Stony Brook University

Learning the Intention embedded in the Natural Language Texts: Focused Studies on Connotation and Deception

A Dissertation presented

by

Song Feng

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

Stony Brook University

August 2014

Stony Brook University

The Graduate School

Song Feng

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree, hereby recommend

acceptance of this dissertation

Yejin Choi - Dissertation Advisor
Assistant Professor, Computer Science Department, Stony Brook University

Steven Skiena - Chairperson of Defense
Professor, Computer Science Department, Stony Brook University

I.V. Ramakrishnan–Third Inside Member
Professor, Computer Science Department, Stony Brook University

Rada Mihalcea–Outside Member
Associate Professor
Department of Electrical Engineering and Computer Science, University of Michigan

This dissertation is accepted by the Graduate School
Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Learning the Intention embedded in the Natural Language Texts: Focused Studies on Connotation and Deception

by

Song Feng

Doctor of Philosophy

in

Computer Science

Stony Brook University

2014

In natural-language texts, certain information intended by the author, such as connotation, deception, sarcasm, humor, may not be stated explicitly. Recognizing such authorial intention is one of the keys to truly understanding human communications. There are rapidly increasing interests in uncovering the intention that is embedded in the textual content for real-life applications, such as opinion mining, deception detection, news-gathering, text generation, and educational testing. However, identifying the intended information computationally can be very challenging as it usually requires appropriate syntactic and semantic schemes for interpretations or inferences, and sometimes, the factor of the world knowledge. Previous work addressing authorial intention from different perspectives such as linguistics, rhetoric, psychology and sociology, showing the potentials of computational linguistic techniques for detecting the implicit intention; however, the topic remains largely uncharted.

This thesis describes our focused and in-depth study on how to automatically identify the authorial intention in the textual content. In particular, our study focuses on two types of applications that have not been explored much so far. One is learning the general connotation, which is essentially to identify the nuanced

sentiment that is not necessarily expressed or strictly implied in the text. We aim to exploit the algorithms that are suitable for leveraging large-scale text data with minimalism of world knowledge or human guidance. Therefore, we develop the approaches in light of various linguistic insights and learn the general connotation in a nearly unsupervised manner. We present the first large-scale connotation lexicon over a network of words and senses. The other is detecting the intent of deceit in the writings, which potentially helps suppressing the rampant deceptive behavior in the online community. In this work, we extract salient and discriminating linguistic features from the text and apply supervised learning to predict intended deception in the writing. In addition, this work investigates on the efficacy of assorted informative cues and provides insights based on web resources using computational linguistic techniques. Further more, to generalize our study, we develop automated approaches to collect corpora for deception detection.

Contents

1 Introduction **1**

2 Learning the Lexical Connotation **7**

2.1 Introduction 7
2.2 Task Definition 10
2.2.1 Connotation Lexicon 10
2.2.2 Connotative Predicates 12
2.3 Data 14
2.4 Linguistic Insights 15
2.4.1 Semantic Prosody 16
2.4.2 Semantic Parallelism of Coordination 18
2.4.3 Semantic Relations 20
2.4.4 Practice Use of Google 1T Data 20
2.5 Approaches for Evaluating Connotation Lexicon 21
2.5.1 Evaluation I: Comparison against Sentiment Lexicon . . . 22
2.5.2 Evaluation II: Extrinsic Evaluation via Sentiment Analysis 24
2.5.3 Evaluation III: Intrinsic Evaluation via Human Judgment . 25
2.6 Lexicon Induction Overview 27
2.7 Lexicon Induction via Random Walk 28
2.7.1 Graph Representation 29
2.7.2 Hyperlink-Induced Topic Search (HITS) 30
2.7.3 PageRank 33
2.7.4 Constructing Lexicon 35
2.7.5 Evaluations 35
2.7.6 Conclusion 39
2.8 Lexicon Induction via Label Propagation 39
2.8.1 Graph Representation 40
2.8.2 Propagating over the Overlay of Two Sub-graphs 44

2.8.3 Evaluations . . . 46
2.8.4 Discussion of Graph-based Algorithms . . . 48
2.8.5 Conclusion . . . 49
2.9 Lexicon Induction via Constraint Optimization . . . 49
2.9.1 Graph Representation . . . 50
2.9.2 Induction using Integer Linear Programming . . . 51
2.9.3 Induction using Linear Programming . . . 55
2.9.4 Evaluations . . . 58
2.9.5 Conclusion . . . 62
2.10 Sense-level Connotation . . . 62
2.10.1 Graph Representation . . . 65
2.10.2 Task Overview . . . 66
2.10.3 Induction using Belief Propagation . . . 70
2.10.4 Evaluations . . . 74
2.10.5 Conclusion . . . 84
2.11 Related Work . . . 85
2.11.1 Connotation v.s. Sentiment . . . 85
2.11.2 Sentiment Lexicon . . . 86
2.11.3 Graph Representation . . . 89
2.11.4 Lexicon Induction Techniques . . . 90
2.12 Conclusions and Future Work . . . 91
2.12.1 Summary of Results and Contributions . . . 92
2.12.2 Future Work . . . 93

3 Identifying the Intent to Deceit in the Writings 96
3.1 Introduction . . . 96
3.2 Related Work . . . 98
3.2.1 Deception Detection of Online Resource . . . 99
3.2.2 Writing Styles . . . 101
3.3 Data . . . 102
3.3.1 Overview . . . 102
3.3.2 Four Datasets . . . 104
3.4 Automated Data Acquisition . . . 106
3.4.1 Statistical Analysis . . . 106
3.4.2 Deception Detection Strategies . . . 109
3.4.3 Strategy Evaluation . . . 113
3.4.4 Constructing Datasets . . . 117
3.5 Deceptive Writing Style Analysis . . . 118

3.5.1 Stylometry Analysis 119
3.5.2 Features 120
3.5.3 Experiments 123
3.5.4 Discussion 125
3.5.5 Conclusion 131
3.6 Demographics 132
3.6.1 Challenges 133
3.6.2 Deceptive Review Corpora 135
3.6.3 Experimental Results 137
3.6.4 Elements of Deceptive Wring Styles 140
3.6.5 Conclusion 141
3.7 Conclusions 143
3.7.1 Summary of Results and Contributions 143
3.7.2 Future Work 144

4 Conclusions **150**
4.1 Conclusions and Future Work 150

List of Tables

2.1 Example Words with Learned Connotation: Nouns(n), Verbs(v), Adjectives(a). 11
2.2 Connotative Predicates 13
2.3 Distribution of Answers from AMT. 26
2.4 Examples of newly discovered words with connotations: these words are treated as neutral in some conventional sentiment lexicons. 37
2.5 Example Named Entities (Proper Nouns) with Polar Connotation. 37
2.6 Comparison Result with Sentiment Lexicons (%) 38
2.7 SemEval Classification Result(%) — (〃) denotes that all features in the previous row are copied over. 38
2.8 Twitter Classification Result(%) — (〃) denotes that all features in the previous row are copied over. 39
2.9 Evaluation of the Induction Algorithms with respect to Sentiment Lexicons (prec%). 47
2.10 Accuracy on Sentiment Classification (%) 48
2.11 Evaluation of Induction Algorithms with respect to Sentiment Lexicons (prec%). 57
2.12 ILP/LP Comparison on MQPA′ (%). 58
2.13 Accuracy on Sentiment Classification (%). 60
2.14 Distribution of Connotative Polarity from AMT. 61
2.15 Agreement (Accuracy) against AMT-driven Gold Standard. . . . 61
2.16 Various types of nodes and edges, and counts, in the heterogeneous connotation graph. 67
2.17 Symbols 68
2.18 Instantiation of compatibility potentials. 74
2.19 Connotation inference performance on various graphs. '-w' indicates weighted versions. 78

2.20 SemEval evaluation results, for N=15 81
2.21 Word-/Sense-level evaluation results 83
2.22 Results of pair-wise intensity evaluation, for intensity difference threshold = 2.0 85

3.1 Notational Definitions. 109
3.2 Classification on 5-star reviews: BASELINES 116
3.3 Classification on 5-star reviews: STRATEGY-$avg\Delta$ 116
3.4 Classification on 5-star reviews: STRATEGY-$dist\Phi$. 117
3.5 Classification on 5-star reviews: STRATEGY-$peak\uparrow$. 117
3.6 Deception Detection Accuracy (%). 124
3.7 Cross topic deception detection accuracy: Essay data 127
3.8 Most discriminative production rules in gold standard data 128
3.9 Most discriminative production rules in pseudo gold standard data 128
3.10 Most discriminative sentence outlines of gold standard data. . . . 130
3.11 Most discriminative sentence outlines of pseudo gold standard data. 131
3.12 Most discriminative phrasal tags in PCFG parse trees: TripAdvisor data. 131
3.13 Three Classification Configurations 137
3.14 Classification Accuracy 138
3.15 Similarity of POS distributions between different pairs of classification setups 139
3.16 LIWC & POS are grouped per their agreement/disagreement between the two classification setups. 142
3.17 Comparison of Prominent POS features (sorted by σ) — hotel domain v.s. restaurant domain 147
3.18 Comparison of Prominent POS features (sorted by σ) — real reviews (T_{YELP}) v.s. AMT reviews (T_{AMT}) 148
3.19 Differences (σ) of the distribution of POS tags on Restaurant Reviews. 149

List of Figures

2.1 A Part of AMT Task Design. 26
2.2 Graph Structure for Random Walk Algorithms. 29
2.3 Graph Structure for Graph Propagation. 41
2.4 Graph Structure for Linear Programming 50
2.5 Graph Structure for Word + Sense. 66
2.6 Performance is stable across various ϵ. 79
2.7 Trend of SemEval performance over N, the number of CV folds . 82
2.8 Trend of accuracy for pair-wise intensity evaluation over threshold 84

3.1 Representative distributions of review-ratings for hotels with average rating $\bar{r} \in [3.2, 3.9]$ 107
3.2 Representative distributions of review-ratings for year $y \in [2007, 2011]$ 108
3.3 *Distribution of distribution* of review-ratings by *any-time* reviewers. 111
3.4 *Distribution of distribution* of review-ratings by *single-time* reviewers. 111
3.5 Parse Tree (Example I) 122
3.6 Parse Tree (Example II) 127

Acknowledgements

First and foremost, I wish to thank Yejin Choi, my adviser and role model. You set a great example for me as a true researcher with your ideas, motivation, enthusiasm, and dedication. I admire you being positive, responsible and amiable when facing all sorts of challenges. I deeply appreciate your time and patience for the countless discussions (even late at night) upon my requests (often last-minute), and calling me back to exchange ideas even when you were in the middle of the scuba diving! Your advice on both research and my career will equip me for life.

I have a special thanks to Professor I.V. Ramakrishnan for enlightening me on how to do research for computer science. In addition, I owe my sincere gratitude to my mentor Dr. Janos Hajagos for the continuous support and guidance of my research on Health Informatics. I would like to extend my appreciation to my committee members Professor Steven Skiena and Professor Rada Mihalcea for your time, interests, and insightful comments.

It had been wonderful to work with my fellow labmates Polina Kuznetsova, Jun Seok Kang, Ritwik Banerjee and all my other co-authors. I will always cherish the time that we were motivating, supporting each other, and spending the sleepless nights working together before the deadlines. And also, many thanks to Brian Tria, Paul Fodor for all the help over the years.

My friends at Stony Brook University Lingling, Yiyang, Xiaolei, Yan, Shengnan, Lin, Zhexi, FuBai Group, thank you for giving me a myriad of joy, and

warming my heart with all your kindness (sending me home-made food when I was too busy to go home before deadlines); and Zhiyuan, Tu, Jui-Hao, Naznin, Faisal, Huaqing, thank you for always making things fun for me. My friends Linan, Shuang, Gege, Mao, VivienT, CannyH, Tong&Tong, JackC, thank you for always being there for me no matter what.

Last but definitely not the least, dear Mom and Dad, words cannot express how much I am grateful to you. I love you.

Chapter 1

Introduction

In natural-language texts, certain information intended by the author, such as connotation, deception, sarcasm, humor, may not be stated explicitly. Recognizing such authorial intention is one of the keys to truly understanding human communications. There are rapidly increasing interests in uncovering the intention that is embedded in the textual content for real-life applications, such as opinion mining, deception detection, news-gathering, text generation, and education. However, identifying the intended information computationally can be very challenging as it usually requires appropriate syntactic and semantic schemes for interpretations or inferences, and sometimes even the factor of the world knowledge.

There are some previous work addressing this problem from different aspects, such as Furedy and Ben-Shakhar (1991), Winner and Leekam (1991), Pennebaker et al. (2003), Davidov et al. (2010), Bond and Lee (2005), Chung and Pennebaker (2008), Ott et al. (2011), Reyes et al. (2012), showing the potentials of computational linguistic techniques for recognizing implicit information in the natural-language texts. However, it is ineffective to merely rely on the computational

power (e.g., raw classifier) or to invest too much human effort to develop rules and patterns in real life scenarios. Given the fact that the enormous amount of textual data is generated and made available when human communicating online, the need to detect the intention of the text computationally is integral. Hence, we aim to uncover the intention-level information delivered by the written texts. On that account, we explore the approaches for automatically deriving informative cues based on plain textual content to further enhance the capability of machines with minimalism of world knowledge and guidance from human.

In this work, we particularly focus on two types of applications that have not been explored much so far. One is learning the general connotation based on lexical associations; the other is identifying the intent to deceive in the writings. To effectively learn the intention embedded in the text, we explore different methodologies for both applications. For the former, we leverage large-scale unlabelled data, learn the lexical associations and compute the general connotation in a nearly unsupervised manner. For the latter, we extract salient and discriminative linguistic features from the text and apply supervised learning to detect whether there exists the intended deception in the writing.

Learning the Connotation Connotation is a commonly understood subjective cultural or emotional association that a word or phrase invokes. It is generally described as positive or negative. Note that connotation refers to subtle nuances that intended by the use of language, which can be different from its denotation, the literal meaning. Separating grammatical denotation from connotation of words is important because the connotation of the chosen words usually reflect the intentionality of the author. This work concentrates on learning the connotation lexi-

con which is potentially helpful in discovering the intentionality beyond surface meaning of text. For instance, consider the following:

"Geothermal replaces oil-heating; it helps reducing greenhouse *emissions*."

