lovetriangle` and `true1` are inherently unstable, whereas the others are stable. Hence one might have expected weights to follow this intuition. However, the learned weights make a significant departure from this conjecture. Specifically, the learnt affinity for the configuration `lovetriangle` seems higher than expected. This is unsurprising, however, if we consider the domain of the data (movies), where it might be a common plot element. We also note that the ‘friend of a friend is a friend’ maxim is not supported by the feature weights (even though it is a stable configuration), and hence a model based on this as a hard transitive constraint could be expected to perform poorly.

3.6.3 Qualitative Discussion

We observe that relation characterizations for character pairs are reasonable for most texts in the test set. Figure 3.5 shows labels inferred by the model for two well-known movies in the test set. We can see that the model correctly identifies all inter-character relationships.

Error analysis revealed that mismatched coreference labelings are the most

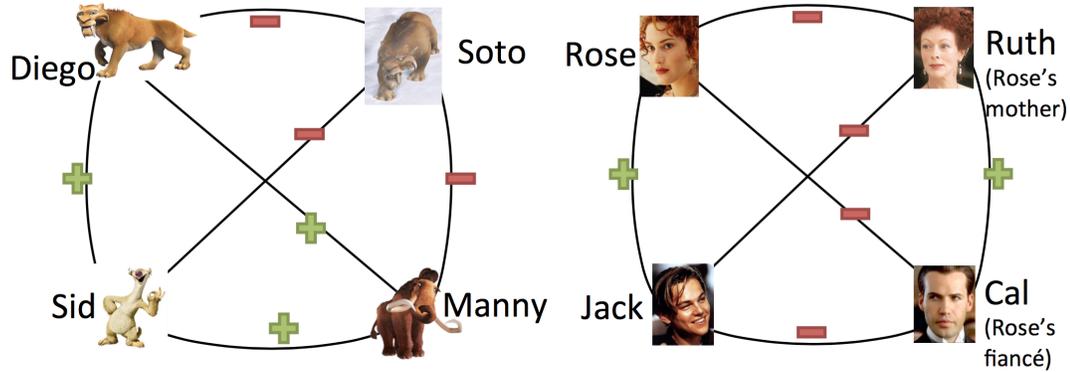


Figure 3.5: Inferred relationships for movies ‘Titanic’ (1997) and ‘Ice Age’ (2002)

common source of errors for the model. Secondly, in some cases, the text-based features mistakenly identify negative sentiments due to our coarse model of sentiment used in the feature-extraction process. For example, Figure 3.6 shows some examples of the relationships inferred for the movie ‘Smokin’ Aces 2: Assassins’ Ball’. The movie contained several characters, which yielded a large network. In the figure, we show the two relevant subgraphs of the inferred network for clarity. The model identifies a non-cooperative relationship between Agents Baker and Redstone, which is incorrect. This happens because in the following sentence, the linguistic features pay attention to the negative connotation of ‘drag’ without understanding that Baker drags Redstone to keep him safe: ‘*Baker drags the wounded Redstone to the “spider trap” ... used to safeguard people*’. In the same figure, the model incorrectly identifies a cooperative relationship between Lester and McTeague. This happens because in the following sentence from the movie’s summary, the lexical features include a few words with strong positive connotations like *father*, *children* and *family* biasing the model towards identifying a cooperative relationship between them: ‘*These include Ariella Martinez, ... Finbar McTeague who is also known as ‘The*

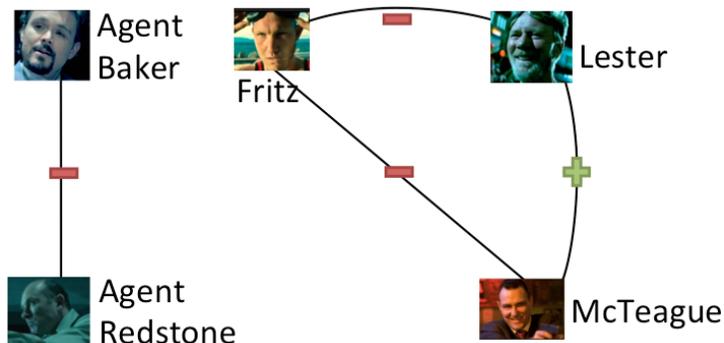


Figure 3.6: Two relevant subsets of the inferred relationships for movie: ‘Smokin’ Aces 2: Assassins’ Ball’ (2010). The model incorrectly learns a cooperative relationship between Lester and McTeague, and a non-cooperative relationship between Agents Baker and Redstone.

Surgeon’, who brutally tortures his victims; the Southern Tremor family consisting of father Fritz and children; Lester...’.

3.7 Conclusion

In this chapter we addressed the problem of identifying inter-character relationships in movie summaries. While making a simplifying assumption that relationships can be of only two types, we demonstrated that the task of relationship modeling can benefit from including linguistic cues as well as structural cues. Specifically, we empirically demonstrated that this task benefits from viewing inter-character relations in context of their relationships with other characters in the narrative. Our experiments on a dataset of movie summaries from Wikipedia demonstrate how a structured model that incorporates this context performs better than a text based

model. While our test-bed was movie summaries, in the future, the framework could potentially apply to other domains of texts with social narratives, such as news stories. The framework could also be extended to finer grained relation categorizations beyond coarse sentiment connotations, as well as comparing narratives based on their interpersonal relationships. Conceptually, a natural extension would be to use predictions about character relations to infer subtle character attributes such as agenda, intentions and goals.

Chapter 4: Modeling Evolving Relationships Between Literary Characters

Work described in this chapter was published in AAAI 2016 [28]

Like the previous chapter, in this chapter we model polarities of inter-character relationships, i.e. we again assume that relationships are of two types: cooperative and non-cooperative. However, we now assume that inter-personal relationships evolve with the progress of the narrative, and present the problem of modeling relationships as a structured prediction task. We demonstrate that for this task, it is important to employ structured models that are capable of modeling relationship between characters at any point in the narrative in *context of the history of their relationship*.

4.1 Problem Motivation and Definition

Existing character-centric approaches to understanding texts [105, 11, 12], model each character as assuming a static narrative role, and these roles define the relationships between characters and also govern their actions through out the narrative. While such a simplified assumption provides a good general overview of the narrative, it is not sufficient to explain *all* events in the narrative. We

believe that in most narratives, relationships between characters are not static but *evolve* as the novel progresses. For example, consider the relationship between *Tom* and *Becky* depicted in Figure 4.1 which shows an excerpt from the summary of *The Adventures of Tom Sawyer* by *Mark Twain*¹. For most of the narrative, the characters are participants in a romantic relationship, which explains most, but not all, of their mutual behavior. We can observe that their relationship changed during the narrative. They presumably start as lovers (sentence S1 in the Figure), which is indicated by (and explains) becoming engaged. The relationship sours when Tom reveals his previous love interest (S2 and S3). However, later in the narrative they reconcile (S4 and S5). A model that assumes a fixed romantic relationship between characters would fail to explain their behaviors during the phase when their relationship was under stress.

Therefore, in this chapter we assume that the relationship between characters evolves with the progress of the novel and address the task of **learning relationship sequences**. For instance in Figure 4.1, the relationship between Tom and Becky can be represented by the sequence ⟨cooperative, non-cooperative, cooperative⟩.

¹SparkNotes Editors. SparkNote on The Adventures of Tom Sawyer. SparkNotes LLC. 2003.
<http://www.sparknotes.com/lit/tomsawyer/>

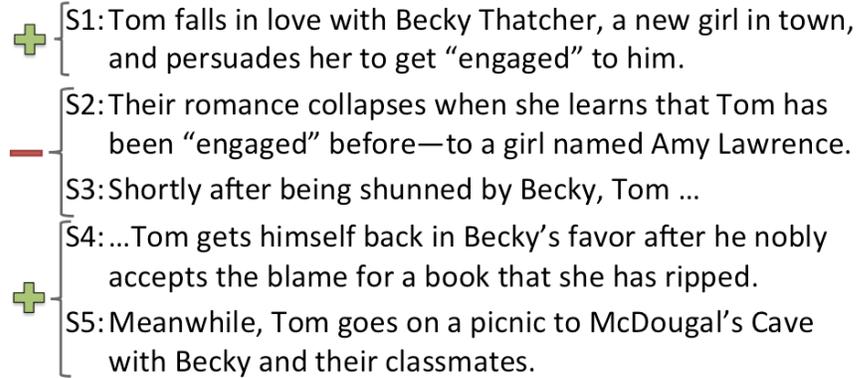


Figure 4.1: Sample sentences from a narrative depicting evolving relationship between characters: Tom and Becky. The relationship changes from cooperative (+) to non-cooperative (-) and then back to cooperative (+). ‘...’ represents text omitted for brevity.

4.2 Role of Context

To address this problem, we model relationship as a structured variable – a sequence of variables each denoting relation state at a point in the narrative. The narrative fragment of interest for us is represented by the set of sentences in which the two characters of interest appeared together, arranged in the order of occurrence in the narrative. More formally, given the narrative text in form of a sequence of sentences, $\mathbf{x} = \langle x_1, x_2, \dots, x_l \rangle$, we address the problem of segmenting it into non-overlapping and semantically meaningful segments that represent continuities in relationship status. Each segment is labeled with a single relationship status $r_j \in \{-1, +1\}$ hence yielding a relationship sequence $\mathbf{r} = \langle r_1, r_2, \dots, r_k \rangle k \leq l$. Like Chapter 3, we take a coarse-grained view, and model relation states as binary variables (indicating *cooperative/non-cooperative* relation).

Our task requires us to assign a relationship state to each sentence in the narrative. However, instead of viewing these sentences independently, it is natural to view them as a sequence and solve the problem using a structured model. We use a Markov model, which is capable of viewing the historical context while deciding the relationship state for a sentence.

4.3 Model

Our approach utilizes a second order Markovian latent variable model for segmentation that is embedded in semi-supervised framework to utilize varying levels of labeling in the data. We now describe our segmentation model and the semi-supervised framework in detail.

4.3.1 Segmentation Model

This model forms the core of our framework. It assumes that each sentence in the sequence is associated with a state that represents its relationship status. While making this assignment, it analyzes the content of individual sentences using a rich feature set and simultaneously models the flow of information between the states by treating the prediction task as a structured problem. Specifically, our approach employs a second order Markovian segmentation model, which assumes that each sentence in the sequence is associated with a state that represents its relationship status.

It collectively maximizes the following linear scores for individual sequences:

$$\text{score} = \sum_i \mathbf{w}\phi(\mathbf{x}, y_i, y_{i-1}, y_{i-2}) \quad (4.1)$$

where x is the input sequence and y_i denotes the latent state assignment of its i^{th} sentence to a relationship segment. Individual y_i s collectively yield the relationship sequence, \mathbf{r} (by collapsing consecutive occurrences of identical states). ϕ represents features at the i^{th} sentence that depend on the current state, y_i , and the previous two states, y_{i-1} and y_{i-2} , and \mathbf{w} represents their weights. The second order Markov assumption of our features ensures continuity and coherence of behavior of the two characters within individual relationship segments.

The Markovian segmentation model proposed above is trained using an averaged structured perceptron [31]. For inference, it uses a Viterbi based dynamic programming algorithm. The extension of Viterbi to incorporate second order constraints is straightforward. We replace the reference to a state (in the state space $|Y|$) by a reference to a state pair (in the two fold product space $|Y| \times |Y|$). Note that this precludes certain transitions while computing the Viterbi matrix, viz.: if the state pair at any point in narrative, t , is of the form (s_i, s_j) , then the set of state pair candidates at $t + 1$ only consists of pairs of the form (s_j, s_k) . Incorporating these constraints, we compute the Viterbi matrix and obtain the highest scoring state sequence by backtracking as usual [96].

4.3.2 Semi-supervised Framework

Given the nature of the task, we acknowledge that obtaining a huge dataset of labeled sequence can be expensive. However, it might be more convenient to obtain partially labeled data especially in cases in which only a subset of the sentences of a sequence have an obvious relationship state membership. We, therefore, propose to embed the above mentioned segmentation model in a semi-supervised framework, which assumes that the training dataset consists of two types of labeled sequences: fully labeled, in which the complete state sequence is observed $y_i \forall i \in \{1 \dots l\}$ and partially labeled, in which some of the sentences of the sequence are annotated with y_i such that $i \subset \{1 \dots l\}$.

Algorithm 2 Training the semi-supervised framework

- 1: **Input:** Fully F and partially P labeled sequences; and T : number of iterations
 - 2: **Output:** Weights \mathbf{w}
 - 3: **Initialization:** Initialize \mathbf{w} randomly
 - 4: **for** $t : 1$ to T **do**
 - 5: $\hat{\mathbf{y}}_j = \arg \max_{\mathbf{y}_j} [\mathbf{w}_t \cdot \phi(\mathbf{x}, \mathbf{y})_j] \forall j \in P$ such that $\hat{\mathbf{y}}_j$ agrees with the partial annotated states (ground truth).
 - 6: $\mathbf{w}_{t+1} = \text{StructuredPerceptron}(\{(\mathbf{x}, \hat{\mathbf{y}})_j\} \in \{P, F\})$
 - 7: **end for**
-

This framework uses a two step algorithm (Algorithm 2) to iteratively refine feature weights, \mathbf{w} , of the segmentation model. In the first step, it uses existing weights, \mathbf{w}_t , to assign state sequences to the partially labeled instances. For state

assignment we use a constrained version of the Viterbi algorithm that obtains the best possible state sequence that agrees with the partial ground truth. In other words, for the annotated sentences of a partially annotated sequence, it precludes all state assignments except the given ground truth, but segments the rest of the sequence optimally under these constraints. In the second step, we train the averaged structured perceptron based segmentation model, using the ground truth and the state assignments obtained in the previous step, to obtain the refined weights, \mathbf{w}_{t+1} .

4.4 Features

This section describes the features used by our model.

Pre-processing

We first pre-processed the text of various novel summaries to obtain part-of-speech tags and dependency parses, identify major characters and perform character names clustering (assemble ‘Tom’, ‘Tom Sawyer’, etc.) using the Book-nlp pipeline [12]. However, the pipeline, designed for long text documents involving multiple characters, was slightly conservative while resolving co-references. We augmented its output using coreferences obtained from the Stanford Core NLP system [73]. We then obtained a frame-semantic parse of the text using Semafor [33]. We also obtained connotation [46], sentiment [69] and prior-polarity [110] of words when needed during feature extraction.

Finally, given two characters and a sequence of pre-processed sentences in

which the two appeared together, we extracted the following features for individual sentences.

4.4.1 Content features

These features help the model in characterizing the textual content of the sentences. They are based on the following general template which depends on the sentence, x_j , and its state, y_j : $\phi(x_j, y_j) = \alpha$ if the current state is y_j ; 0 otherwise where, $\alpha \in F1$ to $F33$, and $F1$ to $F33$ are defined below. Figure 5.2 shows examples of these feature categories for a sample sentence.

1. Actions based: These features are motivated by Vladimir Propp’s Structuralist narrative theory [89] based insight that characters have a ‘sphere of actions’. We model the actions affecting the two characters by identifying all verbs in the sentence, their agents (using ‘nsubj’ and ‘agent’ dependency relations) and their patients (using ‘dobj’ and ‘nsubjpass’ relations). This information was extended using verbs conjunct to each other using ‘conj’. We also used the ‘neg’ relation to determine the negation status of each verb. We then extracted the following features:

- *Are Team [F1]:* This binary feature models whether the two characters acted as a team by indicating if the two characters were agents (or patients) of a verb together.
- *Acts Together [F2-F7]:* These features explicitly model the behavior of the two characters towards each other using verbs for which one of the characters was the agent and the other was patient. These six numeric features look at

positive/negative connotation, sentiment and prior-polarity of the verbs while considering their negation statuses (see Pre-processing for details).

- *Surrogate Acts Together [F8-F13]*: These are high-recall features that analyze actions for which a character was an implicit/subtle agent or patient. For example, Tom is not the direct patient of *shunned* in S3 in Figure 4.1. We define a set of six surrogate features that, like before, consider connotations, sentiments and prior-polarities of verbs (considering negation). However, only those verbs are considered which have one of the characters as either the agent or the patient, and occur in sentences that did not contain any other character apart from the two of interest.

2. Adverb based: These features model narrator’s bias in describing characters’ actions by analyzing the adverbs modifying the verbs identified in ‘Action based’ features (using ‘advmod’ dependency relations). For example, in S4 in Figure 4.1 the fact that Tom *nobly* accepts the blame provides an evidence of a positive relationship.

- *Adverbs Together [F14-F19]* and *Surrogate Adverbs Together [F20-F25]*: Six numeric features measuring polarity of adverbs modifying the verbs considered in ‘Acts Together’ and ‘Surrogate Acts Together’ respectively.

3. Lexical [F26-27]: These bag-of-words style features analyze the connotations of all words (excluding stop-words) occurring between pairs of mentions of the two characters in the sentence. E.g. in S5 in Figure 4.1 the words occurring between the

Type	Frame[Frame-elements]
Negative	<i>killling</i> [<i>killer, victim</i>]
	<i>attack</i> [<i>assailant, victim</i>]
Positive	<i>forgiveness</i> [<i>judge, evaluatee</i>]
	<i>supporting</i> [<i>supporter, supported</i>]
Ambiguous	<i>cause bodily experience</i> [<i>agent, experiencer</i>]
	<i>friendly or hostile</i> [<i>side_1, side_2, sides</i>]
Relationship	<i>kinship</i> [<i>alter, ego, relatives</i>]
	<i>subordinates_superiors</i> [<i>superior, subordinate</i>]

Table 4.1: Frame samples used by ‘Semantic Parse’ features.

pair of mentions the characters, *Tom* and *Becky*, are “goes on a picnic to McDougal’s cave with” (stopwords included for readability).

4. Semantic Parse based: These features incorporate information from a Framenet-style semantic parse of the sentence. To design these features, we manually compiled lists of positive (or negative) *frames* (and relevant *frame-elements*) depending on whether they are indicative of positive (or negative) relationship between participants (identified in the corresponding frame-elements). We also compiled a list of ambiguous frames like ‘cause_bodily_experience’ for which the connotation was determined on-the-fly depending on the connotation of lexical unit at which that frame

was evoked. Lastly, we had a list of ‘Relationship’ frames that indicated familial or professional relationships. Table 4.1 shows examples of various frame-types and relevant frame-elements. The complete lists are shown in Appendix A ². Based on these lists, we extracted the following two types of features:

- *Frames Evoked [F28-F30]*: Three numeric features counting number of positive, negative and ‘relationship’ frames evoked such that at least one of the characters belonged to the relevant frame-element.
- *Surrogate Frames Evoked [F31-F33]*: Three features counting number of positive, negative and ‘relationship’ frames evoked.

4.4.2 Transition features

While content features assist the model in analyzing the text of individual sentences, these features enable it to remember relationship histories, thus discouraging it from changing relationship states too frequently within a sequence.

- $\phi(y_j, y_{j-1}, y_{j-2}) = 1$ if current state is y_j and the previous two states were y_{j-1}, y_{j-2} ; 0 otherwise
- $\phi(y_j, y_{j-1}) = 1$ if current state is y_j and the previous state was y_{j-1} ; 0 otherwise
- $\phi(y_0) = 1$ if state of the first sentence in the sequence is y_0 ; 0 otherwise

²We do not claim these lists to be exhaustive

4.5 Data

We have used the following two datasets in our experiments both of which are based on novel summaries.