Although this sentence could be considered as a factual statement from the general standpoint, the subtle intentionality of this sentence may not be entirely objective: this sentence is likely to have an influence on readers' minds in regard to their opinion toward *"geothermal"*. In order to sense the subtle overtone of sentiments, one needs to know that the word *emissions* has generally negative connotation, which geothermal *reduces*. In fact, depending on the pragmatic contexts, it could be precisely the intention of the author to transfer his opinion into the readers' minds. Therefore, understanding the connotation of words plays an important role in interpreting subtle shades of sentiment beyond denotative or surface meaning of text, as seemingly objective statements sometimes allude nuanced sentiment.

There has been a substantial body of research in sentiment analysis over the last decade (Pang and Lee (2008)), where a considerable amount of work has focused on recognizing sentiment that is generally explicit and pronounced than implied and subdued. However in many real-world text, drawing a definite distinction between objective and subjective text can be difficult, perhaps even impractical, because even seemingly objective statements can be opinion-laden in that they often allude nuanced sentiment of the writer (Greene and Resnik (2009)), or purposefully conjure emotion from the readers' minds (Mohammad and Turney (2010)). Although some researchers have explored formal and statistical treatments of those implicit and implied sentiments (e.g. Wiebe et al. (2005); Esuli

and Sebastiani (2006); Greene and Resnik (2009), davidov2010semi), automatic analysis of them largely remains as a big challenge. Many tasks related to sentiment or opinion analysis rely on sentiment lexicons, lexical resources containing information about the emotional implications of words (e.g., sentiment orientation of words, positive or negative).

Understanding the rich and complex layers of connotation remains to be a challenging task. As a starting point, we study a more feasible task of learning the *polarity* of connotation. In this work, we present the first large-scale connotation lexicon. Although there has been a number of previous work that constructed sentiment lexicons (e.g., Esuli and Sebastiani (2006); Wilson et al. (2005); Kaji and Kitsuregawa (2007); Qiu et al. (2009); Chen and Skiena (2014)), which seem to be increasingly and inevitably expanding over words with (strongly) connotative sentiments rather than explicit sentiments alone (e.g., "gun"), little prior work has directly tackled this problem of learning connotation, and much of subtle connotation of many seemingly objective words are yet to be determined. We learn the correlation between words based on web data with the guidance of various linguistic insights and then cast the connotation lexicon induction task as a collective inference problem. To our knowledge, we are the first to explore data-driven approaches to learn the connotation connotation of a large scale of words. We will discuss more details based on our work Feng et al. (2011), Feng et al. (2013) and Kang et al. (2014) in Chapter 2.

Detecting the Deception The online posts today have a significant impact on the formation of public opinions, especially for online review website, which has instantaneous influence on the reputations business entities, guiding the behavior

of billions of consumers world wide. Therefore, many websites are becoming targets of deceptive spams (e.g., Caspi and Gorsky (2006); Jindal and Liu (2008); Ott et al. (2012)). In response to this relatively new challenge in deception detection, there has been burgeoning research that uncovers various cues and anomalous patterns for detecting deceptive writings: ranging from linguistic patterns (e.g., Newman et al. (2003); Mihalcea and Strapparava (2009); Ott et al. (2011); Feng et al. (2012a)) to behavioral patterns of individuals (e.g., Lim et al. (2010); Feng et al. (2012c)) and groups (e.g., Mukherjee et al. (2012); Fei et al. (2013)).

In this work, we concentrate on identifying the intent to deceit in one's writing. In particular, we aim to make further progress to solicit linguistic features that serve as deception cues. In this context, we cast the task as identifying implicitly intended deception as a supervised learning problem and then identify the most discriminative linguistic features discerning deceptive writings from truthful writings. Therefore, we examine the labelled (deceptive, truthful) text and extract a variety of linguistic features in order to set forth a detailed analysis of deceptive text. Most previous studies in computerized deception detection of writings have relied only on shallow lexico-syntactic patterns (e.g., Hancock et al. (2007); Vrij et al. (2007); Mihalcea and Strapparava (2009); Ott et al. (2011)). Given that more advanced language parsing tools are available, we also derive deep syntactic stylometry and evaluate the performance for deception detection, adding a somewhat unconventional angle to prior literature.

For the applications of deception detection, one of the major challenges is evaluation, primarily due to the lack of annotated data with good quality. Literature has shown that humans are not good at catching deception in general (e.g., Bond and DePaulo (2006)), and it proves to be no exception for online reviews

(Ott et al. (2011)). The implication of this observation is that human annotators may not be able to reliably label existing data as deceptive or truthful, except for those less common scenarios where nonlinguistic contextual information such as review time stamps, or IP addresses provide undeniable evidence of dubious acts (e.g., Jindal and Liu (2008); Lim et al. (2010); Mukherjee et al. (2012)). As a result, several previous work has attempted to build deception corpora by instructing participants to lie for a given topic (e.g., Newman et al. (2003); Potthast et al. (2010)), the work by Mihalcea and Strapparava (2009) is among the first to use Amazon Mechanical Turk to collect truthful and deceptive writings. Ott et al. (2011) also resorts to crowdsourcing to create deceptive gold standard. However, such manufactured data is still quite expensive to obtain. In order to conduct a scalable study on deception detection, we also investigate an alternative approach to acquire deceptive text in an unsupervised manner. We will present more details based on our work (Feng et al. (2012b), Feng et al. (2012a) and Feng et al. (2012c)) in Chapter 3.

Chapter 2

Learning the Lexical Connotation

2.1 Introduction

A connotation generally refers to the suggestive meaning, including the implicature of emotions or associations a word has beyond its literal meaning. When people communicate in form of writings *between writers and readers* or conversations *between participants*, one may use connotative information, in other words, intentional inclusion to help create mood and tone, as well as aim to control how other people (a reader or a conversation partner) will think of a person, place, thing, or concept. Consider the following product description,

> "Part sculpture, part table, all artisanal. Craftspeople in Jaipur, India, hand carved the delicate rosettes on this low-lying solid mango wood table, which takes its original inspiration from a ceremonial stool used by Bamileke royalty in the African country of Cameroon."

The majority of highlighted words in the text[1] above are not listed in the existing sentiment lexicons, and not necessarily sentiment-laden in the explicit way. However, often times, it is these seemingly objective words that can be impactful in evoking an positive or negative sentiment. Similarly, the verb "discipline", which is also not in the existing sentiment lexicons, means to train (someone) to obey rules or a code of behavior, and it carries a negative connotation, because, in practice, it is usually regulated through punishment. In some cases, two words can have the same literal meaning, but distinctive connotations. Consider "home" and "house", both refer to places where people live; however, the word "home" might remind readers of a place of warmth and family, while the word "house" is more unfeeling and impersonal. In addition, the literal meaning of a word or short phrase hardly changes, but the connotation varies in different domains. For instance, the word "robot", when it is used in product, "robot" indicates it is assisting and free human from labor in certain sense; however, when it is related to Internet, often times, it is used with a negative or malicious connotation [2]. Therefore, such connotative knowledge at lexical level would be helpful in understanding the underlying *intent* of text in the situations when the authors have the intention of conjuring positive/negative implicature. As seen in the example, the author aims to invoke an exotic and artistic feel, in particular from the customers' minds, even though it does not rely on a load of more explicitly sentiment-laden words that are more common in domains such as product reviews.

In addition, while all the words have literal meanings, there are a considerably large number of words also have a connotation in addition to the surface meanings (Feng et al. (2011)). Therefore, understanding the connotation of words is integral

[1] From WestElm.com, an online furniture store.

[2] See the term "Botnet" (`http://en.wikipedia.org/wiki/Botnet`)

to gaining deeper meaning from the text in practice. Learning the rich and complex layers of connotation remains to be a challenging task. As a starting point, we study a more feasible task of learning the *polarity* of connotation. For one thing, connotation is traditionally considered to be of an associative and subjective nature and deemed peripheral to the understanding of the linguistic sign (positive or negative). For another, many tasks related to sentiment analysis rely on lexicons - lexical resources containing information about the polarized implications of words. Hence, we aim to build a large scale of connotation lexicon.

To learn the connotative polarities of words and phrases, the methodologies largely vary depending on the available knowledge / resources, on the nature of information processor. Here we aim to mainly rely on the statistics of word relations that we can mine from the web data with minimum amount of prior knowledge. Learning the connotation of words seems to require much common sense and world knowledge, which in turn might require the human encoding of knowledge base that is obviously a luxury for most of the real life applications. In addition, the connotation of a word varies in different context, which raises the need to develop domain-specific lexicon. So we aim to find an approach to computationally learn the connotation with minimum prior knowledge. In this work, we have found that much of the connotative polarity of words can be statistically inferred from natural language text based on *semantic prosody* (in Section 2.4.1) that is basically under a central premise of collocational frequency of words that affect and shape the polarity of connotation. Therefore, with a small set of seed words as prior knowledge and collocational statistics, we can cast the connotation lexicon induction task as a collective inference problem.

In the following sections, we first define the task in Section 2.2. And we discuss the linguistic insights in Section 2.4 that can be incorporated with the inference algorithms. Then we present the graph representations based on different sets of linguistic insights tailored for different algorithms. Last, we discuss the details of the inference algorithms we have explored in Section 2.6 followed by experiments for evaluations. Our experiments show that including the connotative meaning of words generally enhances the sentiment analysis tasks.

2.2 Task Definition

In this section, we will formally define the task of learning the general connotation. We will first elaborate the concept of "connotation" and define connotation lexicons, and then introduce "connotative predicates", which plays an import role in our task.

2.2.1 Connotation Lexicon

Connotation or connotative meaning generally refers to the additional or secondary meaning of a word or phrase, associated with a polarized (positive or negative) sign in addition to its denotative content. For the study of learning the connotative intentional inclusion in the text, we propose a new type of lexicon, connotation lexicon. A *connotation lexicon* is a lexicon that lists words with connotative polarities, i.e., words with positive connotations (e.g., "award", "promotion" and "tenure") and words with negative connotations (e.g., "cancer", "war" and "trap"), more examples with different POS labeled by our approaches are presented in Table 2.1.

	POSITIVE	NEGATIVE	NEUTRAL
n.	avatar, stakeholder, adrenaline, keynote, debut, omaha, cooperation	unbeliever, shortfall, onlineshop, katrina, overpayment, microscope	header, heat, outline, clothing, mark, grid, table, course, preview
v.	handcraft, accredit, volunteer, party, personalize, nurse, google, adjust	sentence, cough, trap, stalk, scratch, debunk, rip, misspell, overcharge	state, edit, send, put, arrive, type, drill, name, stay, echo, register
a.	floral, vegetarian, prepared, ageless, funded, contemporary, detailed	debilitating, impaired, communist, swollen, intentional, jarring, unearned	same, middle, west, uncut, automatic, hydration, routine, sided

Table 2.1: Example Words with Learned Connotation: Nouns(n), Verbs(v), Adjectives(a).

- **Words with positive connotation:** We define words with positive connotation as those that are used to describe physical objects or abstract concepts that people generally value, cherish or care about. For instance, we regard words such as "freedom", "life", or "tenure" as the ones with positive connotation.

- **Words with negative connotation:** We define words with negative connotation as those that are used to describe physical objects or abstract concepts that people generally devalue, dislike or avoid. Some of these words may express subjectivity (e.g., "disappointment", "humiliation"), while many other are purely objective (e.g., "bedbug", "arthritis, "funeral").

Please note that the connotation lexicon differs from conventional sentiment lexicons that are studied in much of previous research (e.g., Stone and Hunt (1963); Wiebe et al. (2005); Esuli and Sebastiani (2006); Wilson et al. (2005);

Qiu et al. (2009)): the latter concerns words that *express* sentiment either explicitly or implicitly, while the former concerns words that *evoke* or even simply *associate with* a specific polarity of sentiment. In fact, most positive or negative connotative words in Table 2.1 are considered as neutral by conventional lexicons such as MPQA Subjectivity Lexicon (Wiebe et al. (2005)) and General Inquirer (Stone and Hunt (1963)). A substantial number of words with positive or negative connotation merely carries a nuanced sentiment. Due to practical reasons, we do not differentiate positive / negative sentiment from positive / negative connotation in this work.

2.2.2 Connotative Predicates

In our study, a *connotative predicate* is defined as a predicate that has selectional preference on the connotative polarity of some of its semantic arguments. For instance, in the case of the connotative predicate *"prevent"*, there is a strong selectional preference on negative connotation with respect to the semantic role. That is, statistically speaking, people tend to associate negative connotation with *"prevent"*, e.g., *"prevent cancer"* or *"prevent war"*, rather than positive connotation, e.g., *"prevent promotion"*. Similarly, *"congratulate"* or *"praise"* has a strong selectional preference on positive connotation with respect to the semantic role. More formally defined as below.

- **Positively connotative predicate:** We define positively connotative predicates as those that expect positive connotation in some its arguments. This definition can be readily extended to govern other thematic roles. For example, "congratulate" or "save" are positively connotative predicates that expect words with positive connotation in the arguments: people typically

POSITIVE PREDICATES	NEGATIVE PREDICATES
accomplish, achieve, advance, advocate, admire, applaud, appreciate, compliment, congratulate, develop, desire, enhance, enjoy, improve, praise, promote, respect, save, support, win	alleviate, accuse, avert, avoid, cause, complain, condemn, criticize, detect, eliminate, eradicate, mitigate, overcome, prevent, prohibit, protest, refrain, suffer, tolerate, withstand

Table 2.2: Connotative Predicates

congratulate something positive, and save something people care about. Please see Table 2.2 for more examples of positively connotative predicates.

- **Negatively connotative predicate:** We define negatively connotative predicates as those that have a selectional preference on the negative connotation in some of its arguments. For instance, predicates such as "prevent" or "suffer" tend to project negative connotation in the argument. Please see Table 2.2 for more examples of negatively connotative predicates.

One interesting linguistic phenomenon is that positively connotative predicates are not necessarily positive sentiment words. For instance "save" is not a positive sentiment word in the lexicon published by Wiebe et al. (2005) but it has a selectional preference on the positive connotation of its argument. On the other hand, (strongly) positive sentiment words are not necessarily (strongly) positively connotative predicates, e.g., "illuminate", "agree". Likewise, negatively connotative predicates are not necessarily negative sentiment words. For instance, predicates such as "prevent", "detect", or "cure" are not negative sentiment words, but they tend to correlate with negative connotation in their argument. Inversely, (strongly) negative sentiment words are not necessarily (strongly) negatively connotative predicates, e.g., "abandon" ("abandoned [something valuable]").