SparkNotes: This dataset consists of a collection of summaries (‘Plot Overviews’) of 300 English novels extracted from the ‘Literature Study Guides’ section of SparkNotes.³ We pre-processed these summaries as described in Section 4.4. Thereafter, we considered all pairs of characters that appeared together in at least five sentences in the respective summaries and arranged these sentences in order of appearance in the original summary. We refer to these sequences of sentences as simply a sequence. We considered the threshold of 5 sentences to harvest sequences long enough to manifest ‘evolving relationships, but also sufficiently many to allow learning. This yielded a collection of 634 sequences consisting of a total of 5542 sentences.

For our experiments, we obtained annotations for a set of 100 sequences. Annotators were asked to read the complete summary of a novel and then annotate character-pair sequences associated with it. Sentences in a character-pair sequence were labeled as cooperative (when the two characters supported each others actions/intentions or liked each other) or non-cooperative (otherwise). Annotators had access to the complete summary throughout the annotation process. They were required to fully annotate the sequences whenever possible and partially annotate them when they couldn’t decide a relationship status for some of the sentences in the sequence. It was permissible to annotate a sequence with all cooperative or all

³<http://www.sparknotes.com/lit/>

non-cooperative states. In fact, that happened in 70% of the sequences. The dataset thus obtained contained 50 fully annotated sequences (402 sentences) and 50 partially annotated sequences (containing 390 sentences, of which 201 were annotated). Of all annotated sentences, 472 were labeled with a cooperative state.

AMT: We considered another dataset only for evaluating our model. This dataset [74] was collected independently by another set of authors using Amazon Mechanical Turk.⁴ The annotators were shown novel-summaries and a list of characters appearing in the novel. Given a pair of characters, they annotated if the relationship between them changed during the novel (binary annotations). They were also asked other questions, such as the overall nature of their relationship, etc., which were not relevant for our problem. There was some overlap between the novel summaries used by the two datasets described here, due to which, 62 pairs of characters from this dataset could be found in the SparkNotes dataset. This dataset of 62 pairs can be viewed as providing additional binary ground truth information and was used for evaluation only after training on SparkNotes data. The relationship was annotated as ‘changed’ (positive class) for 20% of these pairs.

4.6 Evaluation

We now describe our baselines and evaluation results.

Baselines and Evaluation Measures

Our primary baseline is an unstructured model that trains flat classifiers using

⁴<https://www.mturk.com/>

Model Type	Model Name	P	R	F	ED
Unstructured	Decision Tree	67.89	75.60	69.86	0.98
	Logistic Regression	72.33	77.51	72.94	1.06
Structured	Order 1 Model	74.32	73.16	73.70	0.9
	Order 2 Model	76.58	76.97	76.76	0.66

Table 4.2: Cross validation performances on SparkNotes data. The second order model based framework outperforms the one that uses a first order model and the unstructured baselines.

the same content features as used by our framework but treats individual sentences of the sequences independently. We compare our model with this baseline to validate that relationship sequence prediction is a structured problem, which benefits from remembering intra-novel history of relationship between characters.

We also compare our framework, which employs a second order Markovian segmentation model, with an identical framework, with a first order Markovian segmentation model. This baseline is included to understand the importance of remembering a longer history of relationship between characters. Also, since a higher order model can look further back, it will discourage frequent changes in relationship status within the sequence more strongly.

For evaluation, we use two different measures comparing model-performances at both sentence and sequence levels. Our first measure accesses the goodness of

the binary relationship state assignments for every sentence in the sequence using averaged Precisions (P), Recalls (R) and F1-measures (F) of the two states. The second evaluation measure, mimics a more practical scenario by evaluating from the perspective of the predicted relationship sequence, \mathbf{r} , instead of looking at individual sentences of the sequence. It compares the ‘proximity’ of the predicted sequence to the ground truth sequence using Edit Distance and reports its mean value (ED) over all test sequences. A better model will be expected to have a smaller value for this measure.

4.6.1 Evaluation on the SparkNotes dataset

Table 4.2 compares 10-fold cross validation performances of our second order Semi-supervised Framework (Order 2 Model) with its first order counterpart (Order 1 Model) and two unstructured baselines: Decision Tree and Logistic Regression. Since the performance of the semi-supervised frameworks depends on random initialization of the weights, we report mean values over 50 random restarts in the table. The number of relationship states, $|Y|$, was set to be 2 to correspond to the gold standard annotations. The table shows that the framework with the first order Markov model yields slightly better performance (higher averaged F-measure and lower mean Edit Distance) than the unstructured models. This hints at a need for modeling the information flow between sentences of the sequences. The further performance improvement with the second order model emphasizes this hypothesis and also demonstrates the benefit of remembering longer history of characters while

Model Type	Model Name	P	R	F
Unstructured	Decision Tree	68.18	43.55	48.54
	Logistic Regression	71.93	46.77	51.48
Structured	Order 1 Model	72.36	50.64	52.52
	Order 2 Model	71.62	56.45	60.76

Table 4.3: Performance comparison on the AMT dataset. The second order model based framework outperforms the one that uses a first order model and the unstructured models.

making relationship judgments.

4.6.2 Evaluation on the AMT dataset

Table 4.3 compares performances of the various models on the AMT dataset using averaged Precision, Recall and F measures on the binary classification task of change prediction. The problem setting, input sequences format and the training procedure for these models are same as above. However, the models produce structured output (relationship sequences) that need to be converted to the binary output of change prediction task. We do this simply by predicting the positive class (change occurred) if the outputted relationship sequence contained at least one change. We can see that while the performance of the framework using the first order model is similar to that of the baseline Logistic Regression, the second order model shows a

considerable improvement in performance. A closer look at the F measures of the two classes revealed that while the performance on the positive class was similar for all the models (except Decision Tree which was considerably lower), the performance on the negative class (no change) was much higher for the structured models (56.0 for Logistic Regression and 57.4 and 67.8 for the First and Second order models respectively). This might have happened because the unstructured model looks at independent sentences and cannot incorporate historical evidence so it is least conservative in predicting a change, which might have resulted in low recall. The structured models, on the other hand, look at previous states and can better learn to make coherent state predictions.

4.6.3 Feature Ablation

Figure 4.2 plots 10-fold cross validation F-measure to study the predictive importance of various feature-families using Logistic Regression on the SparkNotes data. The black bar (labeled ‘All’) represents the performance using the complete feature set and the rest of the bars represent the scenario when various feature-families are incrementally omitted. We can note that the ‘Are Team’ and ‘Acts Together’ features seem to be very informative as removing them degrades the performance remarkably. On the other hand, the ‘Adverbs Together’ feature seems to be least informative, possibly because it was sparsely populated in our dataset. Nevertheless we can conclude that, in general, removing any feature-family degrades model’s performance indicating their predictive utility.

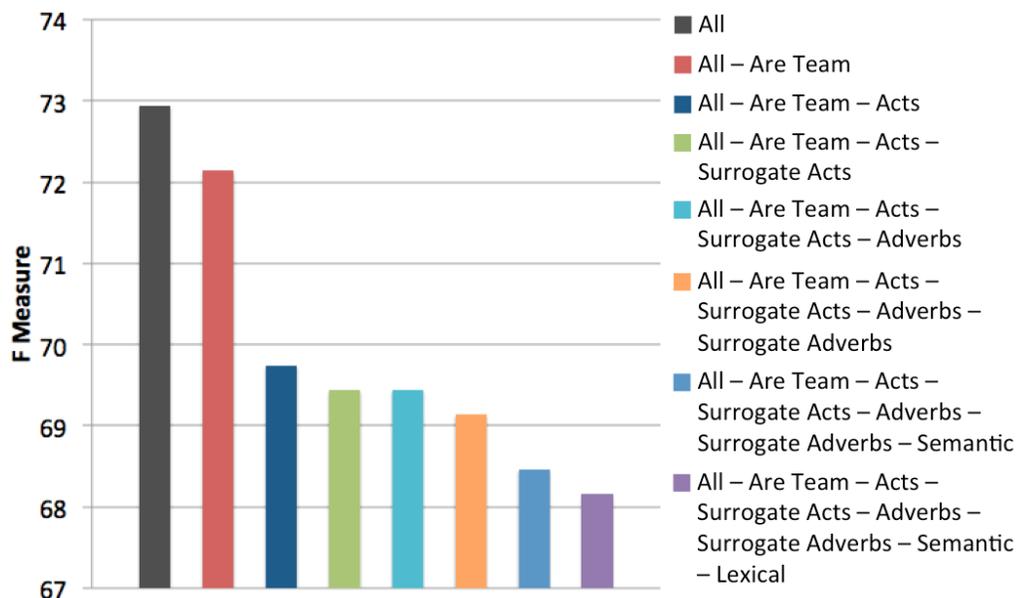


Figure 4.2: Ablation results on SparkNotes dataset. All represents performance with full feature-set and rest of the bars indicate performance with incrementally removing various feature-families.

4.6.4 Case Study

We now use our framework to gain additional insights into our data. To this end, we use the framework to make predictions about various character pairs from summaries of the seven *Harry Potter* novels by *J. K. Rowling*. As before, only those pairs were considered for which the two characters appeared together in at least five sentences and none of these pairs were manually annotated. We then clustered the various pairs according to the Edit-distance based similarity between their relationship sequences. Table 4.4 shows sample pairs for three such clusters. Note that some of the character pairs appear more than once because several characters are shared across the seven books. While performing this clustering no information other than

Cluster 1	Cluster 2
Harry, Dobby [<i>Chamber of Secrets</i>]	Ron, Hermione [<i>Deathly Hallows</i>]
Harry, Dumbledore [<i>Half-Blood Prince</i>]	Harry, Ron [<i>Deathly Hallows</i>]
Hagrid, Harry [<i>Prisoner of Azkaban</i>]	Sirius, Ron [<i>Prisoner of Azkaban</i>]
Ron, Harry [<i>Order of the Phoenix</i>]	Sirius, Harry [<i>Prisoner of Azkaban</i>]
Harry, Hermione [<i>Deathly Hallows</i>]	Hermione, Sirius [<i>Prisoner of Azkaban</i>]

Cluster 3
Harry, Snape [<i>Prisoner of Azkaban</i>]
Draco, Harry [<i>Half-Blood Prince</i>]
Voldemort, Dumbledore [<i>Half-Blood Prince</i>]
Voldemort, Dumbledore [<i>Deathly Hallows</i>]
Harry, Voldemort [<i>Half-Blood Prince</i>]

Table 4.4: Sample character pairs (and book titles) from the clusters obtained from the summaries of the *Harry Potter* series. Pairs in clusters 1 and 3 had a cooperative and non-cooperative relationship throughout the novel (respectively). Cluster 2 contains pairs for which the relationship became non-cooperative once in the novel but then finally became cooperative.

the relationship sequence itself (such as character identities) was used. Also, the pairs are unordered.

Cluster 1 consists of pairs whose relationship remained mostly cooperative throughout the novel. This includes relationships of Harry with his friends Ron and Hermione, benefactors Dumbledore, Hagrid and Dobby. Cluster 2 consists of pairs like $\langle \text{Ron, Hermione} \rangle$ and $\langle \text{Harry, Ron} \rangle$ from ‘Harry Potter and the Deathly Hallows’. Their relationships were similar in the sense that the three characters started out as friends and go on a quest. However, because of their initial failures Ron abandons the other two, but reunites with them towards the later part of the novel. Hence Ron’s relationship with each of the other two was both cooperative and non-cooperative during the course of the novel.

Similarly, Cluster 3 consists of character pairs, which had a non-cooperative relationship for most of the novel. However, a more careful examination revealed that in some of these pairs the model assigned a cooperative status to a few sentences in the beginning. For example, for Voldemort and Dumbledore in ‘Harry Potter and the Half-Blood Prince’, when Dumbledore (along with Harry) tries to learn more about Voldemort. A human reader, who has access to the complete text of the summary as well as context from previous novels, can understand that they learn about him to fight him better and hence the reader can infer that the relationship is non-cooperative. However, in our setting, we ignore all sentences except those in which the two characters appear together, hence depriving the model of the valuable information present in between the input sentences. We believe that a more sophisticated approach that understands the complete narrative text (instead of

sporadic sentences about the two characters of interest) will make better inferences.

4.7 Conclusion

In this chapter we modeled the dynamic or evolving nature of inter-personal relationships. We analyzed summaries of novels to extract relationship trajectories that describe how the relationship evolved. We addressed this as a sequence prediction task, and presented a semi-supervised framework that uses a structured segmentation model. Our model makes second-order Markov assumption to remember the ‘history’ of characters and analyzes textual contents of summaries using rich semantic features that incorporate world knowledge. We demonstrated the utility of including the long-range historical context in addition to textual cues by comparing our model with an unstructured model that treats individual sentences independently and also with a lower order model that remembers shorter history.

Future work could experiment with higher order and semi-Markov models. Also, this work treats different character pairs from the same novel independently and does not attempt to understand the complete text of the narrative. Future work could explore a more sophisticated model that exploits intra-novel dynamics while predicting relationships.

An important contribution this work is identifying the evolutionary nature of relationships. Further work in this direction could be used to answer questions like “What kind of novels have happy endings?”, “Are there general narrative templates of relationship evolution between the protagonist and his/her lover?”, etc.

Chapter 5: Unsupervised Modeling of Evolving Inter-character Relationships

Work described in this chapter is submitted for review.

The previous two chapters make a limiting assumption on types of relationships. They assume that relationships are only of two types: cooperative or non-cooperative. In this chapter we relax this condition and assume that relationships can be of multiple types. We introduce data-driven models that can learn these types automatically. Like Chapter 4, we model evolving inter-character relationships with a focus on novel summaries. While modeling the dynamic nature of these relationships we incorporate *historical context* by solving this as a structured prediction problem. At the same time, while taking a decision about nature of relationship between the characters of interest at various points within a narrative, we incorporate *global context* about the overall nature of relationships between them.

5.1 Problem Motivation and Definition

Most existing methods to analyze inter-personal relationships in narratives, including those presented in the previous chapters in this dissertation, are inadequate in one or more of the following three ways: (i) Firstly, they assume a static relation-

P1: Harry overhears Ron's parents discussing that Sirius Black is after Harry. Harry overhears Cornelius Fudge discussing Sirius Black's responsibility for Harry's parents' deaths...Harry, Ron, and Hermione manage to disarm Black, and before Harry can kill Black...

P2: Lupin and Black then explain that the real murderer of Harry's parents is not Black...Sirius Black responds by turning into the large black dog in order to protect Harry...Harry and Hermione fly to the tower where Black is imprisoned, and they rescue Black...

Figure 5.1: Excerpts from the summary of the novel ‘Harry Potter and the Prisoner of Azkaban’ depicting non-stationary nature of relationship between Harry Potter and Sirius Black. Their relationship changes from hostility to cooperative alliance. Relationship between characters within a narrative, represented by a single variable (ii) Secondly, they characterize relationships by coarse sentiment polarities, e.g., friendly vs. adversarial, which conflates distinct semantic categories and ignores other aspects such as power (iii) Thirdly, they require expensive and resource-intensive manual annotation. The work presented in this chapter attempts to address these limitations.

The shortcomings of such methods in understanding a narrative are apparent. Figure 5.1 shows excerpts from the summary of the novel ‘Harry Potter and the Prisoner of Azkaban’ focusing on the relationship between ‘Harry Potter’ and ‘Sirius Black’. We can see that in the text labeled P1 in the figure, Harry believes that Sirius is responsible for his parents’ deaths and so tries to kill him. Later (P2) he learns that Sirius is innocent and they both protect each other, and hence their relationship changes from that of enmity on Harry’s part to cooperative alliance. Clearly, describing their relationship by a single category will not explain all of their

actions. For instance, if we assume that they are allies then we cannot explain why Harry tried to kill Sirius initially. Thus, like Chapter 4, there is a need to model inter-character relationships as dynamic variables that evolve with the progress of the narrative. Also, real-world relationships are nuanced with facets such as family, romance, formality/informality, supervision/subordination, etc. and an ideal model would allow flexibility to express these.

In this chapter we present a framework for **unsupervised modeling of inter-character relationships** from unstructured text with a focus on novel summaries. The input to our models is the text of a narrative summary and a pair of characters appearing in the narrative. We address this problem by extracting a vector representation for each of the sentences in which the two characters of interest appear together. The vector representation of the individual sentences attempts to capture cues about the relationship between the two characters in that sentence. These sentences (and their vector representations) are arranged in their order of appearance in the narrative. Finally, each vector is associated with a latent variable, representing the relationship state between the two characters at that point in the narrative.

5.2 Role of Context

In this chapter we present three models, which address this problem as an unsupervised structured prediction task. Given a narrative (presented as a sequence of sentences) and a pair of characters appearing in the narrative, our models learn a sequence of latent variables representing their (dynamic) relationship over the course

of the narrative. Our models make Markovian assumption to capture the ‘flow of information’ or the historical context between individual sentences. While deciding the relationship state at any point in the narrative, the Markovian assumption enables our models to look at not just the contents of the current sentence, but also the previous few relationship states.

However, it may be argued that inferring relationship state requires a longer context, or a *global perspective*. In other words, judging the relationship state at any juncture needs consideration of not only the current sentence, but also the *overall* nature of the relationship between the characters. E.g., in the novel titled ‘Harry Potter and the Half Blood Prince’ by J. K. Rowling, Harry, the protagonist, investigates into the history of the villain, Voldemort, and ‘learns more’ about him. While ‘learning more’ by itself does not give us much insight into their relationship, knowing that they are enemies in general, tells us that Harry is learning more about Voldemort to fight him better and that the relationship at this point is that of animosity. Therefore, we introduce a third model, Globally Aware GHMM, which incorporates short historical context via Markovian assumptions as well as longer global context using two distinct local and global components.

5.3 Feature-vectors Extraction

Given a narrative text and two characters appearing in it, our approach aims to represent their relationship as a sequence of latent variables. The sentences in the narrative that are of interest to us are those in which the two characters appear

together. These sentences have a natural order of their appearance in the narrative, yielding a sequence, $\mathbf{s} = \langle s_1, s_2 \dots s_T \rangle$. We represent this sequence of sentences as a sequence of D -dimensional feature-vectors $\mathbf{f} = \langle \vec{f}_1, \vec{f}_2 \dots \vec{f}_T \rangle$ such that $\vec{f}_t \in \mathbb{R}^D$. We provide this sequence of feature-vectors representing the interactions between the two characters as input to our models. Our models assign each feature-vector, \vec{f}_t , to a discrete latent relationship state, $r_t \in \{1, 2, \dots R\}$, thus outputting a sequence of latent relationship states, $\mathbf{r} = \langle r_1, r_2 \dots r_T \rangle$.

This section describes the process of extracting a sequence of feature-vectors $\mathbf{f} = \langle \vec{f}_1, \vec{f}_2 \dots \vec{f}_T \rangle$ from a sequence of sentences $\mathbf{s} = \langle s_1, s_2 \dots s_T \rangle$.

We pre-processed the narratives using the BookNLP pipeline [12], Stanford’s CoreNLP system [73], and frame-semantic parser [33] as described in Section 4.4. We then extracted the following sets of words from each sentence. These sets are motivated by the feature-extraction process discussed in Section 4.4 albeit without the various connotation lexicons. However, we briefly summarize these sets here again for the sake of clarity and encourage the reader to refer to the previous chapter for more details.