In this work, we use a small set of the connotative predicates, presented in Table 2.2, as the prior knowledge in our inference models. The key to our approach is *selectional preference* of *connotative predicates* or *semantic prosody* within the predicate-argument structure. The quality of the connotative predicates is critical for soliciting good candidates for the connotation lexicon in our approach. We consider a positively (negatively) connotative predicate as of high quality if the probability of its argument with positive (negative) sentiment/connotation is significantly higher than the negative (positive) or neutral. For instance, "prevent" appears a better candidate than "stop" for a negatively connotative predicates while "achieve" is likely to be a better word than "obtain" for a positively connotative predicates. Therefore, we cannot just simply include the synonyms for the seed connotative predicates. For brevity, we explore only verbs as the predicates, and words that appear in the thematic role of the predicates as arguments. The arguments are considered as the candidates of the connotative words. Even though our work is based on verb predicates, it however can be readily extended to exploit other type kind of predicates based on predicate-argument structures.

2.3 Data

Before we discuss how we explore linguistic heuristics, we first describe the datasets that we resort to for learning the correlation between words. In this work, we aim to attain a broad coverage lexicon while maintaining good quality. Therefore, we refer to the statistical linguistic patterns from the web-driven data to mine the relatedness between words. We also exploit existing lexical resources such as semantic relations of synonyms and antonyms as an additional inductive bias.

Web-driven Data: One data source in this work is from the Web. Considering that the success of the learning depends on the proper quantization of lexical associations in the form of co-occurrence statistics, which would be directly proportionate to the amount of data available, we therefore need a substantially large amount of documents. Given the non-trivial challenge of collecting and handling web-scale data, the data we resort to is the Web 1T corpus (Brants and Franz (2006)) based on Web articles. It provides English n-grams (from unigrams to 5-grams) and the observed frequency counts calculated over 1 trillion words from web page text, the n-gram with the frequency lower than 40 times is already eliminated. The use of Web 1T data helps lessening the challenge with respect to data acquisition, while still allows us to enjoy the co-occurrence statistics of web-scale data. In addition, it is relatively convenient to calculate the approximation to the particular co-occurrence statistics of the online language usage we are interested in. More details on how we exploit the data will be explained in Section 2.4.

Dictionary-driven Data The other corpus we exploit is readily available lexical resource WordNet (Miller (1995)), the design of which is inspired by psycholinguistic. We gain access to the semantic relation such as synonyms, antonyms in the WordNet synsets, which has been used to explore the sentiment orientation of words (e.g., Kamps et al. (2004); Hu and Liu (2004)).

2.4 Linguistic Insights

For many NLP applications, such as question-answering, multi-document summarization, information retrieval, one central challenge is to estimate the relatedness between a random pair of words. In this work, we aim to determine the associ-

ation between word pairs so as to infer the labels based on the seeds predicates. Therefore, we are particularly interested in the words pairs that are implicitly and explicitly connected under the polarized connotative relation. For this, we exploit two types of lexical relations based on different lexico–syntactic and semantic patterns, that is, predicate–argument pairs of words based seed connotative predicates, and pairs of words based on semantic parallelism of coordination based on statistical information calculated based on the Web data.

2.4.1 Semantic Prosody

In corpus linguistics, semantic prosody describes how some of the seemingly neutral words (e.g., "cause") can be perceived with positive or negative polarity because they tend to collocate with words with corresponding polarity (e.g., Sinclair (1991); Louw (1993); Stubbs (1995); Stefanowitsch and Gries (2003)). Therefore, we propose statistical approaches that exploit this very concept of semantic prosody to infer the connotative polarities of words. Specifically, we rely on the statistics of semantic prosody with predicate-argument structure based on seed connotative predicates. For brevity, we explore only connotative predicates of verbs as prior knowledge. Our idea can be readily extended to exploit other predicate-argument relations such as nouns and phrases. Through this work, we may refer the words to evaluate as "argument words" at times as they are practically considered to be the possible words that would appear at the argument position of the seed connotative predicates. As we only consider the verb predicates and the predicate–argument pairs that the argument of the verb predicate appears on the right hand side of the verb, we also assume that the argument is within the close range of the predicate.

We now describe how to derive co-occurrence statistics of each predicate–argument pair using the Google Web 1T data. For a given predicate p and an argument a, we add up the count (frequency) of all n-grams ($2 \leq n \leq 5$) that match the following pattern:

$$[p]\ [\star]^{n-2}\ [a]$$

where p must be the first word (head), a must be the last word (tail), and $[\star]^{n-2}$ matches any $n-2$ number of words between p and a. Note that this rule enforces the argument a to be on the right hand side of the predicate p. Furthermore, this rule ensures that we do not double-count the frequencies of the same word sequence appearing across different length of n-gram. For instance, above matching rule does not allow the frequency of a 3-gram $[p]\ [a]\ [\star]$ to be included, which is good since such count is already included in the 2-gram count of $[p]\ [a]$. To reduce the level of noise, we do not allow the wildcard $[\star]$ to match any punctuation mark, as such n-grams are likely to cross sentence boundaries representing invalid predicate – argument relations. We consider a word as a predicate if it is tagged as a verb by a Part-of-Speech tagger (Toutanova and Manning (2000)). For argument $[a]$, we only consider content-words. There are 8427838 n-gram records that meet the criteria listed above, with 508171 unique tail words (candidates for arguments). In this work, we only consider the unigram arguments in n-grams. Learning the multi-word expressions are considered for our future work.

The use of web n-gram statistics necessarily invites certain kinds of noise for predicate-argument pairs. For instance, some of the $[p]\ [\star]^{n-2}\ [a]$ patterns might not correspond to a valid predicate–argument relation. However, we expect that, statistical, our graph-based algorithms will be able to discern the valid relations from the noise by focusing on the important part of the graph. In other words,

we expect that predicates with a strong selectional preference will be supported by connotative arguments, and vice versa; thereby resulting in a reliable set of predicates and arguments that are mutually supported by each other.

To quantify the mutually reinforcing relation between predicates and arguments, one option is pairwise mutual information, which has been used by many previous research to quantify the association between two words (e.g., Church and Hanks (1990); Turney; Newman et al. (2009)), which forms undirected edges of a graph.

$$PMI(p, a) = log \frac{P(p, a)}{P(p)P(a)}$$

If we consider the direction of the predict-argument edges, we define the edge weight with conditional probability as follows,

$$w(p \rightarrow a) := P(a|p) = \frac{P(p, a)}{P(p)}$$

$$w(a \rightarrow p) := P(p|a) = \frac{P(p, a)}{P(a)}$$

$P(p, a)$ - the count of occurrence of p and a. $P(p)$ ($P(a)$) - the frequency of p (a).

2.4.2 Semantic Parallelism of Coordination

In addition to connotative predicates and the corresponding argument words, we also explore the relation between argument and argument words. To suppress the noise, we only consider the word pairs within *semantic parallelism of coordination* (e.g., Bock (1986); Hatzivassiloglou and McKeown (1997); Pickering and Branigan (1998)). In particular, we extract the syntactic pattern of "a_i and a_j"

from Google n-grams, sentiment consistency. We consider the "a_j and a_i" and "a_j and a_i" the same and combine them. To quantify the semantic similarity between two argument words, we refer to two kinds of statistical measurements: one is pairwise mutual information and the other is distributional similarity.

Here we explore the distributional properties, which we obtain from the web-driven data in order to infer robust numerical lexical relations between words. In particular, we build a co-occurrence word vector for each word a_i based on pattern of a_i and a_j only if they occurred together in the "$a_i\ and\ a_j$" or "$a_j\ and\ a_i$" coordination in the Google Web 1T data. The co-occurrence vector for each word is computed using PMI scores with respect to the top n co-occurring words. Here we discard edges with cosine similarity ≤ 0, as those indicate either independence or the opposite of similarity. n (=50) is selected empirically. The PMI scores are calculated as follows,

$$PMI(a_i, a_j) = log \frac{P(a_i, a_j)}{P(a_i)P(a_j)}$$

Thus, we obtain continuous word vector representation for argument word and then we calculate the cosine similarity between two vectors for the two argument words a_1 and a_2, as follows:

$$w(a_1 - a_2) = CosineSim(\overrightarrow{a_1}, \overrightarrow{a_2}) = \frac{\overrightarrow{a_1} \cdot \overrightarrow{a_2}}{||\overrightarrow{a_1}||\ ||\overrightarrow{a_2}||}$$

where $\overrightarrow{a_1}$ and $\overrightarrow{a_2}$ are co-occurrence vectors for a_1 and a_2 respectively.

2.4.3 Semantic Relations

In addition to statistical correlation based on the Web data, we also refer to the semantic relations. Lexical semantic relations such as synonyms and antonyms are additional inductive bias that has been shown generally helpful for the inference of the polarity of the words (Lu et al. (2011)). For this, we take advantage of existing dictionary-driven information – WordNet (Miller (1995)) synsets to draw the semantic relations between words. Thus, by including dictionary-driven semantic relations, we aim at enhancing the precision of labelling results; we also leverage dictionary-driven words, i.e., unseen words with respect to those relations explored, towards increasing the size of the labelled data, which potentially expands the coverage of the connotation lexicon.

2.4.4 Practice Use of Google 1T Data

Since the use of Google 1T Data is closely related to linguistic insights that we explore for the work, we will discuss the practice use of the data in this section after we introduce those linguistic insights. Remind that, to quantify various kind of the association between different type of words, we compute the corresponding statistical information based Google Web 1T data. It provides us great convenience in term of processing data and extract syntactic patterns. We also notice some limitations such as noise due to the nature of web, and the limited range of n-gram ($1 \leq n \leq 5$), which prevents us from exploiting a more complete co-occurrence statistics based on the semantic relations, i.e., predicate-argument patterns in a longer text span.

Upon a closer look at the noise in Google n-grams, there are many sources

of noise that may be introduced in the construction of graph. For instance, some of the predicates might be negated, changing the semantic dynamics between the predicate and the argument. In addition, there might be many unusual combinations of predicates and arguments, either due to data processing errors or due to idiosyncratic use of language. Some of such combinations can be valid ones (e.g., "prevent promotion"), challenging the learning algorithm with confusing evidence. In this work, we largely reply on the statistical frequency of co-occurrence readily available from the data. We hypothesize that this mutually reinforcing relation between connotative predicates and their arguments can be captured via graph centrality in theory. The graph representation (to be described) captures general semantic relations between predicates and arguments, rather than those specific to connotative predicates and arguments. Therefore in the following sections, we will explore techniques to augment the graph representation so as to bias the centrality of the network of words toward connotative predicates and arguments.

To obtain a bit better accuracy, we do some pre-processing on the n-grams by using Stanford Log-linear Part-Of-Speech Tagger (Toutanova and Manning (2000)) to POS-tag n-grams and then filter out the n-grams with punctuations and other special characters to reduce the noise.

2.5 Approaches for Evaluating Connotation Lexicon

Before discussing the details of the algorithms, we will first describe how we evaluate the connotation lexicon in this section. The evaluation results will be presented in the section where each algorithm is described. In this work, we propose intrinsic (corpus-based) evaluation along with extrinsic evaluation (application-

based). Firstly, we adopt the typical mean of comparing the generated lexicons with a comparable "gold-standard" – sentiment lexicons. In addition to this, we propose to assess the lexicon creation methods based on the performance in actual sentiment analysis application, which can be more insightful in terms of practical use, as the lexicons are usually developed for such applications. Lastly, we ask human annotators to label the words and then create a gold standard for connotative polarity.

2.5.1 Evaluation I: Comparison against Sentiment Lexicon

One straightforward way to evaluate the connotative polarity of words is to compare against a gold-standard lexicon. In our case, the polarity defined in the connotation lexicon differs from that of conventional sentiment lexicons, in which we aim to recognize more subtle sentiment that correlates with words. Intuitively, any word with polar sentiment (positive or negative) should have the connotation of the matching polarity, while the inverse does not necessarily hold.

Nevertheless, we provide agreement statistics between our connotation lexicon and conventional sentiment lexicons for comparison purposes. We collect statistics with respect to the following two resources: General Inquirer (Stone and Hunt (1963)) and MPQA (Wiebe et al. (2005)). We consider SentiWordNet (Baccianella et al. (2010)) when evaluate sense-level connotation. Generally, the most common polar words are used consistently in terms of their polarity; or rather, we expect a moderate agreement between our lexicon and existing sentiment lexicon. The degree of agreement can implicitly reflect the quality of our lexicon.

- General Inquirer (Stone and Hunt (1963)) is a human annotated dictionary consisting of $1,915$ words with positive sentiment and $2,291$ words with

negative sentiment. It has been used by many previous studies as a benchmark reference lexicon (e.g., Esuli and Sebastiani (2006); Mohammad et al. (2009)).

- MPQA (Wiebe et al. (2005)) is a combination of human annotations and other resources, resulting in $4,879$ positive words, 2821 negative words, 498 with both positive and negative polarities.

- SentiWordNet (Baccianella et al. (2010)) provides assigns a positive and negative score to synsets. A synset with a positive score higher than its negative score is considered negative and vice versa. Thus, it contains $25,674$ positive words and $15,185$ negative words.