Actions: We represent the relationship between characters using their actions, especially to each other. We capture the notion of actions by extracting the set of verbs, which had one character as an agent, and the other as a patient. For example in the sample sentence in Figure 5.2, the action word is ‘asked’.

Surrogate Actions: The above set of verbs can be affected by limitations of the NLP pipeline. For example, in the sentence in Figure 5.2, ‘Jim’ is the implicit agent of ‘confronting’ ‘Maria’. To include such cases, we extract another set of verbs, which

Sentence: After confronting Maria, Jim furiously asked her to end the friendship.
Actions: asked
Surrogate Actions: confronting
Lexical: furiously asked
Semantic: friendship, ‘personal relationships’

Figure 5.2: Various relationship feature-sets extracted for a sample sentence depicting relationship between Jim and Maria.

had either of the two characters as the agent or patient, provided the sentence did not contain a mention of another character.

Lexical: This bag-of-words set consists of all words (except stopwords) that appeared between pair of mentions of the two characters. Example in Figure 5.2.

Frame-semantic: This set makes use of the frame-semantic parse of the sentence and the frame-polarity lexicon, which contains a list of frames indicative of relationship status of the frame-elements (see Section 4.4 and Appendix A). This set includes all frames (and the tokens at which they were fired), included in the above mentioned lexicon, fired for at least one of the characters as a *frame-element*. For example in Figure 5.2, a ‘personal relationships’ frame is evoked at the token ‘friendship’. Hence, this set will include the words ‘friendship’ and ‘personal relationships’.

After extracting these sets of words from individual sentences, we obtain a vector representation, $\vec{f}_t \in \mathbb{R}^D$, for each sentence, s_t , by averaging the vector space embeddings [76] of the individual words in the union of these sets (motivated by the additive model of vector compositionality [78]).

5.4 Learning Relationship Sequences

Given the feature vectors as input, $\mathbf{f} = \langle \vec{f}_1, \vec{f}_2 \dots \vec{f}_T \rangle$, we now describe our models which learn the relationship sequence, $\mathbf{r} = \langle r_1, r_2 \dots r_T \rangle$. Our first approach, *GHMM* – a Gaussian Hidden Markov Model, uses a non-Bayesian Hidden Markov Model with Gaussian Emissions. The hidden states comprise of relationship states and vector representation of sentences form the observations. Our second approach, *Penalized HMM*, extends GHMM by *smoothing* the relationship sequences and discouraging frequent changes in relationship states within a sequence. Finally, our last approach, *Globally Aware GHMM*, attempts to simulate the intuition of a *global belief* about the nature of relationship between the characters, while analyzing the individual sentences of the sequence.

5.4.1 GHMM

Our first approach consists of a Hidden Markov Model with Gaussian Emissions, which generates the feature-vector sequence as:

For every vector, $\vec{f}_t \forall t \in \{1, 2, 3 \dots T\}$:

1. If $t = 1$, choose $r_1 \sim \text{Categorical}(\boldsymbol{\pi})$
2. If $t > 1$, choose $r_t \sim \text{Categorical}(\boldsymbol{\phi}_{\mathbf{r}_{t-1}})$
3. Emit vector $\vec{f}_t \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{r}_t}, \boldsymbol{\Sigma}_{\mathbf{r}_t})$

where, $\boldsymbol{\pi}$ is the start state probabilities – an R -dimensional probability distribution indicating the probability of starting in a given state. Also, $\boldsymbol{\phi}_{r_{t-1}}$ represents the transition probabilities, such that ϕ_{ij} is the probability of transitioning from state i to state j , and $\sum_{j=1}^R \phi_{ij} = 1$. Finally, it is assumed that the vectors belonging to a state r are normally distributed with mean $\boldsymbol{\mu}_r$ and covariance, $\boldsymbol{\Sigma}_r$.

This model thus defines the joint distribution over a sequence of feature-vectors as:

$$p(\mathbf{f}, \mathbf{r}) = \prod_{t=1}^T p(r_t|r_{t-1})p(\vec{f}_t|r_t) \quad (5.1)$$

where, $p(r_t|r_{t-1})$ and $p(\vec{f}_t|r_t)$ are obtained from $\boldsymbol{\phi}_{r_{t-1}r_t}$ and the Normal distribution with parameters $\boldsymbol{\mu}_{r_t}$ and $\boldsymbol{\Sigma}_{r_t}$. We use the Baum-Welch algorithm to fit the various parameters of this model.

5.4.2 Penalized GHMM

In practice GHMM resulted in highly fluctuating relationship sequences. While this might be a good feature for traditional sequence modeling tasks like POS tagging, real-world relationship sequences are smooth and tend to remain consistent over long parts of a narrative. We, therefore, introduce a more domain-specific model, Penalized GHMM, which is similar to GHMM, except that in the generative process, every time the model makes a transition from state i to state j , it incurs a penalty, ρ_{ij} .

$$\rho_{ij} = \begin{cases} 1, & \text{if } i = j \\ \epsilon, & \text{otherwise} \end{cases} \quad (5.2)$$

This model defines the joint distribution over a sequence of feature-vectors as:

$$p(\mathbf{f}, \mathbf{r}) = \prod_{t=1}^T p(r_t|r_{t-1})\rho_{r_{t-1},r_t}p(\vec{f}_t|r_t) \quad (5.3)$$

The parameter estimation process for this model is similar to that of GHMM.

5.4.3 Globally Aware GHMM

The above models are *local* in nature, in the sense that at any point in the sequence, t , the relationship state, r_t , depends only on the previous state, r_{t-1} , and the emitted features, \vec{f}_t . However, as discussed in Section 5.2, inferring relationship state requires a more global perspective, which is aware of the overall nature of the relationship between the characters.

To incorporate this behavior, we introduce a third model, Globally Aware GHMM. This model makes a decision about the current relationship state, r_t , after weighing in information from a local and a global component using a *choice* variable, $c_t \in \{0, 1\}$. The local component uses the Penalized GHMM style transitions to determine the current relationship state. Whereas, the global component (represented by θ) uses a logistic regression model based on a global feature-set, \vec{F} , extracted from the whole sequence, $\mathbf{s} = \langle s_1, s_2 \dots s_T \rangle$.

This model defines the distribution over a sequence as:

$$p(\mathbf{f}, \mathbf{r}) = \prod_{t=1}^T [\gamma \cdot p(r_t|r_{t-1}) \cdot \rho_{r_{t-1},r_t} + (1 - \gamma) \cdot \theta(r_t|\vec{F})] \cdot p(\vec{f}_t|r_t) \quad (5.4)$$

Here γ is the model's preference towards the local model ($\gamma = p(c = 1)$) and

the global component, θ , is modeled as:

$$\theta(r|\vec{F}) = \frac{\exp(\vec{w}_r \cdot \vec{F})}{\sum_{r'=1}^R \exp(\vec{w}_{r'} \cdot \vec{F})} \quad (5.5)$$

where, \vec{F} is the global feature-set and \vec{w}_r are the weights corresponding to the relationship state r that are learned during training.

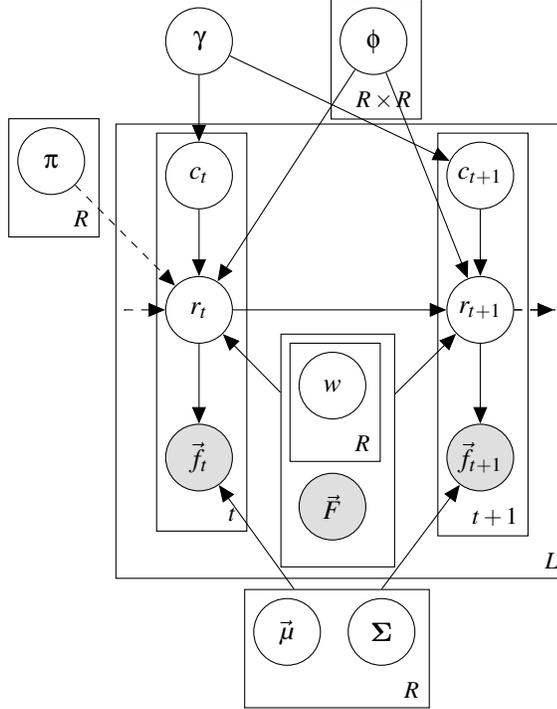


Figure 5.3: Diagram for the Globally Aware GHMM (for 2 sentences of a sequence). Feature-vector representations of individual sentences, \vec{f}_t , and that of the complete sequence (global), \vec{F} , are observed. Relationship states, r_t and global versus local choices, c_t are hidden. γ is the model's preference towards the local component. \vec{w} represents the weights of the global component (with a logistic regression form), and ϕ and π represent the local component (transition and start-state probabilities respectively). Penalty ρ is not shown for clarity. $\vec{\mu}$ and Σ are the parameters of Normal distribution. L is the number of sequences in the dataset and R is the number of relationship states.

Figure 5.3 pictorially describes our model and its generative story can be described as follows:

For every vector, $\vec{f}_t \forall t \in \{1, 2, 3 \dots T\}$:

1. Toss a *choice* variable, $c_t \sim \text{Bernoulli}(\gamma)$.
2. If $c_t = 0$, choose $r_t \sim \theta(r|\vec{F})$
3. If $c_t = 1$ & $t = 1$, choose $r_1 \sim \text{Categorical}(\boldsymbol{\pi})$
4. If $c_t = 1$ & $t > 1$, choose $r_t \sim \text{Categorical}(\boldsymbol{\phi}_{\mathbf{r}_{t-1}}) \cdot \rho_{r_{t-1}r_t}$
5. Emit vector $\vec{f}_t \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{r}_t}, \boldsymbol{\Sigma}_{\mathbf{r}_t})$

Training: Globally Aware GHMM is defined using its parameters, $\lambda = (\boldsymbol{\pi}, \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{w}, \gamma)$.

We train the model using EM. In each step of the process, let λ represent the current model and λ' represent a candidate model. Our goal is to make $p_{\lambda'}(\mathbf{f}) > p_{\lambda}(\mathbf{f})$. It can be shown that this is equivalent to maximizing the following auxiliary function:

$$Q(\lambda, \lambda') = \sum_l \sum_r \sum_c p_{\lambda}(\mathbf{r}^l, \mathbf{c}^l | \mathbf{f}^l) \log p_{\lambda'}(\mathbf{r}^l, \mathbf{c}^l, \mathbf{f}^l) \quad (5.6)$$

where, l is the index over various sequences in the dataset (L in number) and x^l represents the variable, x , for the l^{th} sequence. In the above equation, $p(\mathbf{r}, \mathbf{c}, \mathbf{f})$ for a sequence is modeled as:

$$p(\mathbf{r}, \mathbf{c}, \mathbf{f}) = \prod_{t=1}^T \{ \gamma \cdot p(r_t | r_{t-1}) \cdot \rho_{r_{t-1}, r_t} \}^{\delta_1(c_t)} \{ (1 - \gamma) \cdot \theta(r_t | \vec{F}) \}^{\delta_0(c_t)} \cdot p(\vec{f}_t | r_t) \quad (5.7)$$

where, $\delta_a(x)$ is the Kronecker delta function which takes the value of 1 whenever

$x = a$ and 0 otherwise. In the E-step, we use the scaled Forward-backward algorithm to compute scaled forward ($\alpha_{rc}(t)$) and backward ($\beta_r(t)$) probabilities for a sequence, l , as:

$$\alpha_{rc}^l(t) = p(r_t^l = r, c_t^l = c | \vec{f}_1^l, \vec{f}_2^l \dots \vec{f}_t^l, \lambda) \quad (5.8)$$

$$\beta_r^l(t) = p(\vec{f}_{t+1}^l, \vec{f}_{t+2}^l \dots \vec{f}_{T^l}^l | r_t^l = r, \lambda) \quad (5.9)$$

In the M-step we update the parameters as:

$$\mu_{rn}^{new} = \frac{\sum_l^L \sum_t^{T^l} \beta_r^l(t) \cdot \alpha_r^l(t) \cdot f_{tn}^l}{\sum_l^L \sum_t^T \beta_r^l(t) \cdot \alpha_r^l(t)} \quad (5.10)$$

$$\Sigma_r^{new} = \frac{\sum_l^L \sum_t^{T^l} \beta_r^l(t) \cdot \alpha_r^l(t) (\vec{f}_t^l - \vec{\mu}_r) (\vec{f}_t^l - \vec{\mu}_r)^T}{\sum_l^L \sum_t^T \beta_r^l(t) \cdot \alpha_r^l(t)} \quad (5.11)$$

$$\pi_r^{new} = \frac{\sum_l^L \beta_r^l(1) \cdot \alpha_{r1}^l(1)}{\sum_l^L \sum_i^R \beta_i^l(1) \cdot \alpha_{i1}^l(1)} \quad (5.12)$$

$$\gamma^{new} = \frac{\sum_l^L \sum_t^{T^l} \sum_r^R \beta_r^l(t) \cdot \alpha_{r1}^l(t)}{L} \quad (5.13)$$

$$\phi_{ij}^{new} = \frac{\sum_l^L \sum_t^{T^l-1} \xi_{ij}^l(t)}{\sum_l^L \sum_t^{T^l-1} \sum_j^R \xi_{ij}^l(t)} \quad (5.14)$$

using $\alpha_r^l(t)$ to represent $\sum_{c=\{0,1\}} \alpha_{rc}^l(t)$, and $\eta^l(t)$ to represent the scaling factor used in scaling the forward probabilities at the t^{th} point of the l^{th} sequence, we define $\xi_{ij}^l(t)$ as:

$$\xi_{ij}^l(t) = \frac{\beta_j^l(t+1) \cdot \phi_{ij} \cdot \gamma \cdot p(\vec{f}_{t+1}^l | r_{t+1}^l = j) \alpha_r^l(t)}{\eta^l(t)} \quad (5.15)$$

In this step, we also learn the weights, \vec{w}_r , of the global component (Eqn. 5.5) by maximizing the following objective function using a gradient-descent based method:

$$\sum_l^L \sum_r^R \frac{\exp(\vec{w}_r \cdot \vec{F}^l)}{\sum_{r'=1}^R \exp(\vec{w}_{r'} \cdot \vec{F}^l)} \sum_{t=2}^{T^l} \beta_r^l(t) \cdot \alpha_{i0}^l(t) \quad (5.16)$$

5.5 Data

For our experiments we have used the dataset described in Chapter 4. It consists of plot overviews of around 300 English novels from SparkNotes.¹ These summaries were pre-processed to identify major characters, and pairs of characters that appeared together in more than 5 sentences were considered for analysis. This threshold was used in order to obtain character-pairs that interacted long enough to demonstrate the dynamic nature of their relationships but at the same time resulted in a sizeable dataset. These sentences were arranged linearly and form a *sequence*. The final dataset contained 634 such sequences, with an average length of 8.2 sentences per sequence.

5.6 Evaluation

In this section we evaluate our models to answer the following questions:

- How does the performance of our unsupervised models compare with each other and with their supervised counterpart?
- Does incorporating global context help in improving performance?

¹<http://www.sparknotes.com/sparknotes/>

- How does the model performance compare with human judgments in characterizing inter-character relationships?
- Are the latent relationship states learned by our model semantically coherent?

Implementation Details: Section 5.3 described the process of conversion of sentences of the narrative to feature-vectors. To obtain word embeddings in this process, we used the skip-gram model (using the Word2Vec tool [76]). This model was trained with $D = 200$ on a collection of the unstructured text of a subset of the novels from Project Gutenberg.² The novels were pre-processed to remove punctuation and capitalization before training the tool.

Also, Globally Aware GHMM needs a global-features set, F , extracted for the complete sequence of sentences. Our model doesn't make any assumption about the nature of this feature-set. For our experiments, we use the average of the feature-vectors of all sentences in the sequence (i.e $\vec{F} = \text{mean}(\vec{f}_1, \vec{f}_2 \dots \vec{f}_T)$).

We used an existing implementation of the GHMM model [81]. Also, we use the average of the feature-vectors of all sentences in a sequence as its global feature vector for our Globally Aware GHMM (i.e $\vec{F} = \text{mean}(\vec{f}_1, \vec{f}_2 \dots \vec{f}_T)$). We used $\epsilon = 0.8$ (selected using cross-validation). Estimating the covariance matrix Σ degraded performance, which might be due to overfitting [95]. Hence, we only show results for estimating $\vec{\mu}_r$, and we use a fixed diagonal matrix as Σ (with each diagonal entry being 0.01), following previous approaches [68].

²<https://www.gutenberg.org/>

5.6.1 Supervised Evaluation

We begin with indirectly evaluating our models on a supervised task by heuristically aligning learnt latent states against label categories. For this purpose, we use the manually annotated sequences of the data described in Chapter 4. It consists of about 50 sequences in which each sentence is labeled with a binary relationship state, *cooperative* or *non-cooperative*, which we refer to as the gold-classes. Around 30% of the sequences were labeled with more than one class. However, our unsupervised models assign each sentence to a relationship state/cluster but do not provide a label to the states. For this evaluation we heuristically assign each of the learned states a cooperative/non-cooperative label using:

$$\text{label of state } j = \arg \max_{i \in \{\text{coop}, \text{non-coop}\}} \left\{ \frac{m_i^j}{N_i} \right\} \quad (5.17)$$

where, m_i^j is the number of sentences belonging to the learned state j with gold-class= i , and N_i is the total number of sentences in the gold-class i . We adopted this method of labeling states instead of simply assigning each state to the gold-class that was most frequent in the state because of class skew in numbers of manual annotations in the data. Like Chapter 4, we report averaged precision, recall and F-measures of the two gold-classes. Since our models require random initialization of global weights and means of the relationship states, we report average values for 50 runs for each of the models.

Figure 5.4 compares the performance of our models for various values of R –the number of relationship states, which is a user-provided input to our models. We can

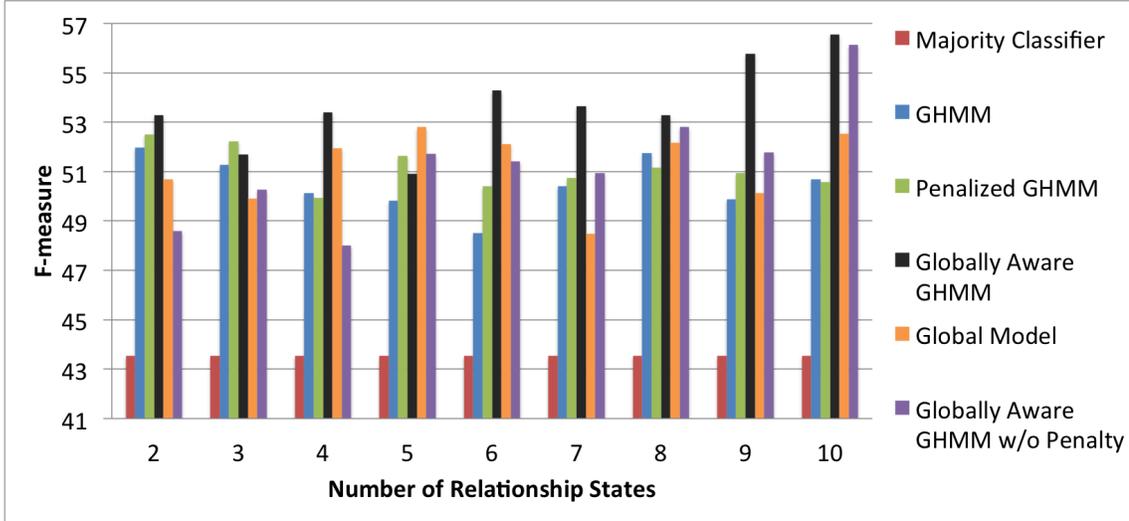


Figure 5.4: Performance comparison of various models. Previously, we obtained an F-measure of 76.76 on the same dataset using supervised models (Chapter 4).

see that, all our models significantly outperform the baseline, which always predicts the majority (cooperative) class. Also the supervised model proposed in Chapter 4 achieved an F-measure of 76.76 on the same dataset but with expensive manually annotated data required for training. The figure shows that Globally Aware GHMM (black), in general, performs better than the Penalized GHMM (green) and the GHMM (blue). It also outperforms the Global Model (orange), which is an unstructured model that clusters the sentences independently (corresponds to the global only component of the Globally Aware GHMM). This indicates that for this task, it is important to have a global as well as local perspective of characters' interactions.

The performance of Penalized GHMM (green) is comparable to that of GHMM (blue), which hints that the penalty term might not be contributing significantly to the model's performance. To investigate this further, we modified Globally Aware

State	Most Frequent words
Familial	kinship, father, relationship, personal, love, mother, son, brother, family, wife, daughter, tell, friend, marry, child, sister
Desire	relationship, become, make, decide, reveal, personal, find, kinship, desiring, forming relationships, learn, marriage, marry
Casual	go, meet, come, leave, take, find, return, together, arrive, get, tell, run, bring, decide, home, enter, begin, make, try, ask
Verbal	tell, ask, say, want, see, know, tells, find, realize, desiring, love, leave, marry, think, learn, believe, asks, try, kill, have
Hostile	kill, killing, attack, protect, cause harm, encounter, die, tell, fight, hostile, try, kinship, take, back, find, make, protecting

Table 5.1: Representative words for relationship states learned by the Globally Aware GHMM.

GHMM to exclude the penalty term. We call this model ‘Globally Aware GHMM w/o penalty’ (purple). We can see that its performance is much worse than that of Globally Aware GHMM (black). This suggests that while the penalty term is not very useful for a local model like GHMM, it is indeed valuable for models like Globally Aware GHMM, which have an ability to switch between local and global components. This is because frequent switching between the two components can result in frequent shifting between relationship states within a sequence. This frequent shifting is unnatural for the given task because the states represent inter-personal relationships, which are usually stable and evolve smoothly with the narrative. Therefore, such a model benefits from a penalty term, which smoothens the relationship trajectories and makes them more consistent with human judgment. This can also be observed in Figure 5.5. The Globally Aware GHMM (black) has much stronger preference for one of the components (global) than Globally Aware

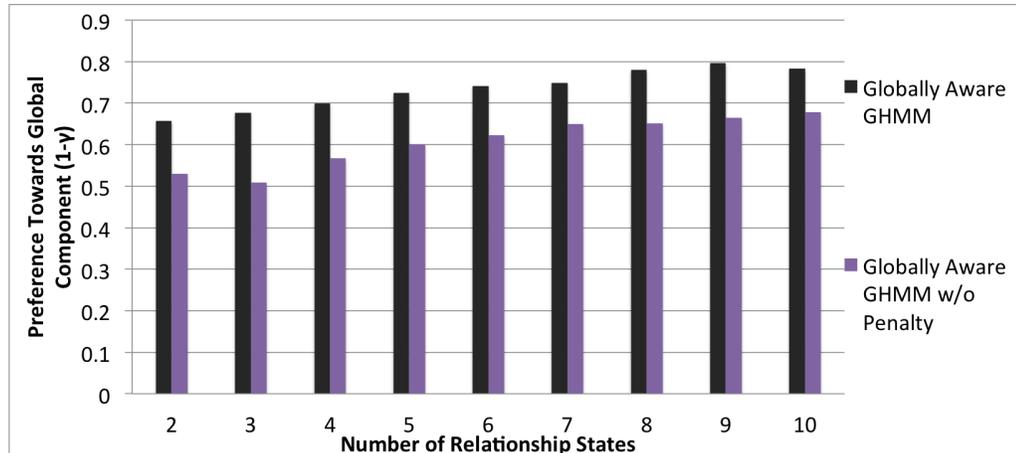


Figure 5.5: Global vs local preference learned by the Globally Aware GHMM with and without penalty. The model without penalty has weaker preference for a global categorization which can result in frequent shifts in relationship states within a sequence, resulting in lower accuracy.

GHMM w/o penalty (purple). The preference learned for the latter model is closer to 0.5 which would result in frequent shifts between the two components.

5.6.2 Relationship Analogy Task

We now evaluate our model against human judgment without restrictive assumptions on the types of relationships. Manually designing a taxonomy of relationship types is challenging. Further, judging the quality of a learned relationship sequence using a given taxonomy is subjective, and can vary considerably with choices in design.

Hence, we evaluate the models on an objective task involving human subjects to answer questions based on semantics of inter-character relationships. Specifically, given a pair (P) of characters in a text, we asked subjects to pick a pair that reflects

Whose relationship is more similar to the relationship between Harry Potter and Ron Weasley in 'Harry Potter and The Chamber of Secrets'?

- Hagrid and Harry Potter in 'Harry Potter and The Prisoner of Azkaban'
- Sirius Black and Ron Weasley in 'Harry Potter and The Prisoner of Azkaban'
- I don't know

Rate your confidence:

- High
- Low

Comments (optional):

Figure 5.6: Screenshot from human evaluation task

a similar relationship from two choices of pairs of characters, O_1 and O_2 . Figure 5.6 shows an example question. The two options were selected from pool of pairs similar (or dissimilar) to the given pair, P , as determined by our model. We use an edit-distance based measure to compute similarity between (normalized) relationships sequences of any two given pairs.

Subjects were required to read the summaries of these books before they answered questions, but also had access to these during the task. An illustrative sample question and answer was initially provided to explain the task. Subjects were also reminded to consider that personal relationships can have multiple facets and can change over time. They were asked to consider not just the nature but also the trajectories of relationships while judging similarity. For each question, the subjects could choose one of the two candidates character-pairs, or a *don't know* option, and could report their confidence in doing so (high/low). To reduce annotator fatigue, each annotator was asked questions about characters mentioned in only three books

and could not answer more than 10 questions per session. Overall, we collected answers for about 100 such questions and each question was answered by at least three annotators. The inter-annotator agreement on this task was 0.73 (chance adjusted Fleiss' $\kappa = 0.46$).

We then evaluated the agreement between the model's judgment and the human provided answers. The annotators agreed with our model's similarity judgment in 66.0% of the questions. This is a considerably hard task (since the inter-annotator agreement is not high) and a random baseline would agree only half the time. Hence we can conclude that the model learns sequences that correlate significantly with human judgment ($p < 10^{-3}$), and can be used to address semantically complicated questions.

5.6.3 Coherence of Relationship States

Table 5.1 shows the most frequent words from the relationship states learned by the Globally Aware GHMM ($R = 5$). For the purpose of this visualization, for each state, we report the most frequent words from the union of the feature-sets extracted (Sec. 5.3) for all the sentences assigned to that state. We can see that the first state corresponds to familial relationships as it consists of words like *father*, *family*, etc. The second state corresponds to a desire to initiate relationships as it contains words like *desiring*, *forming relationships*, *marriage*, etc. The third state consists of sentences in which the characters participate in physical action like *go*, *meet*, *arrive*, etc. The last two states depict casual verbal communication and hostile

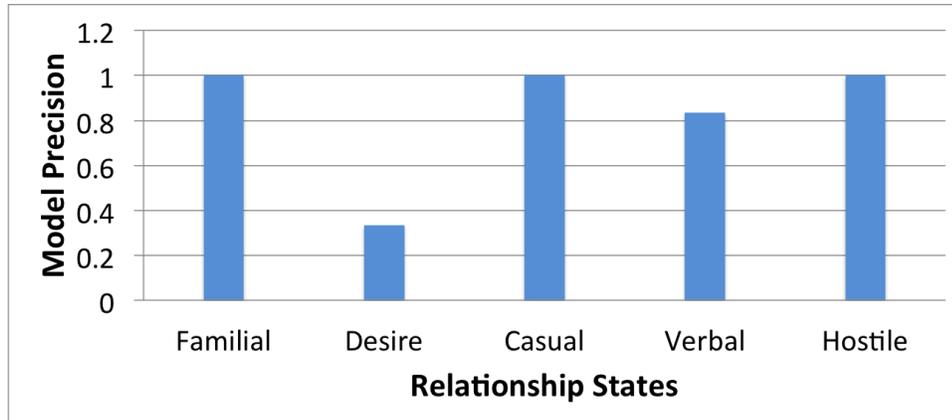


Figure 5.7: Model Precision (MP) for the Word-intrusion detection task for relationship states learned by our model. MP is high for most states indicating semantic coherence.

relationship respectively.

Word-intrusion Detection Task: To further investigate the semantic coherence of the states, we performed the ‘Word-intrusion detection’ task [24]. In this task, a human subject is presented with six randomly ordered words. Five of these words are high frequency words from one of our learned relationship states, and one of the words, the ‘intruder’, belongs to a different (randomly chosen) relationship state. The subjects are then asked to identify the intruder word. The subjects in our experiments were graduate students from varying disciplines and were comfortable with English. Each subject was shown at-least five sets of words (one corresponding to each state), and no subject was shown more than ten sets. We then calculate the ‘Model Precision’ for each topic which is the fraction of times a subject accurately identified the intruder. Figure 5.7 shows the results of our experiment. We can see

that the subjects successfully identified the intruder with high precision in all cases ($p < 0.001$) except the ‘desire’ state ($p \sim 0.3$). We can thus conclude that the states learned by our model are semantically coherent.

5.7 Conclusion

This chapter relaxed the limiting assumptions about the number of possible relationship types made in the previous chapters. In this chapter, we presented methods to discover various types of relationships in an unsupervised manner. In particular, we addressed the problem of unsupervised learning of evolving relationship between characters from novel summaries. Treating this as a structured prediction problem we present three different models that are based on linguistic information as well as real-world knowledge. Our experiments reveal that for solving this problem, it is not sufficient to simply look at local historical cues about the relationships between the two characters of interest from a small part of text at any juncture. Instead, it is important to incorporate a *global context* of the overall nature of relationship between them. We also empirically demonstrate the semantic coherence of the relationship states learned by our model.

Chapter 6: Application: Identifying Existence of Instructor-student Relationship in MOOC Forums

Work described in this chapter was published in ACL 2014 and AAAI 2016 [26, 27].

In this chapter, we present an application of our relationship modeling approach to a real-world task in the domain of educational discussion forums. We work with MOOC (Massive Open Online Course) forums, which provide a platform for students enrolled in MOOCs to interact with the instructional staff. In this domain, we address the problem of predicting if an instructor would intervene on a thread, assuming that posts of threads are chronologically structured in a chain-like fashion. As mentioned earlier, the underlying analogy here models the intervention of an instructor in a discussion as the initiation of a deeper relationship, in context of the historical content of the thread.

We model the problem as identification of relationship existence because in some sense, discussion forums provide a platform for students to form deeper relationships with instructors. In fact in MOOCs, forums are the only way for students to interact personally with the instructors. When a student posts on a thread seeking instructor’s attention, (s)he is attempting to initiate such a relationship with

the instructor. And when the instructor responds to the student's query, (s)he confirms/forms the relationship. Thereafter, depending on the content of their posts, this relationship takes concrete form/nature. For example, if the instructor supports/endorse a student's answer in a thread, their relationship takes a 'positive' form. Hence, in future, the instructor is likely to have more trust on (positive bias towards) this particular student. In this work, we limit ourselves to predicting if the relationship would be formed, and don't delve deeper into the nature of relationship (whether the instructor supports/disproves/trusts the claim/query/comment posted by the student).

We considered two different MOOCs and found their content to be not very different from narratives. The forums follow a coherent well-organized development, with each thread attributed to a title, which summarizes the query posted by a student. Posts in a thread contain comments/developments relevant to the title, and follow a natural temporal order, similar to most narratives. In fact, as discussed in this chapter, we propose two models that utilize this narrative-schema-like lineal arrangement of posts and our experiments demonstrate that doing so helps in making better judgments about the instructor's decision to post. We additionally demonstrate the utility of our model by addressing another task of identifying if a desire expressed in a short paragraph is fulfilled.

6.1 Problem Motivation and Definition

Ubiquitous computing and easy access to high bandwidth internet have reshaped the *modus operandi* in distance education towards Massive Open Online Courses (MOOCs). MOOCs impart inexpensive and high-quality education from field-experts to thousands of learners across geographic and cultural barriers. Even as the MOOC model shows exciting possibilities, it presents a multitude of challenges that must be addressed [111, 112, 85, 90, 5, 58, 82]. MOOCs platforms have been especially criticized on grounds of lacking a personalized educational experience [37]. Unlike traditional classrooms, the predominant mode of interaction between students and instructors in MOOCs is via online discussion forums. Ideally, forum discussions can help make up for the lack of direct interaction, by enabling students to ask questions and clarify doubts. However, due to huge class sizes, MOOCs witness a very large number of threads on these forums. Owing to extremely skewed ratios of students to instructional staff, it can be prohibitively time-consuming for the instructional staff to manually follow all threads of a forum. Hence there is a pressing need for automatically *curating* the discussions for the instructors. Analyzing forum-postings contents and bringing the most pertinent content to the instructor's attention would help instructors receive timely feedback and design interventions as needed. From the students' perspective, the problem is evident from an examination of existing forum content, indicating that if students want instructor's input on some issues, the only way for them to get his/her attention is by 'up-voting' their votes. Figure 6.1 provides some examples of this

“The problem summary: Anyone else having problems viewing the video lecture...very choppy. If you are also experiencing this issue; please upvote this post.”

“I read that by up-voting threads and posts you can get the instructors’ attention faster.”

“Its is very bad to me that I achieved 10 marks in my 1st assignment and now 9 marks in my 2nd assignment, now I won’t get certificate, please Course staff it is my appeal to change the passing scheme or please be lenient. Please upvote my post so that staff take this problem under consideration.”

Figure 6.1: Example posts demonstrating that in order to attract instructor’s attention students have to resolve to the inefficient method of getting their posts upvoted by their classmates. Therefore, there is a need to design models that could automatically identify threads on which instructors intervene.

behavior. This is clearly an inefficient solution.

Thus, we focus on identifying situations in which instructor (interchangeable with “instructional staff”) intervention is warranted. In our description it is assumed that a forum consists of multiple threads. Each thread (t) has a title and consists of multiple posts (p_i). Individual posts do not have a title and the number of posts varies dramatically from one thread to another. We address the problem of predicting if the course instructor would intervene on a thread, t . The instructor’s decision to intervene, r , equals 0 when the instructor doesn’t reply to the thread and 1 otherwise. Individual posts are not assumed to be labeled with any category and the only supervision given to the model during training is in form of intervention decision.

6.2 Role of Context

While the textual content of the discussion forum can be expected to provide significant cues about instructor’s intervention decision, as indicated before, it is equally important to view an individual post in context of the other posts in the thread. Therefore, we propose to leverage the structure of the forum thread in order to maximally utilize its textual contents. We assume that posts of a thread structure it in form of a story or a ‘chain of events’. For example, an opening post of a thread might pose a question and the following posts can then answer or comment on the question.

Our models tap this linear ‘chain of events’ behavior of a thread by assuming that individual posts belong to latent categories which represent their textual content at an abstract level and that an instructor’s decision to reply to a post is based on this chain of events (represented by the latent categories). We present two different ways of utilizing this ‘chain of events’ behavior for predicting instructor’s intervention which can be either simply modeled as the ‘next step’ is this chain of events (*Linear Chain Markov Model-LCMM*) or as a decision globally depending on the entire chain (*Global Chain Model-GCM*).

6.3 Models

In this section, we explain our models in detail. In our description it is assumed that a discussion board is organized into multiple forums (representing topics such

as “Assignment”, “Study Group”, etc.). A forum consists of multiple threads. Each thread (t) has a title and consists of multiple posts (p_i). Individual posts do not have a title and the number of posts varies dramatically from one thread to another. We address the problem of predicting if the course instructor would intervene on a thread, t . The instructor’s decision to intervene, r , equals 0 when the instructor doesn’t reply to the thread and 1 otherwise. The individual posts are not assumed to be labeled with any category and the only supervision given to the model during training is in form of intervention decision.

6.3.1 Logistic Regression (LR)

Our first attempt at solving this problem involved training a logistic regression for the binary prediction task which models $P(r|t)$.

Feature Engineering

Our logistic regression model uses the following two types of features: *Thread only features* and *Aggregated post features*. ‘Thread only features’ capture information about the thread such as when, where, by who was the thread posted and lexical features based on the title of the thread. While these features provide a high-level information about the thread, it is also important to analyze the contents of the posts of the thread. In order to maintain a manageable feature space, we compress the features from posts and represent them using our ‘Aggregated post features’.

Thread only features:

1. a binary feature indicating if the thread was started by an anonymous user

2. three binary features indicating whether the thread was marked as approved, unresolved or deleted (respectively)
3. forum id in which the thread was posted
4. time when the thread was started
5. time of last posting on the thread
6. total number of posts in the thread
7. a binary feature indicating if the thread title contains the words *lecture* or *lectures*
8. a binary feature indicating if the thread title contains the words *assignment*, *quiz*, *grade*, *project*, *exam* (and their plural forms)

Aggregated post features:

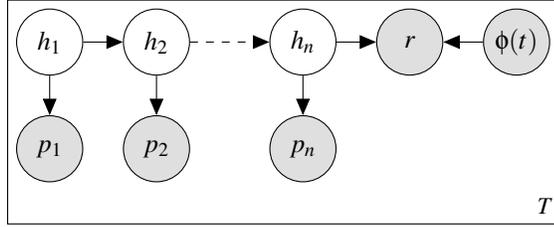
9. sum of number of votes received by the individual posts
10. mean and variance of the posting times of individual posts in the thread
11. mean of time difference between the posting times of individual posts and the closest course landmark. A course landmark is the deadline of an assignment, exam or project.
12. sum of count of occurrences of assessment related words e.g. *grade*, *exam*, *assignment*, *quiz*, *reading*, *project*, etc. in the posts

13. sum of count of occurrences of words indicating technical problems e.g. *problem, error*
14. sum of count of occurrences of thread conclusive words like *thank you* and *thank*
15. sum of count of occurrences of *request, submit, suggest*

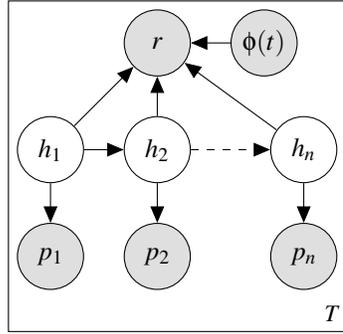
We had also considered and dropped (because of no performance gain) other features about identity of the user who started the thread, number of distinct participants in the thread (an important feature used for a similar task previously [7]), binary feature indicating if the first and the last posts were by the same user, average number of words in the thread's posts, lexical features capturing references to the instructors in the posts, etc.