The way we calculate the agreement between connotation lexicon and semantic lexicon is as follows. For polarity $\lambda \in \{+, -\}$, let $count_{\text{sentlex}(\lambda)}$ denote the total number of words labeled as λ in a given sentiment lexicon, and let $count_{\text{agreement}(\lambda)}$ denote the total number of words labeled as λ by both the given sentiment lexicon and our connotation lexicon. In addition, let $count_{\text{overlap}(\lambda)}$ denote the total number of words that are labeled as λ by our connotation lexicon that are also included in the reference lexicon with or without the same polarity. Then we compute $prec_\lambda$ as follows:

$$prec_\lambda \, \% = \frac{count_{\text{agreement}(\lambda)}}{count_{\text{overlap}(\lambda)}} \times 100$$

2.5.2 Evaluation II: Extrinsic Evaluation via Sentiment Analysis

Next we perform extrinsic evaluation to quantify the practical value of our connotation lexicon in concrete sentiment analysis applications. In particular, we make use of our connotation lexicon for binary sentiment classification tasks in two different ways:

- Unsupervised classification by voting. We define r as the ratio of positive polarity words to negative polarity words in the lexicon. In our experiment, penalty is 0 for positive and -0.5 for negative.

$$score(x_+) = 1 + penalty_+(r, \#positive)$$

$$score(x_-) = -1 + penalty_-(r, \#negative)$$

- Supervised classification using SVM. We use bag-of-words features for baseline. In order to quantify the effect of different lexicons, we add additional features based on the following scores as defined below:

$$score_{raw}(x) = \sum_{w \epsilon x} s(w)$$

$$score_{purity}(x) = \frac{score_{raw}(x)}{\sum_{w \epsilon x} abs(s(w))}$$

Data The two corpora we use are SemEval2007 (Strapparava and Mihalcea (2007a)) and Sentiment Twitter.[3]

[3] http://www.stanford.edu/~alecmgo/cs224n/twitterdata.2009.05.25.c.zip

SemEval is obtained from the SemEval task (Strapparava and Mihalcea (2007a)). It is a set of news headlines with annotated scores (ranging from -100 to 87). The positive/negative scores indicate the degree of positive/negative polarity orientation. We construct several sets of the positive and negative texts by setting thresholds on the scores as shown in Table 2.13. "$\leqslant n$" indicates that the positive set consists of the texts with scores $\geq n$ and the negative set consists of the texts with scores $\leq -n$.

Tweets consists of tweets containing either a *smiley* emoticon (representing positive sentiment) or a *frowny* emoticon (representing negative sentiment), we randomly select 50000 *smiley* tweets and 50000 *frowny* tweets. [4] The classification results are based on a 5-fold cross validation.

2.5.3 Evaluation III: Intrinsic Evaluation via Human Judgment

In order to measure the quality of the connotation lexicon, we also perform human judgment study on a subset of the lexicon. Because we expect that judging a connotation can be dependent on one's cultural background, personality and value systems, we gather judgements from 5 people for each word, from which we hope to draw a more general judgement of connotative polarity. We gather gold standard only for those words for which more than half of the judges agreed on the same polarity. Otherwise we treat them as ambiguous cases. We also allow Turkers to mark words that can be used with both positive and negative connotation, which results in about 7% of words that are excluded from the gold standard set.

[4] We filter out stop-words and words appearing less than 3 times. For Twitter, we also remove user names of the format *@username* occurring within tweet bodies.

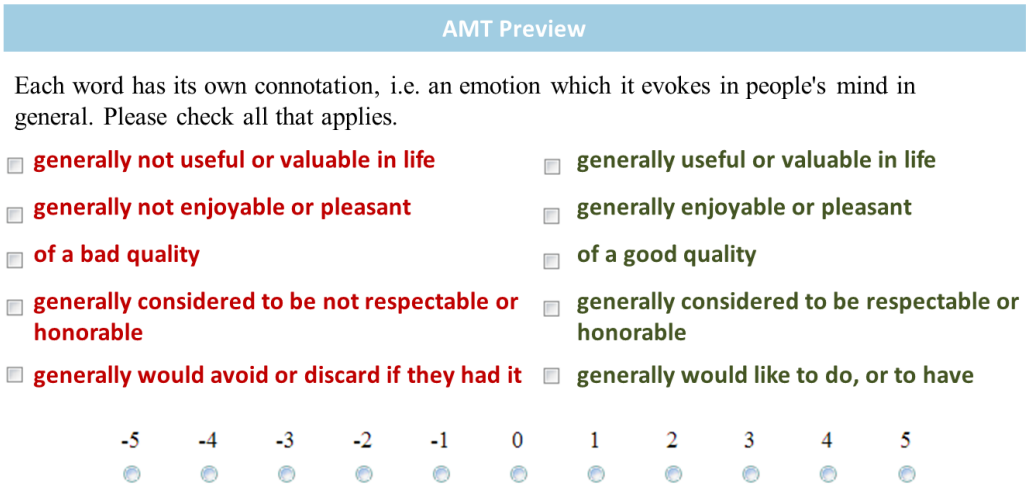


Figure 2.1: A Part of AMT Task Design.

	YES		NO	
QUESTION	%	Avg	%	Avg
"Enjoyable or pleasant"	43.3	2.9	16.3	-2.4
"Of a good quality"	56.7	2.5	6.1	**-2.7**
"Respectable / honourable"	21.0	**3.3**	14.0	-1.1
"Would like to do or have"	52.5	2.8	11.5	-2.4

Table 2.3: Distribution of Answers from AMT.

Figure 2.1 shows a part of the AMT task, where Turkers are presented with questions that help judges to determine the subtle connotative polarity of each word, then asked to rate the degree of connotation on a scale from -5 (most negative) and 5 (most positive). To draw the gold standard, we consider two different voting schemes:

The resulting distribution of judgements is shown in Table 2.3 & 2.14. Interestingly, we observe that *among the relatively frequently used English words, there are overwhelmingly more positively connotative words than negative ones.*

As a control set, we also include 100 words taken from the General Inquirer lexicon: 50 words with positive sentiment, and 50 words with negative sentiment. These words are included so as to measure the quality of human judgment against a well-established sentiment lexicon. When computing the annotator agreement score or evaluating our connotation lexicon against human judgment, we consolidate -1 and -2 into a single negative class and 4 and 5 into a single positive class. The Kappa score between two human annotators is 0.78.

2.6 Lexicon Induction Overview

Our task is essentially to learn connotative words automatically with broad coverage. One of the challenges of such task is that there is the increasing need for domain adaption for sentiment analysis (Blitzer et al. (2007)) and also for many practical data mining applications, unlabeled training examples are readily available, but labeled ones are fairly expensive to obtain. With such concerns in mind, we exploit the approaches that require minimum of supervision or only small amount of annotated data, starting from a small set of seed words and the web data. We consider approaches of several distinct types of algorithmic frameworks that are suitable for incorporating different sets of linguistic insights: (1) random walk based on HITS/PageRank , (2) label propagation (e.g., Zhu and Ghahramani (2002); Velikovich et al. (2010)), (3) constraint optimization based on Integer Linear Programming (e.g., Roth and Yih (2004); Choi and Cardie (2009); Lu et al. (2011)) (4) we also further explore one novel graph-theoretic unified approach for learning the connotation at both word-level and sense-level. A number of different graph representations are exploited for the algorithms respectively, which will

be elaborated in the following section. Our empirical study demonstrates that our approaches are effective in learning both connotation lexicon.

2.7 Lexicon Induction via Random Walk

Remind that the key linguistic insights behind our approach is semantic prosody over predicate-argument syntactic pattern. Therefore, as the very first attempt of learning the general connotation, we first investigate how effective the semantic prosody is on indicating the connotative polarity. The only prior knowledge we consider is a handful of seeds, which are connotative predicates. Motivated by the previous success on opinion mining / sentiment analysis via random walk (e.g., Esuli and Sebastiani (2007); Heerschop et al. (2011); Montejo-Ráez et al. (2012)) and graph-based ranking (e.g. Mihalcea and Tarau (2004); Erkan and Radev (2004); Mihalcea and Csomai (2007)), we employ random walk model, specifically, HITS (Kleinberg (1999)) and PageRank (Page et al. (1999)) algorithms. The idea is derived from the observation that the relation between predicates and their corresponding arguments can be seen as a bipartite graph (to be presented in Section 2.7.1). The model of word relations is structurally akin to the hyperlinked documents of the web. Thus it naturally leads to the analysis of network centrality via link analysis algorithms, with which we rank the words in terms of their possess of positiveness and negativeness. We categorize the words as positive or negative based on the ranking results. We will explore the techniques to incorporate prior knowledge into random walk, as will be elaborated in 2.7.2 and Section 2.7.3 and then evaluate the results in various aspects.

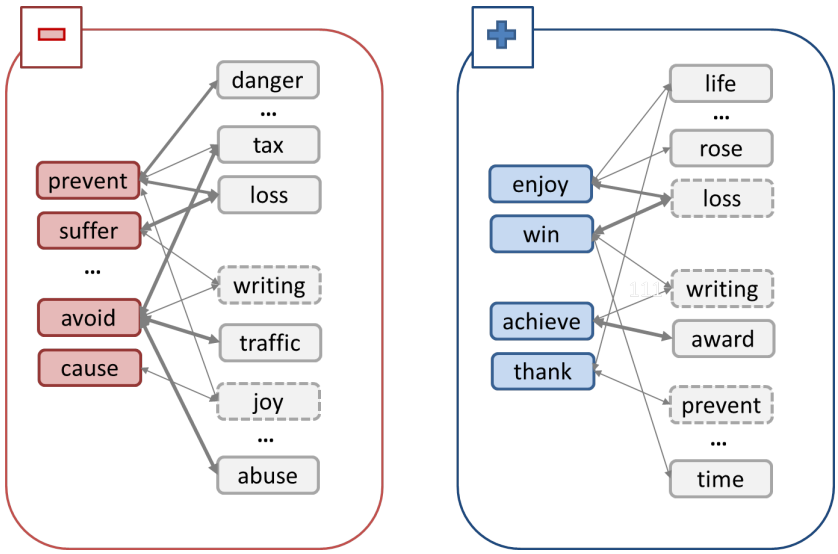


Figure 2.2: Graph Structure for Random Walk Algorithms.

2.7.1 Graph Representation

We model the task based on the semantic relations between connotative predicates (labeled) and their corresponding arguments (to be labeled). To establish a learning bias for the word graph, we start with a small set of connotative predicates as seed words, including 20 positive and 20 negative ones. These seed words act as the sole prior knowledge in our learning process. Note that for each technique in this section, we construct two separate graphs G^+ and G^- corresponding to positive and negative polarity respectively. That is, G^+ learns positively connotative predicates and arguments, while G^- learns negatively connotative predicates and arguments.

Figure 2.2 depicts the key intuition of the bipartite graph - the mutually reinforcing relation between connotative predicts and the words with connotative polarity. The nodes on the left-hand side correspond to connotative predicates,

and the nodes on the right-hand side correspond to words in the argument. There is an edge between a predicate p_i and an argument a_i, if the argument a_i appears in the thematic role of the predicate p_i. The thickness of edges represents the strength of the association between predicates and arguments.
More formally, let $G = (V, E)$ be the bipartite graph, where
V: set of nodes, consists of two types of nodes P and A. P corresponds to the connotative predicates with positive/negative labels and A corresponds to the argument words as the candidates of connotation lexicon to be labeled.
E: set of edges $\{e_i\}$. $e_i = (p_i, a_i)$ (or $< p_i, a_i >$), with $p_i \in P, a_i \in A$.

Next we explore both undirected and directed edges. For undirected graphs, the value of $w(i, j)$ is set to $w(i - j)$ defined as

$$w(p - a) := PMI(p, a) = log\frac{P(p, a)}{P(p)P(a)}$$

For directed graphs, the value of $w(i, j)$ is set to $w(i \rightarrow j)$, which is defined as

$$w(p \rightarrow a) := P(a|p) = \frac{P(p, a)}{P(p)}$$

$$w(a \rightarrow p) := P(p|a) = \frac{P(p, a)}{P(a)}$$

$P(p, a)$ - the count of occurrence of p and a. $P(p)$ ($P(a)$) - the frequency of p (a).

2.7.2 Hyperlink-Induced Topic Search (HITS)

HITS (Hyperlink-Induced Topic Search) algorithm (Kleinberg (1999)), also known as Hubs and authorities, is a link analysis algorithm that is particularly suitable to

model mutual reinforcement between two different types of nodes: hubs and authorities. Briefly, a hub is a page with many out-links, and an authority is a page with many in-links. The definitions of hubs and authorities are given recursively. A (good) hub is a node that points to many (good) authorities, and a (good) authority is a node pointed by many (good) hubs. Notice that the mutually reinforcing relationship is precisely what we intend to model between connotative predicates and arguments. In this work, we conceptualize the seed connotative predicates as hubs and the argument words (the candidates of connotation lexicon) as authorities.

Let $a(A_i)$ and $h(A_i)$ be the authority and hub score respectively, for a given node $A_i \in A$. Then we compute the authority and hub score recursively as follows:

$$a(A_i) = \sum_{P_i, A_j \in E} w(i,j) h(A_j) + \sum_{P_j, A_i \in E} h(P_j) w(j,i)$$

$$h(A_i) = \sum_{P_i, A_j \in E} w(i,j) a(A_j) + \sum_{P_j, A_i \in E} a(P_j) w(j,i)$$

We use a to denote the column vector with all the authority scores, h to denote the column vector with all the authority scores, n be the sum of size of P and size of A.

$$a = (a(1), a(2), ..., a(n))^T$$

$$h = (h(1), h(2), ..., h(n))^T$$

Then,

$$a = L^T h$$

$$h = La$$

The co-occurrence matrix derived is denoted as L, where

$$L_{ij} = \begin{cases} w(i,j) & if(P_i, A_j) \in E \\ 0 & otherwise \end{cases}$$

The final hub-authority scores of nodes are determined after infinite repetitions of the algorithm. Let a_k and h_k denote authority and hub vectors at the kth iteration, the iterations for generating the final solutions are,

$$a_k = L^T L a_{k-1}$$

$$h_k = L L^T h_{k-1}$$

We expect that words that appear often in the THEME role of various positively (or negatively) connotative predicates are likely to be words with positive (or negative) connotation. Likewise, predicates whose THEME contains words with mostly positive (or negative) connotation are likely to be positively (or negatively) connotative predicates. In short, we can induce the connotative polarity of words using connotative predicates, and inversely, we can learn new connotative predicates based on words with connotative polarity.

Prior Knowledge via Truncated Graph When constructing the bipartite graph, we limit the set of predicates P to only those words in the seed set, instead of including all words that can be predicates. In a way, the truncated graph representation can be viewed as the query induced graph on which the original HITS

algorithm was invented (Kleinberg (1999)).

We hypothesize that by focusing on the important part of the graph via centrality analysis , it is possible to infer connotative polarity of words despite various noise introduced in the graph structure. This implies that it is important to construct the graph structure so as to capture important linguistic relations between predicates and arguments. With this goal in mind, we next explore the directionality of the edges and different strategies to assign weights to them.