6.3.2 Linear Chain Markov Model (LCMM)

The logistic regression model is good at exploiting the thread level features but not the content of individual posts. The 'Aggregated post features' attempt to capture this information but since the number of posts in a thread is variable, these features relied on aggregated values. We believe that considering aggregate values is not sufficient for the task in hand. As noted before, posts of a thread are not independent of each other. Instead, they are arranged chronologically such that a post is published in reply to the preceding posts and this might effect an



(a) Linear Chain Markov Model (LCMM)



(b) Global Chain Model (GCM)

Figure 6.2: Diagrams of the Linear Chain Markov Model (LCMM) and the Global Chain Model (GCM). p_i , r and $\phi(t)$ are observed and h_i are the latent variables. p_i and h_i represent the posts of the thread and their latent categories respectively; r represents the instructor’s intervention and $\phi(t)$ represent the non-structural features used by the logistic regression model.

instructor’s decision to reply. For example, consider a thread that starts with a question. The following posts will be students’ attempt to answer the question or raise further concerns or comment on previous posts. The instructor’s post, though a future event, will be a part of this process.

We, therefore, introduce a new structured model, LCMM, to incorporate this complete process (shown in Figure 6.2a). The model abstractly represents the information from individual posts (p_i) using latent categories (h_i). The intervention

For every thread, t , in the dataset:

1. Choose a start state, h_1 , and emit the first post, p_1 .
2. For every subsequent post, $p_i \forall i \in \{2 \dots n\}$:
 - (a) Transition from h_{i-1} to h_i .
 - (b) Emit post p_i .
3. Generate the instructor’s intervention decision, r , using the last state h_n and non-structural features, $\phi(t)$.

Figure 6.3: Instructor’s intervention decision process for the Linear Chain Markov Model.

decision, r , is the last step in the chain and thus incorporates information from the individual posts. It also depends on the thread level features: ‘Thread only features’ and the ‘Aggregated post features’ jointly represented by $\phi(t)$ (also referred to as the non-structural features). This process is explained in Figure 6.3.

We use hand-crafted features to model the dynamics of the generative process. Whenever a latent state emits a post or transits to another latent state (or to the final intervention decision state), emission and transition features get fired which are then multiplied by respective weights to compute a thread’s ‘score’:

$$f_w(t, p) = \max_h [\mathbf{w} \cdot \phi(\mathbf{p}, \mathbf{r}, \mathbf{h}, \mathbf{t})] \quad (6.1)$$

Note that the non-structural features, $\phi(t)$, also contribute to the final score.

Learning and Inference

During training we maximize the combined scores of all threads in the dataset using a generic EM style algorithm. The supervision in this model is provided only in form of the observed intervention decision, r and the post categories, h_i are hidden. The model uses the pseudocode shown in Algorithm 3 to iteratively refine the weight vectors. In each iteration, the model first uses the Viterbi algorithm to decode thread sequences with the current weights w_t to find optimal highest scoring latent state sequences that agree with the observed intervention state ($r = r'$). In the next step, given the latent state assignments from the previous step, a structured perceptron algorithm [31] is used to update the weights w_{t+1} using weights from the previous step, w_t , initialization.

Algorithm 3 Training algorithm for LCMM

- 1: **Input:** Labeled data $D = \{(t, p, r)_i\}$
 - 2: **Output:** Weights w
 - 3: **Initialization:** Set w_j randomly, $\forall j$
 - 4: **for** $t : 1$ to N **do**
 - 5: $\hat{h}_i = \arg \max_h [\mathbf{w}_t \cdot \phi(\mathbf{p}, \mathbf{r}, \mathbf{h}, \mathbf{t})]$ such that $r = r_i \forall i$
 - 6: $w_{t+1} = \text{StructuredPerceptron}(t, p, \hat{h}, r)$
 - 7: **end for**
 - 8: **return** w
-

While testing, we use the learned weights and Viterbi decoding to compute the intervention state and the best scoring latent category sequence.

Feature Engineering

In addition to the ‘Thread Only Features’ and the ‘Aggregated post features’, $\phi(t)$ (Section 6.3.1), this model uses the following emission and transition features:

Post Emission Features:

1. $\phi(p_i, h_i)$ = count of occurrences of question words or question marks in p_i if the state is h_i ; 0 otherwise.
2. $\phi(p_i, h_i)$ = count of occurrences of thank words (*thank you* or *thanks*) in p_i if the state is h_i ; 0 otherwise.
3. $\phi(p_i, h_i)$ = count of occurrences of greeting words (e.g. *hi*, *hello*, *good morning*, *welcome*, etc.) in p_i if the state is h_i ; 0 otherwise.
4. $\phi(p_i, h_i)$ = count of occurrences of assessment related words (e.g. *grade*, *exam*, *assignment*, *quiz*, *reading*, *project*, etc.) in p_i if the state is h_i ; 0 otherwise.
5. $\phi(p_i, h_i)$ = count of occurrences of *request*, *submit* or *suggest* in p_i if the state is h_i ; 0 otherwise.
6. $\phi(p_i, h_i) = \log(\text{course duration}/t(p_i))$ if the state is h_i ; 0 otherwise. Here $t(p_i)$ is the difference between the posting time of p_i and the closest course landmark (assignment or project deadline or exam).
7. $\phi(p_i, p_{i-1}, h_i)$ = difference between posting times of p_i and p_{i-1} normalized by course duration if the state is h_i ; 0 otherwise.

Transition Features:

1. $\phi(h_{i-1}, h_i) = 1$ if previous state is h_{i-1} and current state is h_i ; 0 otherwise.
2. $\phi(h_{i-1}, h_i, p_i, p_{i-1}) = \text{cosine similarity between } p_{i-1} \text{ and } p_i$ if previous state is h_{i-1} and current state is h_i ; 0 otherwise.
3. $\phi(h_{i-1}, h_i, p_i, p_{i-1}) = \text{length of } p_i$ if previous state is h_{i-1} , p_{i-1} has non-zero question words and current state is h_i ; 0 otherwise.
4. $\phi(h_n, r) = 1$ if last post's state is h_n and intervention decision is r ; 0 otherwise.
5. $\phi(h_n, r, p_n) = 1$ if last post's state is h_n , p_n has non-zero question words and intervention decision is r ; 0 otherwise.
6. $\phi(h_n, r, p_n) = \log(\text{course duration}/t(p_n))$ if last post's state is h_n and intervention decision is r ; 0 otherwise. Here $t(p_n)$ is the difference between the posting time of p_n and the closest course landmark (assignment or project deadline or exam).

6.3.3 Global Chain Model (GCM)

In this model we propose another way of incorporating the chain structure of a thread. Like the previous model, this model also assumes that posts belong to latent categories. It, however, doesn't model the instructor's intervention decision as a step in the thread generation process. Instead, it assumes that instructor's decision to intervene is dependent on all the posts in the threads, modeled using the latent post categories. This model is shown in Figure 6.2b. Assuming that p represents posts of thread t , h represents the latent category assignments, r represents the intervention

decision; feature vector, $\phi(\mathbf{p}, \mathbf{r}, \mathbf{h}, \mathbf{t})$, is extracted for each thread and using the weight vector, \mathbf{w} , this model defines a decision function, similar to what is shown in Equation 6.1.

Learning and Inference

Similar to the traditional maximum margin based Support Vector Machine (SVM) formulation, our model’s objective function is defined as:

$$\min_w \frac{\lambda}{2} \|\mathbf{w}\|^2 + \sum_j^T l(-r_j f_w(t_j, p_j)) \quad (6.2)$$

where λ is the regularization coefficient, t_j is the j^{th} thread with intervention decision r_j and p_j are the posts of this thread. \mathbf{w} is the weight vector, $l(\cdot)$ is the squared hinge loss function and $f_w(t_j, p_j)$ is defined in Equation 6.1.

Replacing the term $f_w(t_j, p_j)$ with the contents of Equation 6.1 in the minimization objective above, reveals the key difference from the traditional SVM formulation - the objective function has a maximum term inside the global minimization problem making it non-convex.

We, therefore, employ the optimization algorithm presented in [25] to solve this problem. Exploiting the semi-convexity property [45], the algorithm works in two steps, each executed iteratively. In the first step, it determines the latent variable assignments for positive examples. The algorithm then performs two steps iteratively - first it determines the structural assignments for the negative examples, and then optimizes the fixed objective function using a cutting plane algorithm. Once this process converges for negative examples, the algorithm reassigns values to the latent variables for positive examples, and proceeds to the second step. The

algorithm stops once a local minimum is reached. A somewhat similar approach, which uses the Convex-Concave Procedure (CCCP) is presented by [113].

At test time, given a thread, t , and its posts, p , we use the learned weights to compute $f_w(t, p)$ and classify it as belonging to the positive class (instructor intervenes) if $f_w(t, p) \geq 0$.

Feature Engineering

The feature set used by this model is very similar to the features used by the previous model. In addition to the non-structural features used by the logistic regression model (Section 6.3.1), it uses all the Post Emission features and the three transition features represented by $\phi(h_{i-1}, h_i)$ and $\phi(h_{i-1}, h_i, p_i, p_{i-1})$ as described in Section 6.3.2.

6.4 Data

For evaluating our models, we have used the forum content of two MOOCs from different domains (science and humanities), offered by *Coursera*, a leading education technology company.¹ Both courses were taught by professors from the University of Maryland, College Park.

Genes and the Human Condition (From Behavior to Biotechnology) (GHC)

dataset:² This course was attended by 30,000 students and the instructional staff comprised of 2 instructors, 3 Teaching Assistants and 56 technical support staff.

The discussion forum of this course consisted of 980 threads composed of about

¹<https://www.coursera.org/>

²<https://www.coursera.org/course/genes>

3,800 posts.

Women and the Civil Rights Movement (WCR) dataset:³ The course consisted of a classroom of about 14,600 students, 1 instructor, 6 Teaching Assistants and 49 support staff. Its discussion forum consisted of 800 threads and 3,900 posts.

We evaluate our models on held-out test sets. For the GHC dataset, the test set consisted of 186 threads out of which the instructor intervened on 24 while, for the WCR dataset, the instructor intervened on 21 out of 155 threads.

Also, it was commonly observed that after an instructor intervenes on a thread, its posting and/or viewing behavior increases. We, therefore, only consider the student posts until the instructor’s first intervention. Care was also taken to not use features that increased/decreased disproportionately because of the instructor’s intervention such as number of views or votes of a thread.

In our evaluation, we approximate instructor’s ‘should reply’ instances with those where the instructor indeed replied. Unlike general forum users, we believe that the correlation between the two scenarios is quite high for instructors. It is their responsibility to reply, and by choosing to a MOOC, they have ‘bought in’ to the idea of forum participation. The relatively smaller class sizes of these two MOOCs also ensured that most threads were manually reviewed, thus reducing instances of ‘missed’ threads while retaining the posting behavior and content of a typical MOOC.

³<https://www.coursera.org/course/womencivilrights>

Model	Genes and the Human Condition (GHC)			Women and the Civil Rights (WCR)		
	P	R	F	P	R	F
LR	44.44	16.67	24.24	66.67	15.38	25.00
J48	45.50	20.80	28.55	25.00	23.10	24.01
LCMM	33.33	29.17	31.11	42.86	23.08	30.00
GCM	60.00	25.00	35.29	50.00	18.52	27.03

Table 6.1: Held-out test set performances of chain models, LCMM and GCM, are better than that of the unstructured models, LR and J48.

6.5 Evaluation

6.5.1 Quantitative Results

Since the purpose of solving this problem is to identify the threads which should be brought to the notice of the instructors, we measure the performance of our models using F-measure of the positive class. The values of various parameters were selected using 10-fold Cross Validation on the training set. Table 6.1 presents the performances of the proposed models on the held-out test sets. We also report performance of a decision tree (J48) on the test sets for sake of comparison.

We can see that the chain based models, Linear Chain Markov Model (LCMM) and Global Chain Model (GCM), outperform the unstructured models, namely Logistic regression (LR) and Decision Trees (J48). This validates our hypothesis that

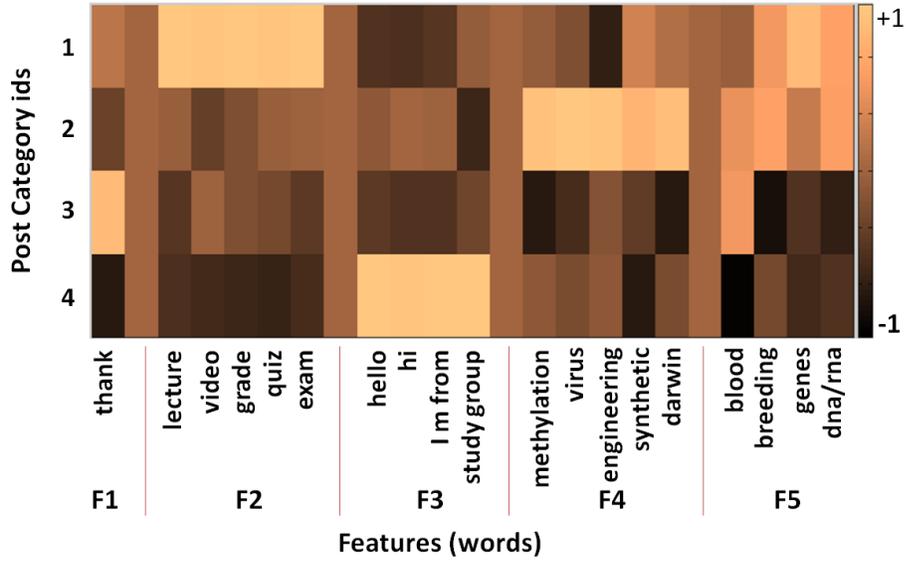


Figure 6.4: Visualization of lexical contents of the categories learnt by our model from the GHC dataset. Each row is a category and each column represents a feature vector. A light color represents high values while lower values are represented by darker shades. Dark monochromatic columns are used to better separate the five feature clusters, F1-F5, which represent words that are common in thanking, logistics-related, introductory, syllabus related and miscellaneous posts respectively. Categories 1,2,3 and 4 are dominated by F2, F4, F1 and F3 respectively indicating a semantic segregation of posts by our model’s categories.

using the post structure results in better modeling of instructor’s intervention.

The table also reveals that GCM yields high precision and low recall values, which is possibly due to the model being more conservative owing to information from all posts of the thread.

6.5.2 Visual Exploration of Categories

Our chain based models assume that posts belong to different (latent) categories and use these categories to make intervention predictions. Since this process of discovering categories is data driven, it would be interesting to examine the contents of these categories. Figure 6.4 presents a heat map of lexical content of categories identified by LCMM from the GHC dataset. The value of H (number of categories) was set to be 4 and was pre-determined during the model selection procedure. Each row of the heat map represents a category and the columns represent values of individual features, $f(w, c)$, defined as: $f(w, c) = \frac{C(w,c)}{\langle C(w,c) \rangle}$ where, $C(w, c)$ is total count of occurrences of a word, w , in all posts assigned to category, c and $\langle C(w, c) \rangle$ represents its expected count based on its frequency in the dataset.

While the actual size of vocabulary is huge, we use only a small subset of words in our feature vector for this visualization. These feature values, after normalization, are represented in the heat map using colors ranging from bright cream (high value) to dark black (low value). The darker the shade of a cell, the lower is the value represented by it.

For visual convenience, the features are manually clustered into five groups (F1 to F5) each separated by a dark beige colored column in the heat map. The first column of the heat map represents the F1 group which consists of words like *thank you, thanks, etc.* These words are characteristic of posts that mark either the conclusion of a resolved thread or are posted towards the end of the course. Rows corresponding to the category 3 in Table 6.2 show two examples of such

posts. Similarly, F2 represents the features related to logistics of the course and F3 captures introductory posts by new students. Finally, F4 contains words that are closely related to the subfield of gene and human conditions and would appear in posts that discuss specific aspects or chapters of the course contents, while F5 contains general buzz words that would appear frequently in any biology course.

Analyzing individual rows of the heat map, we can see that out of F1 to F4, Categories 1, 2, 3 and 4 are dominated by logistics (F2), course content related (F4), thank you (F1) and introductory posts (F3) respectively, represented by bright colors in their respective rows. We also observe similar correlations while examining the columns of the heat map. Also, F5, which contains words common to the gene and human health domain, is scattered across multiple categories. For example, *dna/rna* and *breeding* are sufficiently frequent in category 1 as well as 2.

Table 6.2 gives examples of representative posts from the four clusters. We show only the relevant part of the complete post. We can see that these examples agree with our observations from the heat map.

Furthermore, we compare the semantics of clusters learnt by our models with those proposed by Stump et al. [101] even though the two categorizations are not directly comparable. Nevertheless, generally speaking, our category 1 corresponds to Stump et al. [101]’s *Course structure/policies* and category 2 corresponds to *Content*. Interestingly, categories 3 and 4, which represent valedictory and introductory posts, correspond to a single *Social/affective* from the previous work.

We can, therefore, conclude that the model, indeed splits the posts into categories that look semantically coherent to the human eyes.

Category	Example posts
1	‘I’m having some issues with video playback. I have downloaded the videos to my laptop...’
	‘There was no mention of the nuclear envelope in the Week One lecture, yet it was in the quiz. Is this a mistake?’
2	‘DNA methylation is a crucial part of normal development of organisms and cell differentiation in higher organisms...’
	‘In the lecture, she said there are...I don’t see how tumor-suppressor genes are a cancer group mutation.’
3	‘Thank you very much for a most enjoyable and informative course.’
	‘Great glossary! Thank you!’
4	‘Hello everyone, I’m ... from the Netherlands. I’m a life science student.’
	‘Hi, my name is ... this is my third class with coursera’

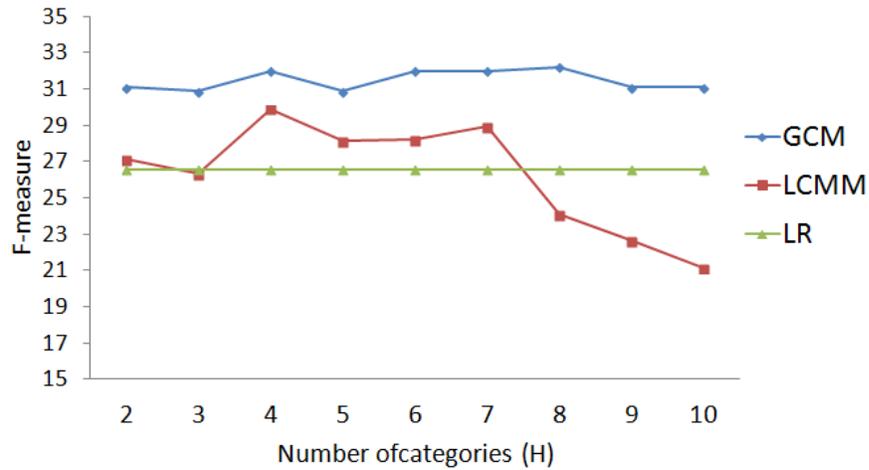
Table 6.2: Representative posts from the four categories learnt by our model. Due to space and privacy concerns we omit some parts of the text, indicated by “...”.