2.7.3 PageRank

In this section, we explore the use of another popular approach for link analysis: PageRank (Page et al. (1999)). PageRank utilizes a random walk model to iteratively update the score at a node, n_i based on the scores of nodes that arc to n_i. As shown earlier, $G = (V, E)$ is the graph, where $v_i \in V = P \cup A$ are nodes (words) for the disjunctive set of predicates (P) and arguments (A), and $e_{(i,j)} \in E$ are edges. Let $In(i)$ be the set of nodes with an edge leading to n_i and similarly, $Out(i)$ be the set of nodes that n_i has an edge leading to. At a given iteration of the algorithm, we update the score of n_i as follows:

$$S(i) = \alpha \sum_{j \in In(i)} S(j) \times \frac{w(i,j)}{|Out(i)|} + (1 - \alpha) \tag{2.1}$$

where the value α is constant *damping factor*. The value of α is typically set to 0.85. The value of $w(i, j)$ is set to $w(i - j)$ for undirected graphs, and $w(i \rightarrow j)$ for directed graphs. The definition of $w(i - j)$ and $w(i \rightarrow j)$ is same as Section 2.7.2.

PageRank was originally devised for a query-induced graph and typically ap-

plied to a full graph. When constructing the bipartite graph, we limit the set of predicates P to only those words in the seed set, instead of including all words that can be predicates. Graph truncation eliminates the noise that can be introduced by predicates of the opposite polarity.

We next explore what is known as teleportation technique for topic sensitive PageRank, which is better at capture the relative "importance" of the network given a set of representative topics (Haveliwala (2002)). In our case, we include the learning bias in the network, we use the following equation that is slightly augmented from Equation 2.1.

$$S(i) = \alpha \sum_{j \in In(i)} S(j) \times \frac{w(i,j)}{|Out(i)|} + (1-\alpha)\,\epsilon_i \tag{2.2}$$

Here, the new term ϵ_i is a *smoothing factor* that prevents cliques in the graph from garnering reputation through feedback (Bianchini et al. (2005)). In order to emphasize important portion of the graph, i.e., subgraphs connected to the seed set, we assign non-zero ϵ scores to only those important nodes, i.e., the seed set. Intuitively, this will cause the random walk to restart from the seed set with $(1-\alpha)$ = 0.15 probability for each step. We also apply PageRank using all verbs as predicate candidates and all potential arguments as connotation candidates in order to rank both sets of words simultaneously. We apply the same weighting scheme for both ϵ and edges. While this method only requires a single graph for each positive and negative connotations, it also introduces more noise. Specifically, it is important to verify that words which occur in correspondence with both positively and negatively polarized predicates maintain similar rankings via this algorithm.

2.7.4 Constructing Lexicon

To classify the words to connotatively positive or negative, we run HITS / PageRank algorithms on the G^+ and G^- and obtain two ranked list for positiveness and negativeness respectively. Then we perform some simple post-processing on the ranked word lists: select the words ranked top N into the lexicon, if a word is ranked at top N in both list, then we label the word based on the better rank. N is selected empirically. We will show the evaluation results with respect to different N in the following section.

We start with 20 positive and 20 negative connotative predicates as seed words. $32,876$ unique words are inferred from the seed predicate. From this set, we find $9,072$ words are exclusively connected to positive predicates and $4,582$ words exclusively connected to negative predicates which are labelled positive and negative respectively. There are also $19,222$ words associated with both polarities. The size of the lexicon may vary depending on how we set the threshold for the polarized polarities.

2.7.5 Evaluations

Next, we will verify the effectiveness of semantic prosody for learning the general connotation. Since we obtain the ranking result via random walk algorithms, we compare $prec_\lambda$ % for three different segments of our lexicon, top N with $N = 100$ and $N = 1000$, and the entire lexicon. Results are shown in Table 2.6. Note that the numbers shown in Table 2.6 are very harsh representation of the actual quality of our connotation lexicon, because we counted those words with connotative polarity as incorrect if such words are marked as neutral by sentiment lexicons.

Table 2.4 and Table 2.5 present the new connotative words we find, including named entities. However, as can be seen in Table 2.4, many newly found connotative words are good, even if many of them are marked as neutral by conventional sentiment lexicon. It is worthwhile to remind that the *connotation lexicon* is by definition more polar than the *sentiment lexicon*.

Baseline We use a simple method dubbed *FREQ*, which uses co-occurrence frequency with respect to the seed predicates. Using the pattern $[p]$ $[\star]^{n-2}$ $[a]$, we collect two sets of n-gram records: one set using the positive connotative predicates, and the other using the negative connotative predicates. With respect to each set, we calculate the following for each word a,

- Given [a], the number of unique $[p]$ as $f1$
- Given [a], the number of unique phrases $[\star]^{n-2}$ as $f2$
- The number of occurrences of $[a]$ as $f3$

We then obtain the score σ_{a_+} for positive connotation and σ_{a_-} for negative connotation using the following equations that take a linear combination of $f1$, $f2$, and $f3$ that we computed above with respect to each polarity.

$$\sigma_{a_+} = \alpha \times \sigma_{f1_+} + \beta \times \sigma_{f2_+} + \gamma \times \sigma_{f3_+} \tag{2.3}$$

$$\sigma_{a_-} = \alpha \times \sigma_{f1_-} + \beta \times \sigma_{f2_-} + \gamma \times \sigma_{f3_-} \tag{2.4}$$

Note that the coefficients α, β and γ are determined experimentally. We assign positive polarity to the word a, if $\sigma_{a_+} >> \sigma_{a_-}$ and vice versa. Otherwise, the word

Positive: boogie, housewarming, persuasiveness, kickoff, playhouse, diploma, intuitively, monument, inaugurate, troubleshooter, accompanist
Negative: seasickness, overleap, gangrenous, suppressing, fetishist, unspeakably, doubter, bloodmobile, bureaucratized

Table 2.4: Examples of newly discovered words with connotations: these words are treated as neutral in some conventional sentiment lexicons.

POSITIVE	NEGATIVE
Mandela, Intel, Google, Python, Sony, Pulitzer, Harvard, Duke, Einstein, Shakespeare, Elizabeth, Swarovski, Clooney, Hoover, Goldman, Hawaii, Yellowstone, Zion, Lenox, Klein, Hollywood,RSA, IBM,	Katrina, Monsanto, Halliburton, Enron, Hiroshima, Holocaust, Afghanistan, Mugabe, Hutu, Saddam, Osama, Qaeda, Kosovo, Bolshevik, Britney, Helicobacter, HIV, Chernobyl, Alzheimers, Sclerotinia

Table 2.5: Example Named Entities (Proper Nouns) with Polar Connotation.

will be considered as neutral.

Evaluation I: Comparison against Sentiment Lexicon

Given that the connotation lexicon is learned only based on semantic prosody patterns in ngram corpus, the comparison results seem promising, especially for top ranked words which have a higher average frequency.

Evaluation II: Extrinsic Evaluation via Sentiment Analysis

Table 2.8 presents the performance for the sentiment classification task (as described in Section 2.5.2), the empirical study demonstrates that the practical value

	GENINQ			MPQA		
LEXICON	FREQ	HITS	PAGERANK	FREQ	HITS	PAGERANK
Top 100	73.6	77.7	77.0	83.0	86.3	87.2
Top 1000	67.8	68.8	68.5	80.3	81.3	80.3
Top MAX	65.8	66.5	65.7	71.5	72.2	72.3

Table 2.6: Comparison Result with Sentiment Lexicons (%)

Algorithm	**1st Round**		**2nd Round**	
	Acc.	F-val	Acc.	F-val
Voting	68.7	65.4	71.0	68.5
Bag of Words	69.9	65.1	69.9	65.1
(〃) + MPQA	74.7	75.0	74.7	75.0
BoW + Top 2,000	73.3	74.5	73.7	75.4
(〃) + MPQA	72.8	73.5	75.0	**77.6**
BoW + Top 6,000	**76.6**	**77.1**	74.5	75.3
(〃) + MPQA	74.1	73.5	**75.2**	76.0
BoW + Top 10,000	74,1	73.5	74.2	73.8
(〃) + MPQA	73.5	74.3	74.7	75.1

Table 2.7: SemEval Classification Result(%) — (〃) denotes that all features in the previous row are copied over.

of the connotation lexicon for sentiment analysis is encouraging, particularly for Twitter dataset, which is known to be very noisy. Notice that the use of Top 6,000 words from our connotation lexicon along with MPQA lexicon boost the performance up to 78.0%, which is significantly better than than 71.4% using only the conventional MPQA lexicon with $p < 0.001$. This result shows that our connotation lexicon nicely complements existing sentiment lexicon, improving practical sentiment analysis tasks.

Algorithm	1st Round		2nd Round	
	Acc.	F-val	Acc.	F-val
Voting	60.4	59.1	62.6	61.3
Bag of Words	69.9	72.1	69.9	72.1
(〃) + MPQA	70.3	71.4	70.3	71.4
BoW + Top 2,000	71.3	65.4	72.7	73.3
(〃) + MPQA	69.4	63.1	73.1	74.6
BoW + Top 6,000	**77.2**	69.0	76.4	77.6
(〃) + MPQA	76.4	72.0	**76.8**	**78.0**
BoW + Top 10,000	73.3	**73.5**	73.7	74.1
(〃) + MPQA	74.1	69.5	73.5	74.2

Table 2.8: Twitter Classification Result(%) — (〃) denotes that all features in the previous row are copied over.

2.7.6 Conclusion

In this section, we presented random work algorithms for learning connotation lexicon together with connotative predicates in a nearly unsupervised manner. Our approaches are grounded on the linguistic insight with respect to semantic prosody. Empirical study demonstrates the effectiveness of the selectional preference of connotative predicates based on the statistical information extracted from Google Web 1T data. Our results also show the practical value of the connotation lexicon for sentiment analysis encouraging further research in this direction.

2.8 Lexicon Induction via Label Propagation

In the previous section, we present our exploratory work on learning the general connotation with semantic prosody encoded random walk models and obtain some promising results. Due to the relatively small set of quality seeds of connotative

predicates available, we find that relying only on the predicate-argument structure of semantic prosody is somewhat limiting. Therefore, in this section, we further explore the lexical relations to enlarge the coverage. One basic assumption behind the graph-based algorithm / graph representation is the *cluster assumption* (Learning (2003)), which states that two points are likely to have the same class label if there is a path connecting them through the region of high density. Therefore, in addition to the semantic relation between connotative predicates and argument, we also explore the semantic similarities between unlabeled words. For this, we consider another linguistic pattern - semantic parallelism of coordination.

Inspired by previous success of applying graph-based algorithm to various fields (Zhu (2006)), we experiment with label propagation (or graph propagation) by Velikovich et al. (2010) for inducing the connotative labels. Comparing to the ranking algorithm HITS / PageRank, it allows to integrate the two bipartite graphs G^+ and G^-. We will compare the results by two algorithms based on the same graph structure.

2.8.1 Graph Representation

We extend the graph structure of predicate – argument with the pairwise relations between unlabeled words (argument–argument) as an *overlay of two sub-graphs*. Figure 2.3 depicts this structure, where the sub-graph circumscribed as "Pred-Arg" (a bipartite graph) corresponds to the former, and the sub-graph marked as "Arg-Arg" (a monopartite graph) corresponds to the latter. Each of these graph components are described more in details below.

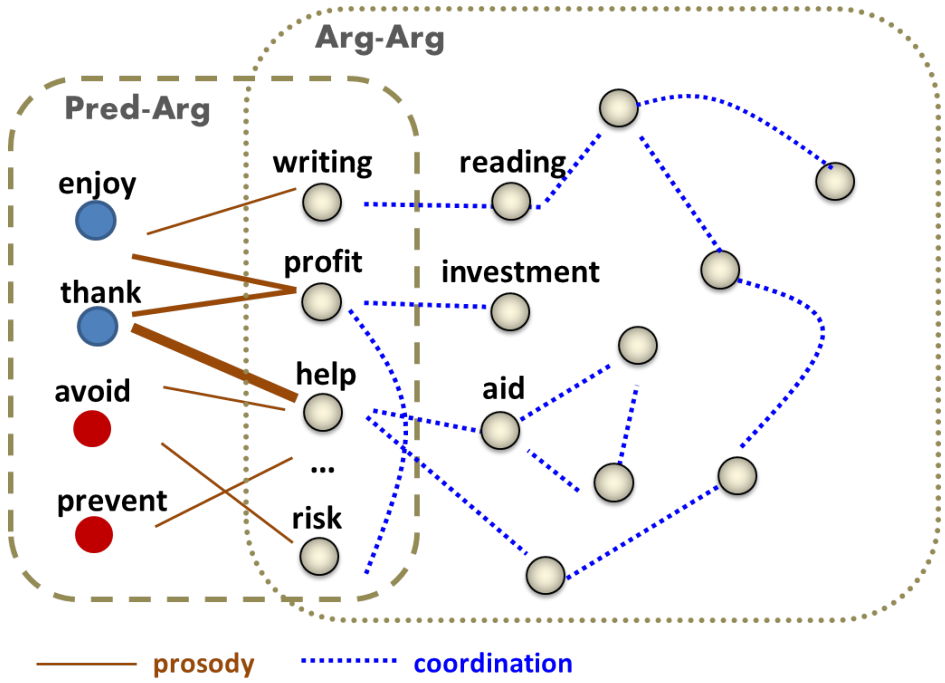


Figure 2.3: Graph Structure for Graph Propagation.

Sub-graph #1: Predicate–Argument Graph

This sub-graph is the bipartite graph that encodes the selectional preference of connotative predicates over their arguments.

Undirected (Symmetric) Graph First we explore undirected edges. In this case, we assign a weight for each undirected edge between a predicate p and an argument a. Intuitively, the weight should correspond to the strength of relatedness or association between the predicate p and the argument a. We use Pointwise Mutual Information (PMI) as we mentioned in the Section 2.4. The PMI score between p and a is defined as follows:

$$w(p-a) := PMI(p,a) = log\frac{P(p,a)}{P(p)P(a)}$$

The log of the ratio is positive when the pair of words tends to co-occur and negative when the presence of one word correlates with the absence of the other word.

Directed (Asymmetric) Graph Next we explore directed edges. That is, for each connected pair of a predicate p and an argument a, there are two edges in opposite directions: $e(p \rightarrow a)$ and $e(a \rightarrow p)$. In this case, we explore the use of asymmetric weights using conditional probability. In particular, we define weights as follows:

$$w(p \rightarrow a) := P(a|p) = \frac{P(p,a)}{P(p)}$$

$$w(a \rightarrow p) := P(p|a) = \frac{P(p,a)}{P(a)}$$

Empirically, when overlaid with the second sub-graph, we found that it is better to keep the connectivity of this sub-graph as uni-directional. That is, we only allow edges to go from a predicate to an argument.

Sub-graph #2: Argument–Argument Graph

The second sub-graph is extended based on the argument words in Sub-graph #1. In this way, we include more unseen words that are semantically associated with the argument words.

To construct the overlay graph, one possible way is to simply connect all nodes and assign edge weights proportionate to the word association scores. We find

that such a completely connected graph can be susceptible to propagating noise, and does not scale well over a very large set of vocabulary. One option is to consider to trim the graph either by retaining only those edges that are above a certain threshold, or by limiting the size of neighboring nodes based on ordered edge weights. The cost of graph construction for either option can get very high however, as it is necessary to compute the edge score of all possible pairs of words. We therefore reduce the graph connectivity by focusing on words with relatively strong association.

There are several ways to quantify the semantic similarity between unlabeled words. Instead of adopting the conventional pairwise relation between two words (such as PMI, conditional probabilities), we propose to include the information of the neighbors of the word. Specifically, we calculate the distributional similarities over coordination (e.g., Bock (1986); Hatzivassiloglou and McKeown (1997); Pickering and Branigan (1998)) among the argument words and their semantically associated words. The computation of distributional similarity is described in Section 2.4. This cosine similarity represents the semantic similarity of the two argument words based on their occurrence in the $a_i and a_j$ pattern. It is important to judiciously select edge connections and corresponding weights in order to reduce the mishap of propagating noise, a common problem in many unsupervised and bootstrapping approaches. Then, by filtering words using this pattern, we expect the cosine similarity to convey stronger connotative similarity, and hence increase the quality of connotation lexicon.

Algorithm 1: GRAPH PROPAGATION

Input: Connotation graph G=(V, P, E);
Output: Connotation label probabilities for each node $i \in V \backslash P$
Initialize:
foreach do
    $i \in V \backslash P$, $pol_i^+ = 0$, $pol_i^- = 0$.
foreach do
    $i \in P^+$, $pol_i^+ = 1$; $i \in P^-$, $pol_i^- = 1$
. **foreach** $i \in V$ **do**
    if $i = j$ **then** $\alpha_{i,j} = 1$
    else $\alpha_{i,j} = 0$
foreach $v_i \in P$ **do**
    $F = \{v_i\}$
    foreach t *:* $1...T$ **do**
        foreach $(v_k, v_j \in E)$ *s.t.* $v_k \in F$ **do**
            $\alpha_{i,j} = max\{\alpha_{i,j}, \alpha_{i,k} \cdot w_{k,j}\}$
            $F = F \cup \{v_j\}$
repeat*steps 11-16*
    $\beta = \sum_i pol_i^+ / \sum_i pol_i^-$
until using N to compute pol
foreach $i \in V \backslash P$ **do**
    if $|pol_i| < \gamma$ **then** $pol_i = 0$
    else $pol_i = pol_i^+ - \beta pol_i^i$

2.8.2 Propagating over the Overlay of Two Sub-graphs

Note that the two sub-graphs described earlier are based on different types of weights on their edges: PMI for predicate-argument graph and distributional similarity for argument-argument graph. Although it is not impossible to use PMI scores for the second sub-graph, as will be shown later, we found that cosine similarity works significantly better than PMI empirically. This is not surprising, as the second sub-graph serves the purpose of discovering words with similar con-

notative polarity that appear in the semantic parallelism. Also note that the cosine similarity does not make sense for the predicate-argument patterns, as a predicate and an argument are not necessarily similar in term of their probability distribution: they typically either belong to different types of part-of-speech or represent very different semantic concepts. We normalize the PMI scores of the predicate-argument graph to the range of cosine similarity scores as below:

$$w(p \rightarrow a) = \quad (PMI(p,a) - PMI_{MIN}) \times \frac{DS_{MAX} - DS_{MIN}}{PMI_{MAX} - PMI_{MIN}} + DS_{MIN} \quad (2.5)$$

where $PMI(p,a)$ denotes the PMI score of an edge from p to a. PMI_{MIN} and PMI_{MAX} denote the maximum and minimum value of PMI respectively within the predicate-argument graph, and DS_{MAX} and DS_{MIN} are the maximum and minimum value of the distributional similarity scores.

We use the graph propagation of Velikovich et al. (2010) to propagate the connotation of seed words throughout this network of words, this model is a variant of label propagation (Zhu and Ghahramani (2002)) that has been used for various NLP tasks for semi-supervised learning and bootstrapping (e.g., Niu et al. (2005); Rao and Ravichandran (2009)) as described in Algorithm 1. In our application, the iterative process converges to a fixed point. Remind that the previous of approach - random work, the inference of words with positive connotation and negative connotation is isolated from each other. While label propagation algorithm induces the distribution over positive and negative connotation of each word. It naturally enables each word vertex to get prior knowledge from all the seeds of both polarities when a path exists between the node and a seed. As will be shown through a series of comparative experiments, it is important to judiciously select edge con-

nections and corresponding weights in order to reduce the mishap of propagating noise, a common problem in many unsupervised and bootstrapping approaches.

2.8.3 Evaluations

We perform two types of evaluations: one is to compare the resulting lexicon with sentiment lexicon as described in Section 2.5.1; one is to utilize the lexicon for sentiment classification tasks as described in Section 2.5.2. We will compare the performances of two kinds of graph-based algorithms.

Evaluation I: Comparison against Sentiment Lexicon

As shown through a series of comparative experiments in Table 2.9, the results show that the graph propagation approach based on the transduction algorithm consistently improves the performance comparing to random work model with bipartite graph. The use of label propagation alone (PRED-ARG (CP), PRED-ARG (PMI)) improves the performance substantially over the comparable graph construction with different graph analysis algorithms, in particular, HITS and PageRank approaches in the previous section. The OVERLAY achieves the best performance among graph-based algorithms, significantly improving the precision over all other baselines listed in the table. At the same time, we also notice that the coverage of the lexicon is substantially larger than that of all other alternatives. This result suggests:

1 The sub-graph #2, based on the semantic parallelism of coordination, is simple and yet very powerful as an inductive bias.

	GENINQ EVAL				MPQA EVAL			
	100	1,000	5,000	ALL	100	1,000	5,000	ALL
OVERLAY	97.0	**95.1**	78.8	78.3	**98.0**	**93.4**	82.1	77.7
PRED-ARG (PMI)	91.0	91.4	76.1	76.1	88.0	89.1	78.8	75.1
PRED-ARG (CP)	88.0	85.4	76.2	76.2	87.0	82.6	78.0	76.3
HITS	77.0	68.8	68.8	66.5	86.3	81.3	81.3	72.2
PAGERANK	77.0	68.5	68.5	65.7	87.2	80.3	80.3	72.3

Table 2.9: Evaluation of the Induction Algorithms with respect to Sentiment Lexicons (prec%).

2 The performance of graph propagation varies significantly depending on the graph topology and the corresponding edge weights.

Evaluation II: Extrinsic Evaluation via Sentiment Analysis

Next, we apply the connotation lexicon for the sentence-level sentiment classification task. CONNOTATION (OVERLAY) corresponds to the lexicon based on the overlay graph structure and CONNOTATION (PRED-ARG) corresponds to the best performing lexicon by the predicate-argument structure in Table 2.9. In Table 2.10, "$\lessgtr n$" indicates that the positive set consists of the texts with scores $\geq n$ and the negative set consists of the texts with scores $\leq -n$. More details about the evaluation is described in Section 2.5.2. CONNOTATION (OVERLAY) performs significantly better than the other connotation lexicon and the commonly used sentiment lexicons, which shows that the potential practical use of connotation lexicon. Please note that we will refer to the best performing connotation lexicon by graph propagation as "connotation (GP)" in the rest of the paper.

LEXICON	DATA				
	TWEET	SEMEVAL			
		$\leqslant$20	$\leqslant$40	$\leqslant$60	$\leqslant$80
CONNOTATION (OVERLAY)	68.5	70.0	72.9	76.8	89.6
CONNOTATION (PRED-ARG)	60.5	64.2	69.3	70.3	79.2
SENTIWN	67.4	61.0	64.5	70.5	79.0
GI+MPQA	65.0	64.5	69.0	74.0	80.5

Table 2.10: Accuracy on Sentiment Classification (%)

2.8.4 Discussion of Graph-based Algorithms

Although graph-based algorithms (2.7, 2.8) provide an intuitive framework to incorporate various lexical relations, we recognize a few fundamental limitations for our task:

1. They allow only *non-negative* edge weights. Therefore, we can encode only positive (supportive) relations among words (e.g., distributionally similar words will endorse each other with the same polarity), while missing on negative relations (e.g., antonyms may drive each other into the opposite polarity).

2. They induce positive and negative polarities in isolation via separate graphs. However, we expect that a more effective algorithm should induce both polarities simultaneously. On a related note, it is practically awkward to model neutral polarity from graph-based algorithms.

3. The framework does not readily allow incorporating a diverse set of *soft* and *hard* constraints, and it is not easy or even possible to incorporate various types of *hard* constraints. Therefore, the kinds of prior knowledge we can incorporate into graph-based algorithms is limited (so far we have not

encoded the synonyms and antonym relation into the graph).

2.8.5 Conclusion

In this section, we exploit another graph structure, which is an overlay of predicate-argument and argument-argument structure. Then we apply graph propagation algorithm for the inducing connotative labels. We find that the graph propagation algorithm performs better than the random walk algorithms when running on the predicate-argument graph structure. By including the argument-argument lexical relation, we are able to expand the coverage while improving the quality of the lexicon. In addition, we also observe a few limitations of the two graph-based algorithms, therefore, we will further explore different types of algorithms to address the limitations in the following section.

2.9 Lexicon Induction via Constraint Optimization

Motivated by the previous experiment results and observations of the connotation lexicon induction via the two graph-based algorithms in Section 2.7 and Section 2.8, we plan to seek an alternative approach that is relatively convenient to incorporate even more diverse linguistic insights. One possible approach is to formulate the task as an optimization problem and encode the prior knowledge as either soft or hard constraints. In this section, we develop an integer linear programing frameworks for learning the connotative labels. In the evaluation section, we will compare the quality and performance of the lexicon by constraint optimization with previous versions.

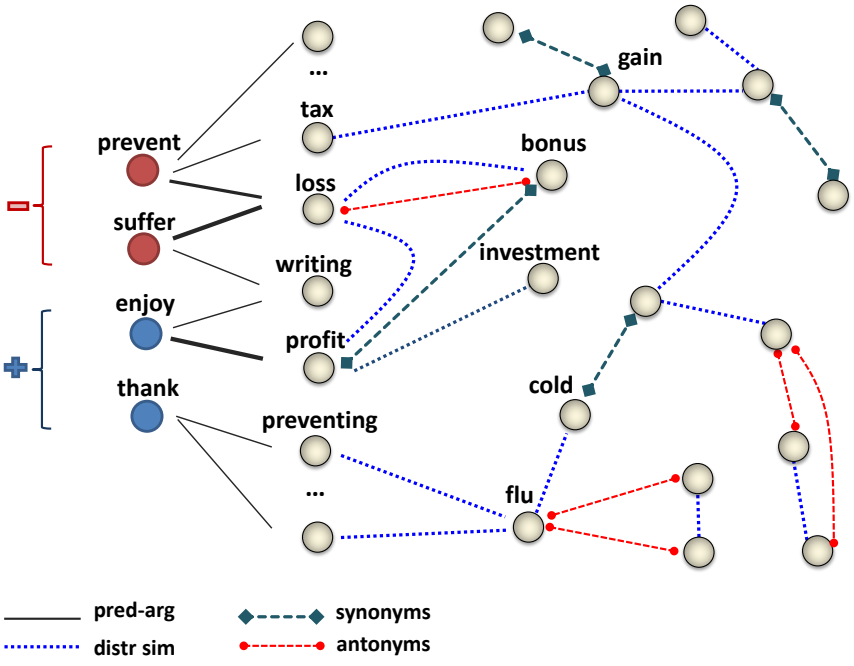


Figure 2.4: Graph Structure for Linear Programming

2.9.1 Graph Representation

Previously, we have been focusing on the linguistic statistics derived from Web resources, next we consider to include the knowledge readily available in the classical dictionary-driven resources. Figure 2.4 depicts the graph structures that encoded assorted of lexical relations introduced in Section 2.4

- For directed edges, the value of $w(i,j)$ is set to $w(i \rightarrow j)$ is defined as

$$w(p_i \rightarrow a_j) := P(a_j|p_i) = \frac{P(p_i, a_j)}{P(p_i)}$$

$$w(a_j \rightarrow p_i) := P(p_i|a_j) = \frac{P(p_i, a_j)}{P(a_j)}$$

$P(p, a)$ - the count of occurrence of p and a. $P(p)$ ($P(a)$) - the frequency of p (a).

- For the edges between unlabeled words, if the pair appears at the parallelism or coordination pattern, we define the wedge weight as distributional similarity between a_i and a_j.

- For the synonyms or antonyms of a node based on WordNet synsets, the unseen ones will be included in the graph and the corresponding edge weight is assigned as 1 for synonyms, or -1 for antonyms.

Hence, in comparison to Figure 2.3, two new components are included: (1) dictionary-driven relations of synonyms and antonyms readily available at WordNet (Miller (1995)) , and (2) dictionary-driven words (i.e., unseen words with respect to those relations explored in Figure 2.3).

2.9.2 Induction using Integer Linear Programming

We define the problem in terms of a collection of discrete random variables representing connotative labels (positive, negative and neutral) of words. Then we formulate linguistically motivated insights in Figure 2.4 using Integer Linear Programming (ILP) as follows:

Notation / Definition of sets of words:

1. $\mathcal{P}^+$: the set of positive seed predicates.
   $\mathcal{P}^-$: the sets of negative seed predicates.

2. $\mathcal{A}$: the set of seed sentiment words.

3. $\mathcal{R}^{syn}$: word pairs in synonyms relation.
   $\mathcal{R}^{ant}$: word pairs in antonyms relation.
   $\mathcal{R}^{coord}$: word pairs in coordination relation.
   $\mathcal{R}^{prosody}$: word pairs in pred–arg relation.