6.5.3 Choice of Number of Categories

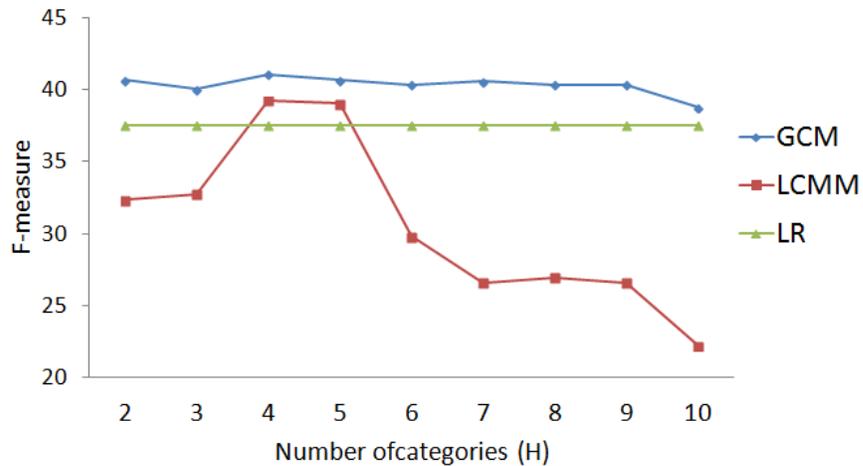
Our chain based models, assigning forum posts to latent categories, are parameterized with H , the number of categories. We therefore, study the sensitivity of our models to this parameter. Figure 6.5, plots the 10-fold cross validation performance of the models with increasing values of H for the two datasets. Interestingly, the sensitivity of the two models to the value of H is very different.

The LCMM model’s performance fluctuates as the value of H increases. The initial performance improvement might be due to an increase in the expressive power of the model. Performance peaks at $H = 4$ and then decreases, perhaps owing to over-fitting of the data.

In contrast, GCM performance remains steady for various values of H which might be attributed to the explicit regularization coefficient which helps combat over-fitting, by encouraging zero weights for unnecessary categories.

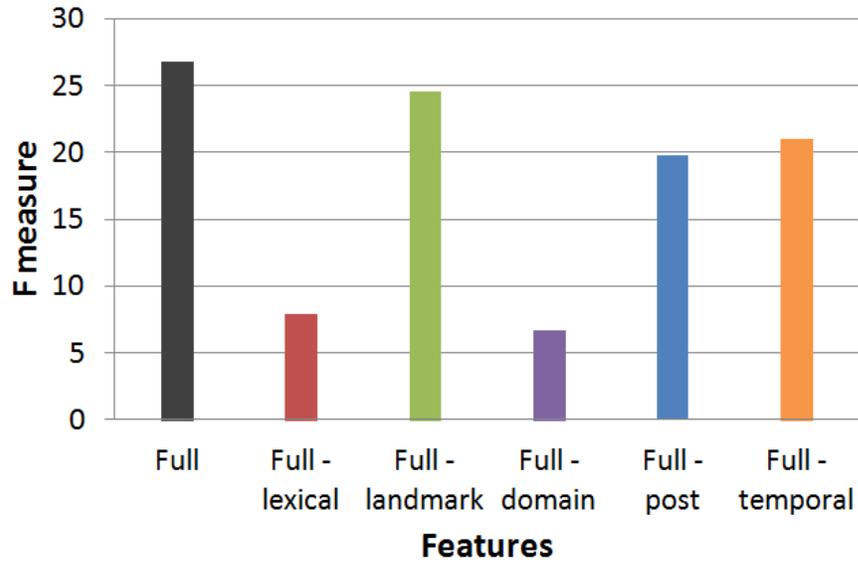


(a) Genes and the Human Condition dataset

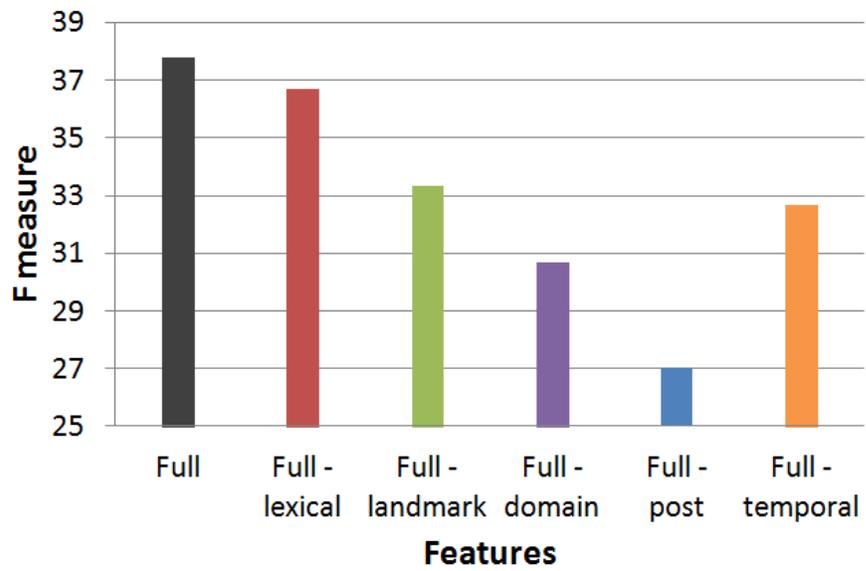


(b) Women and the Civil Rights Movement dataset

Figure 6.5: Cross validation performances of the two models with increasing number of categories.



(a) Genes and the Human Condition dataset



(b) Women and the Civil Rights Movement dataset

Figure 6.6: Cross validation performances of the various feature types for the two datasets.

6.5.4 How important are linguistic features?

We now focus on the structure independent features and experiment with their predictive value, according to types. We divide the features used by the LR into the following categories:⁴

- Full: set of all features (feature no. 1 to 15)
- lexical: based on content of thread titles and posts (feature no. 7 to 8 and 12 to 13)
- landmark: based on course landmarks (e.g, exams, quizzes) information (feature no. 11)
- MOOCs-specific: features specific to the MOOCs domain (lexical + landmark features)
- post: based only on aggregated posts information (feature no. 9 to 15)
- temporal: based on posting time patterns (feature no. 4, 5 and 10)

Figure 6.6 shows 10-fold cross validation F-measure of the positive class for LR when different types of features are excluded from the full set.

The figure reveals that the MOOCs-specific features (purple bar) are important for both the datasets indicating a need for designing specialized models for forums analysis in this domain.

⁴Please refer to Section 6.3.1 for description of the feature ids.

Also, lexical features (red bar) and post features (blue bar) have pretty dramatic effects in GHC and WCR data respectively.

Interestingly, removing the landmark feature set (green bar) causes a considerable drop in predictive performance, even though it consists of only one feature. Other temporal features (orange bar) also turn out to be important for the prediction. From a deeper analysis of feature values, we observed that instructors tend to get more active as the course progresses and their activity level also increases around quizzes/exams deadlines.

We can, therefore, conclude that all feature types are important and that lexical as well as MOOC specific analysis is necessary for modeling instructor's intervention.

6.6 Additional Application: Understanding Desire Fulfillment

We additionally apply the model proposed above to a different problem of identifying if a desire expressed by a subject in a given short piece of narrative text was fulfilled.

6.6.1 Problem Motivation and Definition

Understanding expressions of desire is a fundamental aspect of understanding intentional human-behavior. Desire expressions can be used to provide rationale for character behaviors when analyzing narrative text [53, 41], extract information about human wishes [51], explain positive and negative sentiment in reviews, and

DESIRE EXPRESSION: Before Lenin died, he said **he** wished to be buried beside his mother.
e1: When he died, Stalin let the people in Russia look at his body.
e2: Because people kept coming they decided not to bury him, and preserved his body instead.
e3: A building was built in Red Square, Moscow over the body so that people could see it.
e4: It is called the Lenin Mausoleum.
e5: Many Russians and tourists still go there to see his body today.
Desire Fulfillment status: Not fulfilled

Figure 6.7: Example consisting of: a Desire Expression, Evidence fragments (e1...e5) and a binary Desire Fulfillment Status. The Desire-subject is marked in bold fonts in the Desire-expression.

support automatic curation of community forums by identifying unresolved issues raised by users.

Given text, denoted as *Desire-expression* (e.g., “Before Lenin died, he said he wished to be buried beside his mother.”) containing a desire (“be buried beside his mother”) by the *Desire-subject* (“he”), and the subsequent text (denoted *Evidence fragments* or simply *Evidences*) appearing after the Desire-expression in the paragraph, we predict if the Desire-subject was successful in fulfilling their desire.

Figure 6.7 illustrates our setting.

6.6.2 Model: LCMM

To solve this problem we track the events and emotional states associated with the narrative’s central character (the *Desire-subject*) while modeling the narrative flow offered by the Evidence fragments. To model the narrative flow we associate

a latent state to each Evidence fragment indicating the model’s belief about the desire fulfillment status. We model the transitions between these states using the LCMM model presented above (Figure 6.2a), where the final latent state indicates the binary desire fulfillment status (as opposed to representing the instructor’s reply decision in the other task mentioned in this chapter). In other words, such an approach analyzes individual evidence fragments to extract information about desire fulfillment. However, while doing so it *views each evidence fragment in context of the other evidences*. The model is trained in a similar fashion.

6.6.3 Features

We now describe our features and how they are used by the model. Table 6.3 defines our features. They capture different semantic aspects of the desire-expression and evidences, such as entities, their actions and connotations, and their emotive states using lexical resources like Connotation Lexicon [46], WordNet and our lexicon of conforming and dissenting phrases. Before extracting features, we pre-processed the text to obtain POS tags, dependency parses, and resolved co-references using Stanford CoreNLP [73], and extracted all adjectives and verbs (with their negation statuses and connotations) associated with the Desire-subject.

1. Entailment (F1): This feature simply incorporates the *output of BIUTEE [99, 70] – a Textual Entailment (TE) model*. Since, TE systems often rely on aligning the entities appearing in the text fragments, we reduce the desire fulfillment task into several TE instances consisting of text-hypothesis pairs, by pairing the Desire-

Feature Type	Id	Definition
Entailment	F1	<i>TEPrediction</i> : Binary prediction of the Textual Entailment model [99].
Discourse	F2, F3	<i>ButPresent, SoPresent</i> : Binary features indicating if a ‘but’ or ‘so’ (respectively) followed the Desire-verb (‘wanted to’, ‘wished to’ etc.) in the Desire-expression.
Focal Word	F4, F5, F6	<i>focal count, focal syn and focal ant count</i> : Count of occurrences of the focal word(s), their WordNet [77] synonyms and antonyms (respectively) in the Evidence. Occurrences of synonyms or antonyms were identified only when they had the same POS tag as the focal word(s).
	F7	<i>focal+syn count</i> : Sum of F4 and F5
	F8	<i>focal lemm count</i> : Count of occurrences of lemmatized forms of the focal word(s) in the Evidence.
Desire-subject mention	F9	<i>sub count</i> : Count of all mentions (direct and co-referent) of the Desire-subject in the Evidence.
Emotional State	F10, F11	<i>+adj, -adj count</i> : Counts of occurrences of ‘positive’ and ‘negative’ adjectives (respectively) modifying the direct and co-referent mentions of the Desire-subject in the Evidence.
Action	F12, F13	<i>+Agent, -Agent count</i> : Number of times the connotation of verbs appearing in the Evidence agreed with and disagreed with (respectively) that of the intended action.
	F14, F15	<i>+Patient, -Patient count</i> : Count of occurrences of ‘positive’ and ‘negative’ verbs (respectively) in the Evidence which had the Desire-subject as the patient.
Sustenance	F16, F17	<i>isConforming, isDissenting</i> : Binary features indicating if the Evidence starts with a conforming or dissenting phrase (respectively) (see Appendix B).

Table 6.3: Feature definitions (Section 6.6.3). F1-F3 are extracted for each example while F4-F17 are extracted for each evidence.

Desire Expression: Amy wanted to **help** Ben, so she went with him.
 e1: She also **offered** him a ride.
 e2: Ben **thanked her** and **gifted her** a small present.
 e3: Overall, she was *happy*.
 Desire Fulfillment status: Positive (Fulfilled)

Figure 6.8: Artificial example indicating feature utility. The Desire-subject mentions are marked in blue, actions in bold and emotions in italics. Discourse feature is underlined.

expression (hypothesis) with each of the Evidence fragments (text) in that example. However, we “normalized” the Desire-expression, so that it would be directly applicable for the TE task and resemble the input over which such systems are trained. For example, the Desire-expression, “One day Jerry said *he* **wanted to** paint his barn.”, gets converted to “Jerry painted his barn.”. This process followed several steps:

- If the Desire-subject is pronominal, replace it with the appropriate named entity when possible (we used the Stanford CoreNLP coreference resolution system) [73].
- Ignore the content of the Desire-expression appearing before the Desire-subject.
- Remove the clause containing the Desire-verb (‘wanted to’, ‘wished to’ etc.), and convert the succeeding verb to its past tense.

The desire was considered ‘fulfilled’ if the TE model predicted *entailment* for at least one of the text-hypothesis pairs of the example.

2. Discourse (F2-F3): These features aim to identify *indications of obstacles*

or *progress* of desire fulfillment in the Desire-expression itself, based on discourse connectives. E.g. ‘so’ (underlined) in the Desire-expression in Figure. 6.8 indicates progress of desire fulfillment.

3. Focal words (F4-F8): These features identify the *word(s) most closely related* to the desire, and look for their presence in the Evidences. We define a *focal word* as the clausal complement of the Desire-verb (‘wanted to’, ‘hoped to’, ‘wished to’). If the clausal complement is a verb, the focal word is its past tense form. e.g., the focal word in the Desire expression in Figure. 6.8 is ‘helped’. A focal word is not simply the verb following the Desire-verb: e.g. in the Desire-expression in Figure. 6.7, the causal complement of ‘wished’ is ‘buried’. We then define features counting occurrences of the identified *focal words* and their WordNet *synonyms and antonyms* in each of the Evidences.

4. Desire-subject mentions (F9): This feature looks for mentions of Desire-subject in the Evidences assuming that a *lack of mentions* of the Subject might indicate absence of instances of their taking actions needed to fulfill the desire.

5. Emotional State (F10-F11): Signals about the fulfillment status could also emanate from the *emotional state of the Subject*. A happy or content Desire-subject can be indicative of a fulfilled desire (e.g. in Evidence e3 in Figure. 6.8), and vice versa. We quantify the emotional state of the Subject(s) using connotations of the adjectives modifying their mentions.

6. Action features (F12-F15): These features analyze the *intended* action and the actions taken by various entities. We first identify the *intended action* - the verb immediately following the Desire-verb in the Desire Expression. e.g., in Figure 6.8

the intended action is to ‘help’. Thereafter, we design features that capture the connotative agreement between the intended action and the actions taken by the Desire-subject(s) in the Evidences. We also include features that describe connotations of actions (verbs) affecting the Desire-subject(s). E.g. in e1 of Figure 6.8, the action by the Desire-subject (marked in blue), ‘offered’, is in connotative agreement with the intended action, ‘help’ (both have positive connotations according to Feng et al. [46]). Also, the actions affecting the subject (‘thanked’, ‘gifted’) have positive connotations indicating desire fulfillment.

7. Sustenance Features (F16-F17): LSNM uses a chain of latent states to abstractly represent the content of the Evidences with respect to Desire fulfillment Status. At any point in the chain, the model has an expectation of the fulfillment status. The sustenance features *indicate if the expectation should intensify, remain the same or be reversed by the incoming Evidence fragment*. This is achieved by designing features indicating if the Evidence fragment starts with a ‘conforming’ or a ‘dissenting’ phrase. E.g. e3 in Figure 6.8 starts with a conforming phrase, ‘Overall’, indicating that the fulfillment status expectation (positive in e2) should not change. These phrases were chosen using various discourse senses mentioned in Prasad et al. [88]. The complete list is available in Appendix B.

6.6.4 Evaluation

We evaluated our model on two datasets consisting of excerpts from (i) stories in the Machine Comprehension Test dataset (MCTest) [92] and (ii) the Simple En-

glish Wikipedia (SimpleWiki) ⁵. We compared this model with unstructured models that do not take the narrative flow into account. We also include the majority baseline, which always predicts the majority class. The majority classes for the MCTest and SimpleWiki datasets were the positive and the negative classes respectively. Table 6.4 shows our results. For our model, LCMM, we report median performance values over 100 random restarts, since its performance depends on the initialization of the weights. Also, the number of latent states was set to be 2 and 15 for the MCTest and SimpleWiki datasets respectively using cross-validation. The difference in optimal values for the number of latent states (and F scores) for the two datasets could be attributed to the difference in complexity of the language and concepts used in them. MCTest consists of children stories, focusing on simple concepts and goals (e.g., ‘wanting to go skating’) and their fulfillment is indicated explicitly, in simple and focused language (e.g., ‘They went to the skating rink together.’). On the other hand, SimpleWiki describes real-life desires (e.g., ‘wanting to conquer a country’), which require sophisticated planning over multiple steps, which may provide only indirect indication of the desire’s fulfillment. This resulted in a harder classification problem, and increased the complexity of inference over several latent states.

The table shows that LCMM outperforms the unstructured models indicating the benefit of incorporating the narrative structure of the storyline offered by the text by viewing individual evidences in context of other evidences to better understand desire fulfillment.

⁵http://simple.wikipedia.org/wiki/Main_Page

Data	Model Type	Model Name	P	R	F
MCTest	Unstructured	Majority	27.9	50.0	35.8
		LR	64.7	64.9	64.6
		DT	63.2	63.9	61.7
	Structured	LCMM	69.4	68.8	68.0
SimpleWiki	Unstructured	Majority	36.0	50.0	41.8
		LR	61.6	52.7	49.2
		DT	57.7	51.3	46.3
	Structured	LCMM	57.1	55.1	54.2

Table 6.4: Test set performances of various models reporting averaged Precision, Recall and F measure of the two classes. Our Latent Chain Markov Model, LCMM, outperforms the unstructured models (Logistic Regression, LR and Decision Trees, DT) and the majority baseline (which always predicts the majority class).

6.7 Conclusion

In this chapter we explored an application of the relationship identification task in the domain of MOOC discussion forums. A student’s question on the forum is viewed as a request to initiate a deeper relationship with the instructor, and the instructor’s reply is viewed as the initiation of the relationship. With this view, we address the task of initiation or existence of such instructor-student relationships. Specifically, we address the task of predicting instructor intervention in MOOC discussion forums. For this problem, we analyzed the text of the thread while viewing the individual posts in context of each other. We achieved this by presenting models that incorporate the structure of the thread using latent variables. Our experiments on forum data from two different Coursera MOOCs showed that utilizing thread structure is important for predicting instructors behavior.

We further applied one of our models to another task of identifying if a desire expressed in a short piece of narrative text got fulfilled. We extracted information from individual sentences in the text while viewing them in context of each other and empirically demonstrated the importance of incorporating this context and the utility of our model.

Chapter 7: Conclusion

7.1 Summary

In this dissertation we presented methods to model inter-personal relationships from text. We began with the problem of jointly inferring cooperative or non-cooperative relationships between characters of a narrative, and showed how the task of characterizing the relationship between two characters can benefit from incorporating information about their relationships with others.

We then addressed modeling evolving relationships as a sequence prediction task. For a given pair of characters, we identified a sequence of variables depicting the evolving relationship between the two characters. Each variable in the sequence was binary in nature and indicated a cooperative or a non-cooperative relationship status at a juncture in the narrative. We empirically demonstrated how this task could benefit from including historical dependencies.

In the following chapter, instead of specifying and restricting relationship types, we enabled our models to automatically discover various types of relationships in a data-driven manner. Some examples of relationship types discovered by our model include familial, adversarial, romantic, etc. Like the previous chapter, we modeled evolution of inter-personal relationships. We also demonstrated that

apart from incorporating historical information about the relationship of the two characters of interest, it is beneficial to maintain a global belief about the overall nature of their relationships.

Finally, we presented a practical application of this task in the domain of MOOC (Massive Open Online Courses) discussion forums. While viewing forums as platforms where their users (including students and instructors) interact and form relationships with each other, we focused on predicting the initiation of an involved relationship between students and instructors. In other words, we addressed the problem of analyzing forum threads to identify when an instructor would intervene on the thread. We showed that apart from incorporating various linguistic and domain-specific cues, it is important to model the discourse structure of discussion threads. We additionally demonstrated that such a model could also be used to address other problems, such as reading a piece of text to identify if the desire expressed in it was fulfilled.

An underlying theme that subsumes all of the above tasks is viewing these problems as structured predictions tasks that require incorporating linguistic cues as well as their contexts of appearance.

7.2 Future Work

The rest of this section discusses plausible directions for future work.

7.2.1 Domain-specific Models

The models presented in Chapters 3 to 5 could benefit from including domain-specific knowledge about narrative genres, author styles and preferences, etc. For example, while modeling inter-character relationships in Jane Austen’s novels, it could help to include the information that her novels are mostly romantic.

Similarly, while inferring inter-character relationships in narratives that are a part of a series (e.g. Harry Potter novels, Sherlock Holmes stories etc.), it might help to include a longer historical context or background knowledge about the relationships between the characters of interest in the previous narratives or the complete series.

7.2.2 Enhancing Latent Variable Models

In Chapter 5 we presented an unsupervised method that discovers various types of relationships in a data-driven manner. It assumes that sentences depicting relationships between two given characters belong to latent clusters, and each of these clusters represents a type of relationship. Such a model could benefit from a weak supervision, which can guide the cluster to discover the types of relationships we expect them to learn in a given dataset. One way to achieve this is by optionally ‘seeding’ the clusters with representative words. For example, while exploring a dataset of novels, including crime and romantic novels, one cluster could be seeded with words like ‘love’ and ‘marriage’ while another could be seeded with ‘kill’ and ‘attack’. While such a model will be more inclined to learn these two specific clus-

ters, it would still have the ability to discover unseeded clusters thus leveraging the advantages of unsupervised data-driven methods.

A similar idea could be applied to the models presented in Chapter 6. These models also assume that posts of a discussion forum thread belong to latent categories and learn these categories in a data-driven manner. These categories can also be seeded with domain-specific words or cues, enhancing their semantic coherence.

Another idea to improve the meaningfulness of the learned clusters is by enforcing the learned clusters to look diverse. This could be achieved by including an additional regularizer in the objective function of the models that aims at minimizing the inter-cluster similarity while maximizing the intra-cluster similarity. Such methods could be expected to improve the semantics of the learned clusters.

7.2.3 Applications to other NLP problems

This dissertation primarily focused on modeling inter-personal relationships. Apart from assisting in understanding people’s actions and goals in text, future work could study applications of the ideas presented in this dissertation to various domains. For example, we showed one such application in the domain of MOOC discussion forums. Another task, which can benefit from studying relationships, is that of automatically recommending recipient(s) for an email based on its text. Including information about the style or content of the email coupled with the (evolving) relationship of the composer with other his/her contacts (such as formal/informal, supervisor/subordinate, colleagues/family etc.), can help in obtaining novel per-

spectives and solutions. For example, an email discussing planning the user's son's birthday party is more likely to be meant for family members than office colleagues. Another use-case in this domain would be to curate the incoming emails/messages according to the recipients relationship with the sender. For example, a user might want all emails from family members to automatically get redirected to a dedicated folder. Similarly the user might want all emails from his/her supervisor to be automatically marked as important.

It is also possible to explore similar applications in related domains like instant messaging systems, social networking sites, etc. In the purview of recommendation systems, this research could apply to investigate the role of relationships in modeling individual preferences, and vice versa. For example, people in close relationships might be expected to reflect similar preferences in movies, books, music, etc. A more grounded use-case could be support or therapy groups. One could model the evolution of the relationship between the therapist and the patient from the content of their exchanges over social media, emails or text messages. This could assist therapists and moderators or administrators in analyzing if the trajectory of the relationship is constructive, or as intended.

7.2.4 Widening the definition of relationships

The work presented in this dissertation made several assumptions about relationships that can be relaxed in future work. For example, in the current work, we treat all relationships as symmetric. Future work could focus on studying asym-

metric relationships, such as unrequited love. Potential models operating with this definition could model relationships to be mutual in most cases but allow for asymmetric relationships in some cases.

The current work also doesn't allow simultaneous existence of multiple relationship-types. It assumes that at any point of time, there can be only one type of relationship between the participants. Future work could relax this criterion to allow for more nuanced modeling of relationships while treating this as a multi-label classification problem. For example, siblings could also be rivals, or a person's supervisor could also be his/her friend, etc. An interesting aspect of this work would be limit co-existence of certain relationship pairs. For example it is very unlikely for friends to also be mutually hostile. It would be interesting to learn such compatibilities or incompatibilities between labels using data-driven methods.

A related future direction could be to model relationship between characters as a mixture of several relationship-types. Such methods could use probabilistic models or mixture models to infer statements like a given pair of characters can be described as 90% friends and 10% family-members. Such models could be used to discover subtle relationships that are not explicitly stated in the text. For example, in a story two people might have a romantic attraction for each other but not demonstrate it obviously or might even exhibit a feeling of initial distaste towards each other. A human reader might have an inkling that the dislike might eventually lead to a more intimate relationship later. It would be interesting to see if such mixture models or probabilistic models could discover such hidden relationships by assigning a non-trivial weight/probability to the romantic relationship even when

the explicitly demonstrated relationship is of dislike.

7.2.5 Relevance to Digital Humanities

Research in digital humanities focuses on studying large collections of text. Our relationship identification models could be used to analyze news corpora and Wikipedia articles to study relationships and their evolution between political leaders and organizations (like political parties), notable individuals (like the pope, visionary scientists, etc.), and geo-political or business entities (like USA, UN, NATO), etc. Figure 7.1 shows example of a social network of political entities extracted from a Wikipedia article using the model presented in Chapter 3. In this network, each node represents a political entity and the edges labeled with + or - signs represent cooperative and non-cooperative relationships respectively. This example indicates a possible use-case of this work as a tool for analyzing digital archives and political data.

Our ideas could also be extended to analyze and discover patterns from large collections of literary works. For example our data-driven method in Chapter 5 discovered relationship states with subtle differences which can be difficult to discover manually (such as *Casual* and *Verbal* in Table 5.1), and can be of particular interest to literary scholars. The *Casual* state, depicting physical activity, might be a stronger indicator of beginning of a more intimate relationship than the *Verbal* state. Future work based on this research could also conduct large-scale studies aimed at answering literary questions like “Do certain authors or novels portray relationships

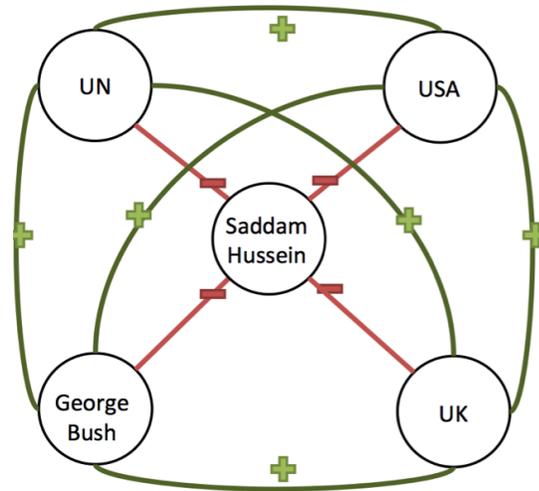


Figure 7.1: Network Inferred from the Wikipedia article on 2003 Invasion of Iraq. The + and - signs indicate cooperative and non-cooperative relationships respectively.

of desire more than others?” [87]; “Do Jane Austen’s female and male protagonists have a pattern in their evolving relationship (e.g. mutual disdain followed by romantic love)?” [19, 100, 57] etc.

7.2.6 Relevance to Psychological and Sociological theories

The ideas presented in this dissertation could also be extended to analyze theories related to inter-personal relationships. For example, Levinger [67] studied development of relationships, and suggested that relationships have a timeline or lifespan and several life-stages. For instance, romantic relationships have stages like Acquaintance, Buildup, Continuation, Deterioration, and Termination. In future, our models could be customized to study the various stages of certain types

of relationships. For example, while focusing only on cues related to romantic relationships, our models could analyze if fictional or real accounts of love stories indeed follow such timelines. Alternatively, they could attempt at exploring whether there is a connection between success/popularity of such narratives and their proximity to such theoretical timelines.

Thus, the current work suggests several interesting avenues for improving and extending existing models, as well as using them to solve other interesting problems from domains ranging from NLP to Psychology. Improved methods for solving these problems could also focus on enriched models of character attributes, goals and intentions, as well grounding character personae in particular narratives to entities in a knowledge base. Such joint models could benefit from treating text as a reflection of social phenomenon, and incorporating world knowledge from multiple sources to assist in its comprehension.

Appendix A: Frames Lexicon

Lists of frames used by our Semantic-parse based features. ‘Frames’ and ‘Frame-elements’ used in the following table refer to those fired during frame-semantic parsing of the text using Semafor [33].

Type	Frame	Frame-elements
Negative	abusing	victim, abuser
	attack	assailant, victim
	avoiding	agent, undesirable_situation
	besieging	assailant, victim
	cause_emotion	agent, experiencer
	cause_harm	agent, victim
	competition	participant_1, participant_2, participants
	defending	assailant, victim
	destroying	destroyer, undergoer
	endangering	agent, valued_entity
	experience_bodily_harm	experiencer, body_part

Type	Frame	Frame-elements
Negative	fall_for	deception, victim
	fear	stimulus, experiencer
	firing	employer, employee
	giving_in	compeller, capitulator
	going_back_on_a_commitment	protagonist, affected_party
	hindering	protagonist, action
	hit_target	agent, target
	hostile_encounter	side_1, side_2, sides
	immobilization	agent, patient
	inhibit_movement	agent, theme
	intentional_deception	deceiver, victim
	kidnapping	perpetrator, victim, co-participant
	killing	killer, victim
	manipulate_into_doing	manipulator, victim
	offenses	perpetrator, victim
	piracy	perpetrator, victim
	prevent_from_having	agent, protagonist
	protecting	danger, asset
	quarreling	arguer1, arguer2, arguers

Type	Frame	Frame-elements
Negative	rape	perpetrator, victim
	revenge	avenger, offender, injured_party
	taking_captive	agent, captive
	thwarting	protagonist, preventing_cause
	trap	deceiver, victim
	violence	aggressor, aggressors, victim
	want_suspect	suspect

Table A.1: Lists of negative frames.

Type	Frame	Frame-elements
Positive	alliance	member_1, member_2, alliance
	be_in_agreement_on_assessment	cognizer_1, cognizer_1, cognizers
	collaboration	partner_1, partner_2, partners
	chatting	interlocutor_1, interlocutor_2
	choosing	cognizer, chosen
	come_together	configuration, individuals, party_1, party_2
	commitment	speaker, addressee
	commonality	entity_1, entity_2, entities
	defending	defender, victim

Type	Frame	Frame-elements
Positive	desiring	experiencer, focal_participant
	emotion_active	experiencer, topic
	personal_relationship	partner_1, partner_2, partners
	forgiveness	judge, evaluatee
	forming_relationships	partner_1, partner_2, partners
	grooming	agent, patient
	hospitality	host, guest
	make_agreement_on_action	party_1, party_2, parties
	offering	offerer, potential_recipient
	participation	participant_1, participant_2, participants
	protecting	protection, asset
	reassuring	speaker, experiencer
	releasing	captor, theme
	reparation	wrongdoer, wrongdoer
	rescuing	agent, asset, patient
	reveal_secret	speaker, addressee
	sharing	protagonist_1, protagonist_2, protagonists
	sign_agreement	signatory, co-participant
	social_connection	individual_1, individual_2, individuals

Type	Frame	Frame-elements
Positive	social_event	attendee
	social_event_collective	attendees
	social_event_individuals	party_1, party_2
	suasion	speaker, addressee
	subjective_influence	agent, cognizer
	supply	supplier, recipient
	supporting	supporter, supported
	visiting	agent, entity
	warning	speaker, addressee

Table A.2: Lists of positive frames.

Type	Frame	Frame-elements
Ambiguous	cause_bodily_experience	agent, experiencer
	cause_to_experience	agent, experiencer
	cotheme	theme, cotheme
	experiencer_focus	experiencer
	friendly_or_hostile	side_1, side_2, sides
	manipulation	agent, entity
	respond_to_proposal	interlocutor, speaker

Table A.3: Lists of ambiguous frames.

Type	Frame	Frame-elements
Relationship	kinship	alter, ego, relatives
	forming_relationships	partner_1, partner_2
	personal_relationship	partner_1, partner_2
	subordinates_and_superiors	superior, subordinate

Table A.4: Lists of relationship frames.

Appendix B: Conformity Lexicon

Our sustenance features described in Section 6.6.3 use a list of conforming and dissenting phrases. These phrases were chosen manually using various discourse senses mentioned in Prasad et al. [88].

For conforming phrase list we considered the explicit connectives in the following discourse senses: ‘Contra-expectation’, ‘Contrast’, ‘Contrast/Precedence’, ‘Contrast/Synchrony’, ‘Contrast/Temporal’, ‘Opposition’ but discarded very frequent phrases like ‘and’, ‘if’, ‘or’, ‘then’, ‘when’, ‘while’, ‘as if’, ‘in the end’.

Similarly, for the dissenting phrase list, we considered the following senses: ‘Equivalence’, ‘Instantiation’, ‘Precedence/Result’, ‘Reason’, ‘Reason/Restatement’, ‘Reason/Synchrony’, ‘Result’, ‘Specification’, ‘Specification/Succession’, ‘Specification/Synchrony’ and discarded ‘and’, ‘as’, ‘or’, ‘for’, ‘then’, ‘when’, ‘rather’, ‘in turn’, ‘but’, ‘if only’.

The complete list of these phrases is shown in Table B.1.

Type	Phrases
Conforming	‘in other words’, ‘indeed’, ‘for example’, ‘for instance’, ‘in fact’, ‘in particular’, ‘finally’, ‘ultimately’, ‘apparently because’, ‘at least partly because’, ‘because’, ‘especially as’, ‘especially because’, ‘especially since’, ‘in large part because’, ‘in part because’, ‘insofar as’, ‘just because’, ‘largely because’, ‘mainly because’, ‘merely because’, ‘not because’, ‘not only because’, ‘now that’, ‘only because’, ‘particularly as’, ‘particularly because’, ‘particularly since’, ‘partly because’, ‘perhaps because’, ‘presumably because’, ‘primarily because’, ‘simply because’, ‘since’, ‘so’, ‘accordingly’, ‘as a result’, ‘consequently’, ‘hence’, ‘in the end’, ‘in turn’, ‘largely as a result’, ‘so that’, ‘thereby’, ‘therefore’, ‘thus’, ‘also’, ‘as though’, ‘much as’, ‘overall’, ‘specifically’, ‘especially after’, ‘especially when’
Dissenting	‘although’, ‘but’, ‘by comparison’, ‘by contrast’, ‘conversely’, ‘even though’, ‘however’, ‘in contrast’, ‘in fact’, ‘instead’, ‘rather’, ‘nevertheless’, ‘nonetheless’, ‘nor’, ‘on the contrary’, ‘on the other hand’, ‘meanwhile’, ‘still’, ‘though’, ‘whereas’, ‘yet’, ‘even as’, ‘even if’, ‘even still’, ‘even then’, ‘regardless’, ‘neither’

Table B.1: Lists of conforming and dissenting phrases used by the Sustenance features.

Bibliography

- [1] Elizabeth Abel. (E)Merging Identities: The Dynamics of Female Friendship in Contemporary Fiction by Women. *Signs*, 6(3):413–435, 1981. ISSN 00979740, 15456943. URL <http://www.jstor.org/stable/3173754>.
- [2] Apoorv Agarwal, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. SIN-NET: Social Interaction Network Extractor from Text. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 33–36, 2013. URL <http://aclweb.org/anthology/I/I13/I13-2009.pdf>.
- [3] Richard D Alexander. The Evolution of Social Behavior. *Annual review of ecology and systematics*, pages 325–383, 1974.
- [4] Irwin Altman and Dalmas A Taylor. *Social Penetration: The Development of Interpersonal Relationships*. Holt, Rinehart & Winston, 1973.
- [5] Ashton Anderson, Daniel P. Huttenlocher, Jon M. Kleinberg, and Jure Leskovec. Engaging with Massive Online Courses. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014*, pages 687–698, 2014. doi: 10.1145/2566486.2568042. URL <http://doi.acm.org/10.1145/2566486.2568042>.
- [6] Yoav Artzi, Patrick Pantel, and Michael Gamon. Predicting Responses to Microblog Posts. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT '12*, pages 602–606, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. ISBN 978-1-937284-20-6. URL <http://dl.acm.org/citation.cfm?id=2382029.2382126>.
- [7] Lars Backstrom, Jon Kleinberg, Lillian Lee, and Cristian Danescu-Niculescu-Mizil. Characterizing and Curating Conversation Threads: Expansion, Focus, Volume, Re-entry. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM '13*, pages 13–22, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1869-3. doi: 10.1145/2433396.2433401. URL <http://doi.acm.org/10.1145/2433396.2433401>.

- [8] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL'98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference.*, pages 86–90, 1998. URL <http://aclweb.org/anthology/P/P98/P98-1013.pdf>.
- [9] David Bamman. *People-Centric Natural Language Processing*. PhD thesis, Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 2015.
- [10] David Bamman and Noah Smith. Unsupervised Discovery of Biographical Structure from Text. *Transactions of the Association for Computational Linguistics*, 2:363–376, 2014. ISSN 2307-387X. URL <https://tacl2013.columbia.edu/ojs/index.php/tacl/article/view/371>.
- [11] 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, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 352–361, 2013. URL <http://aclweb.org/anthology/P/P13/P13-1035.pdf>.
- [12] David Bamman, Ted Underwood, and Noah A. Smith. A Bayesian Mixed Effects Model of Literary Character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 370–379, 2014. URL <http://aclweb.org/anthology/P/P14/P14-1035.pdf>.
- [13] Leonard E. Baum and Ted Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.
- [14] Eric Berne. *Games People Play: The Psychology of Human Relationships*, volume 2791. Penguin UK, 1967.
- [15] Lynda E. Boose. The Father and the Bride in Shakespeare. *PMLA*, 97(3):325–347, 1982. ISSN 00308129. URL <http://www.jstor.org/stable/462226>.
- [16] P. Oscar Boykin and Vwani P. Roychowdhury. Personal Email Networks: An Effective Anti-Spam Tool. *CoRR*, cond-mat/0402143, 2004. URL <http://arxiv.org/abs/cond-mat/0402143>.
- [17] 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-Volume 1*, pages 773–782. Association for Computational Linguistics, 2011.