Notation / Definition of variables: For each word i, we define binary variables $x_i, y_i, z_i \in \{0, 1\}$, where $x_i = 1$ ($y_i = 1$, $z_i = 1$) iff. i has a positive (negative, neutral) connotation respectively. For every pair of word i and j, we define binary variables d_{ij}^{pq} where $p, q \in \{+, -, 0\}$ and $d_{ij}^{pq} = 1$ iff. the polarity of i and j are p and q respectively.

Next, we seek optimal assignments to the variables in the presence of the constraints on the word relations as follows.

Objective function: We aim to maximize:

$$F = \Phi^{prosody} + \Phi^{coord} + \Phi^{neu}$$

where $\Phi^{prosody}$ is the scores based on semantic prosody, Φ^{coord} captures the distributional similarity over coordination, and Φ^{neu} controls the sensitivity of connotation detection between positive (negative) and neutral. In particular,

$$\Phi^{prosody} = \sum_{i,j}^{\mathcal{R}^{pred}} w_{i,j}^{pred} \left(d_{i,j}^{++} + d_{i,j}^{--} - d_{i,j}^{+-} - d_{i,j}^{-+}\right) \quad (2.6)$$

$$\Phi^{coord} = \sum_{i,j}^{\mathcal{R}^{coord}} w_{i,j}^{coord} \left(d_{i,j}^{++} + d_{i,j}^{--} + d_{i,j}^{00}\right) \quad (2.7)$$

$$\Phi^{neu} = \alpha \sum_{i} \cdot z_j \sum_{i,j}^{\mathcal{R}^{pred}} w_{i,j}^{pred} \quad (2.8)$$

Soft constraints for edge weights: The weights in the objective function given above are set as follows:

$$w^{pred}(p,a) = \frac{freq(p,a)}{\sum_{(p,x)\in\mathcal{R}^{pred}} freq(p,x)} \quad (2.9)$$

$$w^{coord}(a_1,a_2) = CosSim(\overrightarrow{a_1},\overrightarrow{a_2}) = \frac{\overrightarrow{a_1}\cdot\overrightarrow{a_2}}{||\overrightarrow{a_1}||\ ||\overrightarrow{a_2}||} \quad (2.10)$$

Note that the same $w^{coord}(a_1, a_2)$ has been used in graph propagation described in Section 2.2. α controls the sensitivity of connotation detection such that higher value of α will promote neutral connotation over polar ones.

Hard constraints for variable consistency:

1. Each word i has one of $\{+, -, o\}$ as polarity:
   $\forall i,\ x_i + y_i + z_i = 1$

2. Variable consistency between d_{ij}^{pq} and x_i, y_i, z_i:

$$x_i + x_j - 1 \leq 2d_{i,j}^{++} \leq x_i + x_j \tag{2.11}$$

$$y_i + y_j - 1 \leq 2d_{i,j}^{--} \leq y_i + y_j \tag{2.12}$$

$$z_i + z_j - 1 \leq 2d_{i,j}^{00} \leq z_i + z_j \tag{2.13}$$

$$x_i + y_j - 1 \leq 2d_{i,j}^{+-} \leq x_i + y_j \tag{2.14}$$

$$y_i + x_j - 1 \leq 2d_{i,j}^{-+} \leq y_i + x_j \tag{2.15}$$

Given that,

$d_{i,j}^{++} = 1$ iff i and j are both positive.

$d_{i,j}^{--} = 1$ iff i and j are both negative.

$d_{i,j}^{+-} = 1$ iff i and j are both positive.

$d_{i,j}^{00} = 1$ iff i and j are both neutral.

$sum_{i,j} = d_{ij}^{+} + d_{ij}^{-} + d_{ij}^{0}$. $sum_{i,j} = 1$ iff i, j have the same polarity; otherwise $sum_{i,j} = 0$.

Determine whether i, j have the opposite polarity:

$d_{i,j}^{+-}, d_{i,j}^{-+} \in \{0, 1\}$.

$d_{i,j}^{-+} = 1$ iff i and j are both negative.

$sum_{i,j} = d_{ij}^{+-} + d_{ij}^{-+}$. $sum_{i,j} = 1$ iff i, j have the opposite polarity; otherwise $sum_{i,j} = 0$.

Determine whether one of the word is neutral. $\sum d_{ij}^{*} = 0$. Restraints on the number of polar words: $\sum x_i = 1000$. $\sum y_i = 1000$.

Hard constrains for WordNet relations:

1. $\mathcal{C}^{ant}$: Antonym pairs will not have the same positive or negative polarity:

$$\forall(i,j) \in \mathcal{R}^{ant},\ x_i + x_j \leq 1,\ y_i + y_j \leq 1$$

2. $\mathcal{C}^{syn}$: Synonym pairs will not have the opposite polarity:

$$\forall(i,j) \in \mathcal{R}^{syn},\ x_i + y_j \leq 1,\ x_j + y_i \leq 1$$

One practical problem with ILP is efficiency and scalability. In particular, we found that it becomes nearly impractical to run the ILP formulation including all words in WordNet plus all words in the argument position in GoogleWeb1T. We therefore explore an alternative formulation based on Linear Programming as described below.

2.9.3 Induction using Linear Programming

One straightforward option for Linear Programming formulation may seem like using the same Integer Linear Programming formulation introduced in Section 2.9.2, only changing the variable definitions to be real values [0, 1] rather than integers. However, because the hard constraints in ILP are defined based on the assumption that all the variables are binary integers, those constraints are not applicable to IL when considered for real numbers. Therefore we revise those hard constraints to encode various semantic relations (WordNet and semantic coordination) more directly.

Definition of variables: For each word i, we define variables $x_i, y_i, z_i \in [0, 1]$. i has a positive (negative) connotation if and only if the x_i (y_i) is assigned the greatest value among the three variables; otherwise, i is neutral.

For the consistency for the word relations as introduced in Section §2.9.2, we define determinative variables $d^*_{i,j}$, where $d^*_{i,j} \in [-1, 0]$. W^{syn}, W^{ant} are positive constants, here assigned 1.

Objective function: We aim to maximize:

$$F = \Phi^{prosody} + \Phi^{coord} + \Phi^{syn} + \Phi^{ant} + \Phi^{neu}$$

In particular,

$$\Phi^{prosody} = \sum_{i,j}^{\mathcal{R}^{pred^+}} w^{pred}_{i,j} \cdot x_j + \sum_{(i,j)}^{\mathcal{R}^{pred^-}} w^{pred}_{i,j} \cdot y_j \quad (2.16)$$

$$\Phi^{coord} = \sum_{i,j}^{\mathcal{R}^{coord}} w^{coord}_{i,j} \cdot (dc^{++}_{i,j} + dc^{--}_{i,j}) \quad (2.17)$$

$$\Phi^{syn} = W^{syn} \sum_{i,j}^{\mathcal{R}^{syn}} (ds^{++}_{i,j} + ds^{--}_{i,j}) \quad (2.18)$$

$$\Phi^{ant} = W^{ant} \sum_{i,j}^{\mathcal{R}^{ant}} (da^{++}_{i,j} + da^{--}_{i,j}) \quad (2.19)$$

$$\Phi^{neu} = \alpha \sum_{i} \cdot z_j \sum_{i,j}^{\mathcal{R}^{pred}} w^{pred}_{i,j} \quad (2.20)$$

Hard constraints We add penalties to the objective function if the polarity of a pair of words is not consistent with its corresponding syntax or semantic relation. For example, for synonyms i and j, we introduce $ds^{++}_{i,j}, ds^{--}_{i,j} \in [-1, 0]$ and W^{syn}

	GENINQ EVAL				MPQA EVAL			
	100	1,000	5,000	ALL	100	1,000	5,000	ALL
ILP	**97.6**	94.5	**84.5**	80.4	**98.0**	89.7	**84.6**	**78.4**
LP	**97.6**	94.5	**84.5**	80.4	**98.0**	89.7	**84.6**	**78.4**
GP	97.0	**95.1**	78.8	78.3	**98.0**	**93.4**	82.1	77.7
HITS	77.0	68.8	68.0	66.5	86.3	81.3	78.5	72.2
PAGERANK	77.0	68.5	67.8	65.7	87.2	80.3	77.6	72.3

Table 2.11: Evaluation of Induction Algorithms with respect to Sentiment Lexicons (prec%).

(a positive constant). Then we set the upper bound of $ds_{i,j}^{++}$ ($ds_{i,j}^{--}$) as the distance of x_i and x_j (y_i and y_j), then the penalty $W^{syn} \cdot (ds_{i,j}^{++} + ds_{i,j}^{--})$ is encouraged to keep small. Consequently, i and j are induced to have the same label.

The constraints for $\mathcal{R}^{syn}$ and $\mathcal{R}^{ant}$ are shown as follows. In a similar manner to $\mathcal{R}^{syn}$, we define the constraints for $\mathcal{R}^{coord}$.

For $(i, j) \in \mathcal{R}^{syn}$,

$$ds_{i,j}^{++} \leq x_i - x_j \quad , \quad ds_{i,j}^{--} \leq y_i - y_j$$
$$ds_{i,j}^{++} \leq x_j - x_i \quad , \quad ds_{i,j}^{--} \leq y_j - y_i$$

For $(i, j) \in \mathcal{R}^{ant}$,

$$da_{i,j}^{++} \leq x_i - (1 - x_j) \quad , \quad da_{i,j}^{--} \leq y_i - (1 - y_j)$$
$$da_{i,j}^{++} \leq (1 - x_j) - x_i \quad , \quad da_{i,j}^{--} \leq (1 - y_j) - y_i$$

To solve the ILP/LP, we run ILOG CPLEX Optimizer CPLEX (2009)) on a 3.5GHz 6 core CPU machine with 96GB RAM. Efficiency-wise, LP runs within

FORMULA	POSITIVE			NEGATIVE			ALL		
	R	P	F	R	P	F	R	P	F
INTEGER LINEAR PROGRAMING (ILP)									
$\Phi^{prosody}$	57.4	82.5	67.7	48.9	84.3	61.9	53.1	83.4	64.9
$\Phi^{prosody} + \Phi^{coord}$	42.0	87.6	56.8	38.1	64.6	47.9	40.0	76.1	52.4
$\Phi^{prosody} + \mathcal{C}^{syn/ant}$	51.4	85.7	64.3	44.7	87.9	59.3	48.0	86.8	61.8
$\Phi^{prosody} + \mathcal{C}^{syn/ant} + \mathcal{C}^{\mathcal{A}}$	61.2	93.3	73.9	52.4	**92.2**	66.8	56.8	**92.8**	70.5
$\Phi^{prosody} + \Phi^{coord} + \mathcal{C}^{syn/ant}$	67.3	75.0	70.9	53.7	84.4	65.6	60.5	79.7	68.8
$\Phi^{prosody} + \Phi^{coord} + \mathcal{C}^{syn/ant} + \mathcal{C}^{\mathcal{A}}$	62.2	**96.0**	75.5	51.5	89.5	65.4	56.9	**92.8**	70.5
LINEAR PROGRAMING (LP)									
$\Phi^{prosody}$	58.3	81.8	68.1	49.6	84.0	62.4	54.0	82.9	65.4
$\Phi^{prosody} + \Phi^{coord}$	21.0	72.7	32.6	31.4	80.1	45.1	26.2	76.4	39.0
$\Phi^{prosody} + \Phi^{syn/ant}$	24.4	76.0	36.9	23.6	78.8	36.3	24.0	77.4	36.6
$\Phi^{prosody} + \Phi^{syn/ant} + \Phi^{\mathcal{A}}$	71.6	87.8	78.9	68.8	84.6	75.9	70.2	86.2	77.4
$\Phi^{prosody} + \Phi^{coord} + \Phi^{syn/ant}$	67.9	92.6	78.3	64.6	89.1	74.9	66.3	90.8	76.6
$\Phi^{prosody} + \Phi^{coord} + \Phi^{syn/ant} + \Phi^{\mathcal{A}}$	**78.6**	90.5	**84.1**	**73.3**	87.1	**79.6**	**75.9**	88.8	**81.8**

Table 2.12: ILP/LP Comparison on MQPA′ (%).

10 minutes while ILP takes hours.

2.9.4 Evaluations

In parallel with previous studies in Section 2.7 and Section 2.8, we conduct the evaluations: (1) comparison with sentiment lexicon (as in Section 2.5.1); (2) sentiment classification tasks based on connotation lexicon (as in Section 2.5.2). In addition, we also compare the best performing lexicon with human annotations.

Evaluation I: Comparison against Sentiment Lexicon

Table 2.12 shows the results evaluated against MPQA for different variations of ILP and LP. We compare the connotation lexicons by different ILP/LP varia-

tions against MPQA[5] Wiebe et al. (2005). When incorporating seed arguments, as indicated by $\mathcal{C}^A$/Φ^A in Table 2.12, we only use General Inquirer Stone and Hunt (1963). The results in Table 2.12 support our hypothesis that it could be advantageous to consider a wide range of linguistic-motivated knowledge. We find that LP variants much better recall and F-score, while maintaining comparable precision. LP versions can achieve comparable precision, higher F-score comparing to ILP versions. Note that the recall is very low for $\Phi^{prosody} + \Phi^{syn} + \Phi^{ant}$ as the solver tends to assign 0.5 to x_i and y_i. But after we add seed arguments Φ^A, both recall and precision are significantly enhanced. In addition, as shown through a series of comparative experiments in Table 2.12, the use of rich lexical resource generally helps improving the performance. The best performance is achieved when we incorporate all lexical resources. Hence, we choose the connotation lexicon by the best performing LP in the rest evaluations.

Evaluation II: Extrinsic Evaluation via Sentiment Analysis

Next, we verify the effectiveness of the different versions of connotation lexicon for the task of sentence-level sentiment classification. In Table 2.13 "$\lessgtr n$" indicates that the positive set consists of the texts with scores $\geq n$ and the negative set consists of the texts with scores $\leq -n$. More details about the evaluation is described in Section 2.5.2.

As shown in Table 2.13, CONNOTATION (CO) generally performs better than the other lexicons on both corpora. Considering that only very simple classification strategy is applied, the result by the connotation lexicon is quite promising.

[5]MPQA′ is the set of words in MPQA excluding the words in General Inquirer

LEXICON	DATA				
	TWEET	SEMEVAL			
		$\leqslant$20	$\leqslant$40	$\leqslant$60	$\leqslant$80
CONNOTATION (CO)	**70.1**	**70.8**	**74.6**	**80.8**	**93.5**
CONNOTATION(GP)	68.5	70.0	72.9	76.8	89.6
SENTIWN	67.4	61.0	64.5	70.5	79.0
GI+MPQA	65.0	64.5	69.0	74.0	80.5

Table 2.13: Accuracy on Sentiment Classification (%).