- [18] Jerome Bruner. *Actual Minds. Possible Worlds*. Harvard University Press, Cambridge, 1986.
- [19] Marilyn Butler. *Jane Austen and the War of Ideas*. Clarendon Press, 1975. ISBN 9780198120681. URL <https://books.google.com/books?id=1CvuAAAAMAAJ>.
- [20] Dorwin Cartwright and Frank Harary. Structure Balance: A Generalization of Heiders Theory. *Psychological review*, 63(5), 1956.
- [21] Nathanael Chambers. Event Schema Induction with a Probabilistic Entity-Driven Model. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1797–1807, 2013. URL <http://aclweb.org/anthology/D/D13/D13-1185.pdf>.
- [22] Nathanael Chambers and Dan Jurafsky. Unsupervised Learning of Narrative Schemas and their Participants. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 602–610, 2009. URL <http://www.aclweb.org/anthology/P09-1068>.
- [23] Nathanael Chambers and Daniel Jurafsky. Unsupervised Learning of Narrative Event Chains. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 789–797, 2008. URL <http://www.aclweb.org/anthology/P08-1090>.
- [24] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In *Neural Information Processing Systems*, 2009.
- [25] Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. Discriminative Learning over Constrained Latent Representations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, HLT '10*, pages 429–437, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL <http://dl.acm.org/citation.cfm?id=1857999.1858065>.
- [26] Snigdha Chaturvedi, Dan Goldwasser, and Hal Daumé III. Predicting Instructor’s Intervention in MOOC forums. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1501–1511. Association for Computational Linguistics, 2014. URL <http://aclweb.org/anthology/P14-1141>.

- [27] Snigdha Chaturvedi, Dan Goldwasser, and Hal Daumé III. *Ask, and shall you receive?: Understanding Desire Fulfillment in Natural Language Text*. In *Proceedings of the Thirtieth Conference on Artificial Intelligence, AAAI'16*, 2016.
- [28] Snigdha Chaturvedi, Shashank Srivastava, Hal Daumé III, and Chris Dyer. Modeling Evolving Relationships between Characters in Literary Novels. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16*, 2016.
- [29] Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. Probabilistic Frame Induction. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 837–846, 2013. URL <http://aclweb.org/anthology/N13/N13-1104.pdf>.
- [30] Robert Coles. *The Call of Stories: Teaching and the Moral Imagination*. Houghton Mifflin, Boston, MA, 1989.
- [31] Michael Collins. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02*, pages 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1118693.1118694. URL <http://dx.doi.org/10.3115/1118693.1118694>.
- [32] Cristian Danescu-Niculescu-Mizil, Lillian Lee, Bo Pang, and Jon M. Kleinberg. Echoes of Power: Language Effects and Power Differences in Social Interaction. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 699–708, 2012. doi: 10.1145/2187836.2187931. URL <http://doi.acm.org/10.1145/2187836.2187931>.
- [33] Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. Frame-Semantic Parsing. *Computational Linguistics*, 40(1): 9–56, 2014. doi: 10.1162/COLI_a_00163. URL http://dx.doi.org/10.1162/COLI_a_00163.
- [34] James A. Davis. Clustering and Structural Balance in Graphs. *Clustering and structural balance in graphs*, 20(2):181–187, 1967.
- [35] Munmun De Choudhury, Hari Sundaram, Ajita John, and Dorée Duncan Seligmann. What Makes Conversations Interesting? Themes, Participants and Consequences of Conversations in Online Social Media. In *18th International World Wide Web Conference (WWW)*, pages 331–331, April 2009. URL <http://www2009.eprints.org/34/>.

- [36] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
- [37] Mark Edmundson. The Trouble With Online Education, July 19 2012. <http://www.nytimes.com/2012/07/20/opinion/the-trouble-with-online-education.html>.
- [38] Lajos Egri. *The Art of Dramatic Writing*. Simon and Schuster, New York, NY, USA, 1946.
- [39] Khalid El-Arini, Min Xu, Emily B. Fox, and Carlos Guestrin. Representing Documents Through Their Readers. In Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy, editors, *KDD*, pages 14–22. ACM, 2013. ISBN 978-1-4503-2174-7. URL <http://dblp.uni-trier.de/db/conf/kdd/kdd2013.html#El-AriniXFG13>.
- [40] David K. Elson. *Modeling Narrative Discourse*. PhD thesis, Columbia University, 2012.
- [41] David K Elson. Detecting Story Analogies from Annotations of Time, Action and Agency. In *Proceedings of the LREC 2012 Workshop on Computational Models of Narrative, Istanbul, Turkey*, 2012.
- [42] David K. Elson, Nicholas Dames, and Kathleen McKeown. Extracting Social Networks from Literary Fiction. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 138–147, 2010. URL <http://www.aclweb.org/anthology/P10-1015>.
- [43] Susan Engle. *The Stories Children Tell: Making Sense of the Narratives of Childhood*. W.H. Freeman & Company, NY, 1995.
- [44] Ursula Fanning. Sentimental Subversion: Representations of Female Friendship in the Work of Matilde Serao. *Annali d’Italianistica*, 7:273–286, 1989. ISSN 07417527. URL <http://www.jstor.org/stable/24003871>.
- [45] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. ISSN 0162-8828. doi: <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2009.167>.
- [46] Song Feng, Jun Seok Kang, Polina Kuznetsova, and Yejin Choi. Connotation Lexicon: A Dash of Sentiment Beneath the Surface Meaning. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages

- 1774–1784, 2013. URL <http://aclweb.org/anthology/P/P13/P13-1174.pdf>.
- [47] Mark Alan Finlayson. *Learning Narrative Structure from Annotated Folktales*. PhD thesis, Massachusetts Institute of Technology, 2012.
- [48] Jane M. Ford. *Patriarchy and Incest from Shakespeare to Joyce*. University Press of Florida, 1998.
- [49] Gillian Frith. *The Intimacy which is Knowledge: Female Friendships in the Novels of Women Writers*. PhD thesis, University of Warwick, December 1988.
- [50] Jennifer Golbeck and James A. Hendler. Reputation Network Analysis for Email Filtering. In *CEAS 2004 - First Conference on Email and Anti-Spam, July 30-31, 2004, Mountain View, California, USA*, 2004. URL <http://www.ceas.cc/papers-2004/177.pdf>.
- [51] Andrew B Goldberg, Nathanael Fillmore, David Andrzejewski, Zhiting Xu, Bryan Gibson, and Xiaojin Zhu. May All Your Wishes Come True: A Study of Wishes and How to Recognize Them. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 263–271. Association for Computational Linguistics, 2009.
- [52] Debra Goodlett. Love and Addiction in ‘Wuthering Heights’. *The Review of English Studies*, 37(3), 1996.
- [53] Amit Goyal, Ellen Riloff, and Hal Daumé III. Automatically Producing Plot Unit Representations for Narrative Text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 77–86, 2010. URL <http://www.aclweb.org/anthology/D10-1008>.
- [54] Anatoliy Gruzd. Automated Discovery of Social Networks in Text-based Online Communities. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work, GROUP '09*, pages 379–380, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-500-0. doi: 10.1145/1531674.1531733. URL <http://doi.acm.org/10.1145/1531674.1531733>.
- [55] Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. Extracting Signed Social Networks from Text. In *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing*, pages 6–14. Association for Computational Linguistics, 2012.
- [56] Fritz Heider. Attitudes and Cognitive Organization. *Journal of Psychology*, 21:107–112, 1946.

- [57] Charles H. Hinant. Jane Austen’s “Wild Imagination”: Romance and the Courtship Plot in the Six Canonical Novels. *Narrative*, 14(3):294–310, 2006. URL <http://www.jstor.org/stable/20107392>.
- [58] Jonathan Huang, Chris Piech, Andy Nguyen, and Leonidas J. Guibas. Syntactic and Functional Variability of a Million Code Submissions in a Machine Learning MOOC. In *AIED Workshops*, 2013.
- [59] Liang Huang, Suphan Fayong, and Yang Guo. Structured Perceptron with Inexact Search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–151. Association for Computational Linguistics, 2012.
- [60] Lynn Jamieson and Lynn Jameison. *Intimacy*. Polity Press Cambridge, 1998.
- [61] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane Training of Structural SVMs. *Machine Learning*, 77(1):27–59, 2009. doi: 10.1007/s10994-009-5108-8. URL <http://dx.doi.org/10.1007/s10994-009-5108-8>.
- [62] Paramjit S. Judge. *Foundations of Classical Sociological Theory: Functionalism, Conflict and Action*. Pearson, 1 edition, March 2012. ISBN 0898591384.
- [63] Jon M. Kleinberg. Computational Perspectives on Social Phenomena at Global Scales. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI, 2013. ISBN 978-1-57735-633-2. URL <http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6896>.
- [64] Vinodh Krishnan and Jacob Eisenstein. “You’re Mr. Lebowsky, I’m the Dude”: Inducing Address Term Formality in Signed Social Networks. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1616–1626, 2015. URL <http://aclweb.org/anthology/N/N15/N15-1185.pdf>.
- [65] Wendy G. Lehnert. Plot Units and Narrative Summarization. *Cognitive Science*, 5(4):293–331, 1981. doi: 10.1207/s15516709cog0504_1. URL http://dx.doi.org/10.1207/s15516709cog0504_1.
- [66] Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. Signed Networks in Social Media. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA, April 10-15, 2010*, pages 1361–1370, 2010. doi: 10.1145/1753326.1753532. URL <http://doi.acm.org/10.1145/1753326.1753532>.
- [67] G Levinger. Development and Change. *H.H. Kelley, et al. (Eds.), Close relationships*, pages 315–359, 1983.

- [68] Chu-Cheng Lin, Waleed Ammar, Chris Dyer, and Lori Levin. Unsupervised POS Induction with Word Embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1311–1316, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/N15-1144>.
- [69] Bing Liu, Mingqing Hu, and Junsheng Cheng. Opinion Observer: Analyzing and Comparing Opinions on the Web. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, pages 342–351, 2005. doi: 10.1145/1060745.1060797. URL <http://doi.acm.org/10.1145/1060745.1060797>.
- [70] Bernardo Magnini, Roberto Zanolli, Ido Dagan, Kathrin Eichler, Guenter Neumann, Tae-Gil Noh, Sebastian Padó, Asher Stern, and Omer Levy. The excitement open platform for textual inferences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 43–48, 2014. URL <http://aclweb.org/anthology/P/P14/P14-5008.pdf>.
- [71] Inderjeet Mani. Computational Narratology. *Hühn, Peter et al. (eds.): the living handbook of narratology*, 2012. URL hup.sub.uni-hamburg.de/lhn/index.php?title=ComputationalNarratology&oldid=2030.
- [72] Inderjeet Mani. *Computational Modeling of Narrative*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012. ISBN 9781608459810. doi: 10.2200/S00459ED1V01Y201212HLT018. URL <http://dx.doi.org/10.2200/S00459ED1V01Y201212HLT018>.
- [73] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*, pages 55–60, 2014. URL <http://aclweb.org/anthology/P/P14/P14-5010.pdf>.
- [74] Philip Massey, Patrick Xia, David Bamman, and Noah Smith. Annotating Character Relationships in Literary Texts. *ArXiv e-prints*, 2015.
- [75] Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. Topic and Role Discovery in Social Networks with Experiments on Enron and Academic Email. *Journal of Artificial Intelligence Research (JAIR)*, 30:249–272, 2007. doi: 10.1613/jair.2229. URL <http://dx.doi.org/10.1613/jair.2229>.
- [76] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger,

- editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. URL <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositional.pdf>.
- [77] George A. Miller. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL <http://doi.acm.org/10.1145/219717.219748>.
- [78] Jeff Mitchell and Mirella Lapata. Vector-based Models of Semantic Composition. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 236–244, 2008. URL <http://www.aclweb.org/anthology/P08-1028>.
- [79] Raymond J. Mooney and Gerald DeJong. Learning Schemata for Natural Language Processing. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence. Los Angeles, CA, August 1985*, pages 681–687, 1985.
- [80] F. Mosteller and D. L. Wallace. *Inference and Disputed Authorship: The Federalist Papers*. Addison-Wesley, Reading, Mass., 1964.
- [81] Kevin Murphy. Hidden Markov Model (HMM) Toolbox for Matlab. <http://people.cs.ubc.ca/~murphyk/Software/HMM/hmm.html>, 1998.
- [82] Andy Nguyen, Christopher Piech, Jonathan Huang, and Leonidas J. Guibas. Codewebs: Scalable Homework Search for Massive Open Online Programming Courses. In *WWW*, pages 491–502, 2014.
- [83] John Walker Orr, Prasad Tadepalli, Janardhan Rao Doppa, Xiaoli Fern, and Thomas G. Dietterich. Learning Scripts as Hidden Markov Models. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 1565–1571, 2014. URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8603>.
- [84] Karl Pichotta and Raymond J. Mooney. Learning Statistical Scripts with LSTM Recurrent Neural Networks. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, Phoenix, Arizona, February 2016. URL <http://www.cs.utexas.edu/users/ai-lab/pub-view.php?PubID=127543>.
- [85] Chris Piech, Jonathan Huang, Zhenghao Chen, Chuong Do, Andrew Ng, and Daphne Koller. Tuned Models of Peer Assessment in MOOCs. In *Proceedings of The 6th International Conference on Educational Data Mining (EDM 2013)*, 2013.
- [86] Plato and W. Hamilton. *The Symposium*. Penguin classics. Penguin Books, 1951. URL <https://books.google.com/books?id=1fIDAQAIAAJ>.

- [87] Robert M. Polhemus. *Being in Love from Jane Austen to DH Lawrence*. Chicago: University of Chicago Press, 1990.
- [88] Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, Livio Robaldo, and Bonnie Webber. The Penn Discourse Tree-bank 2.0 Annotation Manual. Technical report, University of Pennsylvania, Institute for Research in Cognitive Science, 2007. URL <http://www.seas.upenn.edu/~pdtb/PDTBAPI/pdtb-annotation-manual.pdf>.
- [89] Vladimir Iakovlevich Propp. *Morphology of the Folktale*. University of Texas, 1968. URL <http://books.google.com/books?isbn=0292783760>.
- [90] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daumé III, and Lise Getoor. Learning Latent Engagement Patterns of Students in Online Courses. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 1272–1278, 2014. URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8571>.
- [91] Michaela Regneri, Alexander Koller, and Manfred Pinkal. Learning Script Knowledge with Web Experiments. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 979–988, 2010. URL <http://www.aclweb.org/anthology/P10-1100>.
- [92] Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In *EMNLP*, pages 193–203. ACL, 2013. ISBN 978-1-937284-97-8. URL <http://dblp.uni-trier.de/db/conf/emnlp/emnlp2013.html#RichardsonBR13>.
- [93] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The Author-topic Model for Authors and Documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, UAI '04*, pages 487–494, Arlington, Virginia, United States, 2004. AUAI Press. ISBN 0-9749039-0-6. URL <http://dl.acm.org/citation.cfm?id=1036843.1036902>.
- [94] Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures (Artificial Intelligence Series)*. Psychology Press, 1 edition, July 1977. ISBN 0898591384.
- [95] T. Shinozaki and T. Kawahara. HMM Training based on CV-EM and CV Gaussian Mixture Optimization. In *Automatic Speech Recognition Understanding, 2007. ASRU. IEEE Workshop on*, pages 318–322, Dec 2007. doi: 10.1109/ASRU.2007.4430131.
- [96] Noah A. Smith. *Linguistic Structure Prediction*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2011. doi: 10.2200/S00361ED1V01Y201105HLT013. URL <http://dx.doi.org/10.2200/S00361ED1V01Y201105HLT013>.

- [97] Shashank Srivastava, Snigdha Chaturvedi, and Tom Mitchell. Inferring Interpersonal Relations in Narrative Summaries. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, 2016.
- [98] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. BRAT: a Web-based Tool for NLP-assisted Text Annotation. In *Proceedings of Demonstrations Session at EACL 2012*, pages 102–107, 2012.
- [99] Asher Stern and Ido Dagan. BIUTEE: A Modular Open-Source System for Recognizing Textual Entailment. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 73–78, 2012. URL <http://www.aclweb.org/anthology/P12-3013>.
- [100] Bruce Stovel. 'A Contrariety of Emotion': Jane Austen's Ambivalent Lovers in *Pride and Prejudice*. *The International Fiction Review*, 14(1):27–33, 1987.
- [101] Glenda S. Stump, Jennifer DeBoer, Jonathan Whittinghill, and Lori Breslow. Development of a Framework to Classify MOOC Discussion Forum Posts: Methodology and Challenges. In *NIPS Workshop on Data Driven Education*, 2013.
- [102] Xu Sun, Takuya Matsuzaki, Daisuke Okanohara, and Jun'ichi Tsujii. Latent Variable Perceptron Algorithm for Structured Classification. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1236–1242, 2009. URL <http://ijcai.org/papers09/Papers/IJCAI09-208.pdf>.
- [103] Oscar Täckström and Ryan McDonald. Discovering Fine-grained Sentiment with Latent Variable Structured Prediction Models. In *Proceedings of the 33rd European Conference on Advances in Information Retrieval, ECIR'11*, pages 368–374, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-20160-8. URL <http://dl.acm.org/citation.cfm?id=1996889.1996937>.
- [104] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large Margin Methods for Structured and Interdependent Output Variables. *Journal of Machine Learning Research*, 6:1453–1484, December 2005. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1046920.1088722>.
- [105] Josep Valls-Vargas, Jichen Zhu, and Santiago Ontañón. Toward Automatic Role Identification in Unannotated Folk Tales. In *Proceedings of the Tenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2014, October 3-7, 2014, North Carolina State University, Raleigh, NC, USA*, 2014. URL <http://www.aaai.org/ocs/index.php/AIIDE/AIIDE14/paper/view/9004>.

- [106] Max Weber, Guenther Roth, and Claus Wittich. *Economy and Society: An Outline of Interpretive Sociology*. University of California Press, 1978. ISBN 9780520035003. URL <https://books.google.com/books?id=pSdaNuIaUUEC>.
- [107] Robert West, Hristo S. Paskov, Jure Leskovec, and Christopher Potts. Exploiting Social Network Structure for Person-to-Person Sentiment Analysis. *TACL*, 2:297–310, 2014. URL <https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/396>.
- [108] R. S. White. Metamorphosis by Love in Elizabethan Romance, Romantic Comedy, and Shakespeare’s Early Comedies. *The Review of English Studies*, 35(137):14–44, 1984. ISSN 00346551, 14716968. URL <http://www.jstor.org/stable/516810>.
- [109] Robert Wilensky. *Understanding Goal-Based Stories*. Outstanding Dissertations in the Computer Sciences. Garland Publishing, New York, 1978. ISBN 0-8240-4410-X.
- [110] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada*, 2005. URL <http://acl.ldc.upenn.edu/H/H05/H05-1044.pdf>.
- [111] Diyi Yang, David Adamson, and Carolyn Penstein Rosé. Question Recommendation with Constraints for Massive Open Online Courses. In *Proceedings of the 8th ACM Conference on Recommender Systems, RecSys ’14*, pages 49–56, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2668-1. doi: 10.1145/2645710.2645748. URL <http://doi.acm.org/10.1145/2645710.2645748>.
- [112] Diyi Yang, Jingbo Shang, and Carolyn Penstein Rosé. Constrained Question Recommendation in MOOCs via Submodularity. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM ’14*, pages 1987–1990, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2598-1. doi: 10.1145/2661829.2662089. URL <http://doi.acm.org/10.1145/2661829.2662089>.
- [113] Chun-Nam John Yu and Thorsten Joachims. Learning Structural SVMs with Latent Variables. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML ’09*, pages 1169–1176, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553523. URL <http://doi.acm.org/10.1145/1553374.1553523>.