Evaluation III: Intrinsic Evaluation via Human Judgment

We evaluate 4000 words using Amazon Mechanical Turk. We choose words that are not already in GI+MPQA. We only consider the relatively common words so that the human annotators likely have good knowledge about the usage of the words. For this, we obtain most frequent 10,000 words based on the unigram frequency in Google-Ngram, then randomly select 4000 words. In this section, we compare the best performing connotation lexicon in the previous two evaluations against the human annotation.

- Ω^{Vote}: The judgement of each Turker is mapped to $x \in \{pos, neg, neu\}$, then take the majority vote.
- Ω^{Score}: Let $\sigma(i)$ be the sum (weighted vote) of the scores given by 5 judges for word i.

$$l(i) = \begin{cases} positive & \text{if } \sigma(i) > 1 \\ negative & \text{if } \sigma(i) < -1 \\ neutral & \text{if } -1 \leq \sigma(i) \leq 1 \end{cases}$$

The resulting distribution of judgements is shown in Table 2.3 & 2.14. Interestingly, we observe that *among the relatively frequently used English words, there*

	POS	NEG	NEU	UNDETERMINED
Ω^{Vote}	50.4	14.6	24.1	10.9
Ω^{Score}	67.9	20.6	11.5	n/a

Table 2.14: Distribution of Connotative Polarity from AMT.

	CONNOTATION	SENTIWORDNET	HUMAN JUDGES
Ω^{Vote}	77.0	71.5	66.0
Ω^{Score}	73.0	69.0	69.0

Table 2.15: Agreement (Accuracy) against AMT-driven Gold Standard.

are overwhelmingly more positively connotative words than negative ones.

In Table 2.15, we show the percentage of words with the same label over the mutual words by the two lexicon. The highest agreement is 77% by connotation lexicon and the gold standard by AMT^{Vote}. How good is this? It depends on what is the natural degree of agreement over subtle connotation among people. Therefore, we also report the degree of agreement among human judges in Table 7, where we compute the agreement of one Turker with respect to the gold standard drawn from the rest of the Turkers, and take the average across over all five Turkers[6]. Interestingly, the performance of Turkers are not as good as that of connotation lexicon! We conjecture that this could be due to *generally varying perception of different people on the connotative polarity, and that the corpus-driven induction algorithms are effective in learning the general connotative polarity from a large scale text.*

[6]In order to draw the gold standard from the 4 remaining Turkers, we consider adjusted versions of Ω^{Vote} and Ω^{Score} schemes described above.

2.9.5 Conclusion

In this section, we develop linear programming framework with the linguistically motivated constraints to solve the problem. Via comprehensive evaluations, we provide empirical insights into different variations of the induction algorithms, and examine the performance in terms of precision, coverage and efficiency. The results show that it could be considerably beneficial to incorporate a wide spectrum of linguistic insights into the induction algorithm. Therefore, in the following section, we mainly use the graph structure encoded with all proposed linguistic insights.

2.10 Sense-level Connotation

In the Section 2.7, Section 2.8 and Section 2.9, we present automatic methods for learning subtle shades of sentiment a word may conjure (e.g., Feng et al. (2011, 2013)), the resulting lexical corpus includes even those seemingly objective words such as *"sculpture"*, *"Ph.D."*, *"rosettes"*, which generally have not been considered as subjective words per their denotational meanings. In this section, we present our work address one practical problem - learning the connotation polysemous words. This word sense issue has been a universal challenge for a range of Natural Language Processing applications, including sentiment analysis. Recent studies have shown that it is fruitful to tease out subjectivity and objectivity corresponding to different senses of the same word, in order to improve computational approaches to sentiment analysis (e.g. Pestian et al. (2012); Mihalcea et al. (2012); Balahur et al. (2014)). Inspired by these recent successes, we further investigate whether we can attain similar gains if we model the connotative

polarity of different senses separately. Although there have been studies that have studied *sense-level sentiment* lexicons, e.g., SentiWordNet (Esuli and Sebastiani (2006)), no prior work has studied *sense-level connotation* lexicons. Additionally, in contrast to previous studies (for both sentiment and connotation lexicons) that learn the lexical polarity of either the word-level or sense-level respectively, we introduce a unified framework to learning a connotation lexicon (i.e., a lexicon of connotative polarity) over the network of *words* in conjunction with *senses* (Kang et al. (2014)).

For non-polysemous words, which constitute a significant portion of English vocabulary, learning the *general* connotation at the *word-level* (rather than at the *sense-level*) would be a natural operational choice. However, for polysemous words, which correspond to most frequently used words, it would be an overly crude assumption that the same connotative polarity should be assigned for all senses of a given word. For example, consider *"abound"*, for which lexicographers of WordNet prescribe two different senses, which by and large have not been considered as subjective words per their denotational meanings.

- (v) **abound**: (be abundant of plentiful; exist in large quantities)
- (v) **abound, burst, bristle**: (be in a state of movement or action) *"The room abounded with screaming children"*; *"The garden bristled with toddlers"*

For the first sense, which is the most commonly used sense for *"abound"*, the general overtone of the connotation would seem positive. That is, although one can use this sense in both positive and negative contexts, this sense of *"abound"* seems to collocate more often with items that are good to be abundant (e.g., *"resources"*), than unfortunate items being abundant (e.g., *"complaints"*).

However, for the second sense, which *"burst"* and *"bristle"* can be used interchangeably with respect to this particular sense, the general overtone is slightly more negative with a touch of unpleasantness, or at least not as positive as that of the first sense.Hence a *sense* in WordNet is defined by *synset (= synonym set)*, which is the set of words sharing the same sense. Especially if we look up the WordNet entry for *"bristle"*, there are noticeably more negatively connotative words involved in its gloss and examples.

There is one potential practical issue we would like to point out in building a sense-level lexical resource. End-users of such a lexicon may not wish to deal with Word Sense Disambiguation (WSD), which is known to be often too noisy to be incorporated into the pipeline with respect to other NLP tasks. As a result, researchers often need to aggregate labels across different senses to derive the word-level label. Although such aggregation is not entirely unreasonable, it does not seem to be the most optimal and principled way of integrating available resources, may not necessarily yield a more optimal polarity.

Hence, in this work, we present the first unified approach that learns *both* sense-level and word-level connotations at the same time. This way, end-users will have access to more accurate sense-level connotation labels if needed, while also having access to more general word-level connotation labels. We again formulate the lexicon induction problem as collective inference problem, this time, we explore the pairwise-Markov Random Fields (pairwise-MRF) and derive a loopy belief propagation algorithm for inference.

The key aspect of our approach is that we exploit the innate bipartite graph structure between words and senses encoded in WordNet. Although our approach seems conceptually natural, previous approaches, to our best knowledge, have

not directly exploited these relations between words and senses for the purpose of deriving lexical knowledge over words and senses collectively. In addition, previous studies (for both sentiment and connotation lexicons) aimed to produce only either of the two aspects of the polarity: word-level or sense-level, while we address both.

For this, we introduce to apply loopy belief propagation (loopy-BP) as a lexicon induction algorithm. As will shown in section 2.10.4, Loopy-BP in our study achieves statistically significantly better performance over the constraint optimization approaches previously explored in Section 2.9. Here we use the probabilistic representation of pairwise-MRF and Loopy-BP as inference technique, we can also retrieve the notion of *intensity* of connotation, which was not available from the Integer Linear Programing methods. In addition, the Loopy-BP algorithm runs much faster and it is considerably easier to implement than constraint optimization algorithms.

The final outcome in this section is *ConnotationWordNet*, a new lexical resource that has connotation labels over both words and senses following the structure of WordNet.

2.10.1 Graph Representation

The connotation graph for learning *ConnotationWordNet* is shown in Figure 2.5. It is a heterogeneous graph with multiple types of nodes and edges. In particular, it contains two types of nodes: (i) lemma (i.e., words) node, (ii) synset (i.e., senses) nodes; and four types of edges:[i] predicate-argument edges; (ii) argument-argument edges; (iii) argument-synset edges; (iv) synset-synset edges.

Similarly, the predicate-argument edges depict the selectional preference of

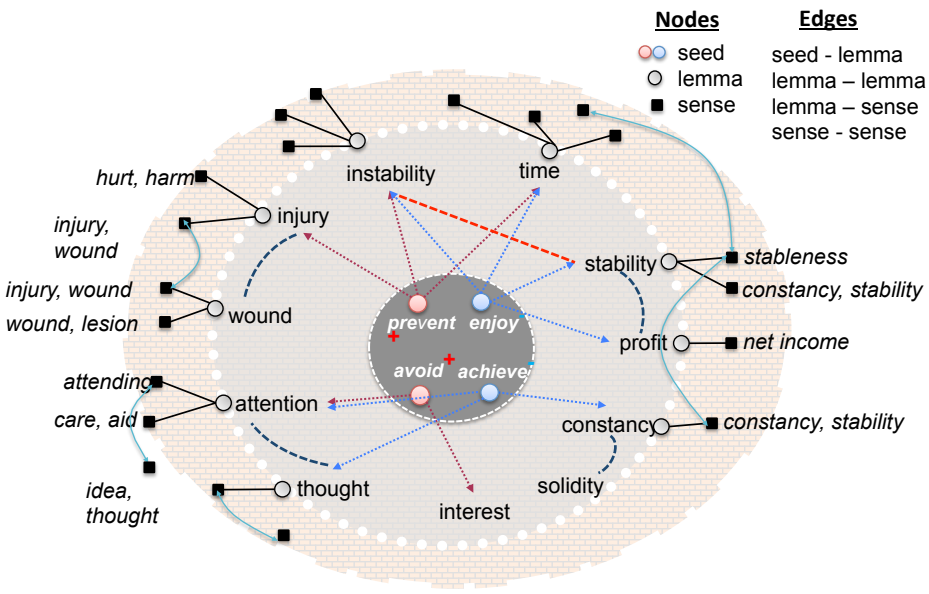


Figure 2.5: Graph Structure for Word + Sense.

connotative predicates over their arguments and encode their co-occurrence relations based on the Google Web 1T corpus. The argument-argument edges are based on the distributional similarities among the arguments. The argument-synset edges capture the synonymy between argument nodes through the corresponding synsets. Finally, the synset-synset edges depict the antonym relations between synset pairs. Hence, comparing to the previous graph representations in Figure 2.4, we introduce the "sense" node and the corresponding edges as shown in Table 2.16.

2.10.2 Task Overview

Our overall goal is to infer the connotative sentiment associated with words as well as synsets. In the previous sections we described how these terms are associated

Connotation Graph	Count
Nodes	179,329
lemmas (words)	115,861
synsets	63,468
Edges	452,008
predicate-argument	178,950
argument-argument	143,856
argument-synset (synonyms)	125,765
synset-synset (antonyms)	3,437

Table 2.16: Various types of nodes and edges, and counts, in the heterogeneous connotation graph.

with each other, and how we constructed the connotation graph by exploiting the relations in Section 2.4.

Our key approach is to treat the connotation sentiment mining task as a network-based classification task. As such, we tackle the problem by formulating and optimizing a graph-based classification objective. This problem setting where the data objects, the class labels of which to be inferred, form a graph is well-known as collective classification (Sen et al. (2008)). We describe the details of our formulation next.

Problem Formulation

Please refer to Table 2.17 for the notations used in the section.

Objective formulation We next define the objective function we seek to optimize for the graph-based connotation sentiment classification task. We propose to use an objective formulation that utilizes pairwise Markov Random Fields (MRFs) (Kindermann and Snell (1980)), which is adapted to our problem setting.

SYMBOL	DEFINITION
V	Sets of nodes in a graph
E	Sets of edges in a graph
G	graph $G = (V, E)$
t	type of edges
	$t \in \{pred - arg, arg - arg, syn - arg, syn - syn\}$
e	edges, $e(v_i, v_j, t) \in E$
$\mathcal{N}$	a neighborhood function
	$\mathcal{N}_v = \{u \mid e(u, v) \in E\} \subseteq V$
$\mathcal{L}$	labels, $\mathcal{L} = \{+, -\}$
$\mathcal{Y}$	nodes to be labeled
y_i	refer to Y_i's label
Ψ	a set of clique potentials
	Prior of node i being in state s
	Edge potential when nodes i and j being in states s 0 and s, respectively
	Belief of node i being in state s

Table 2.17: Symbols

MRFs are a class of probabilistic graphical models that are suited for solving inference problems in networked data. An MRF consists of an undirected graph where each node can be in any of a finite number of states (i.e., class labels). The state of a node is assumed to be dependent on each of its neighbors and independent of other nodes in the graph. This assumption yields a pairwise Markov Random Field (MRF) which is a special case of general MRFs, see Yedidia et al. (2003) for details. In pairwise MRFs, the joint probability of the graph can be written as a product of pairwise factors, parametrized over the edges. These factors are referred as clique potentials in general MRFs, which are essentially functions that collectively determine the graph's joint probability.

Specifically, let $G = (V, E)$ denote a network of random variables, where V consists of the unobserved variables $\mathcal{Y}$ that need to be assigned values from label

set $\mathcal{L}$. Let Ψ denote a set of clique potentials that consists of two types of factors:

- For each $Y_i \in \mathcal{Y}$, $\psi_i \in \Psi$ is a *prior* mapping $\psi_i : \mathcal{L} \rightarrow \mathbb{R}_{\geq 0}$, where $\mathbb{R}_{\geq 0}$ denotes non-negative real numbers.

- For each $e(Y_i, Y_j, t) \in E$, $\psi_{ij}^t \in \Psi$ is a *compatibility* mapping $\psi_{ij}^t : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}_{\geq 0}$.

Given an assignment $\mathbf{y}$ to all the unobserved variables $\mathcal{Y}$ and $\mathbf{x}$ to observed ones $\mathcal{X}$ (variables with known values, if any), our objective function is associated with the following joint probability distribution.

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{Y_i \in \mathcal{Y}} \psi_i(y_i) \prod_{e(Y_i, Y_j, t) \in E} \psi_{ij}^t(y_i, y_j) \tag{2.21}$$

where $Z(\mathbf{x})$ is the normalization function. Our goal is then to infer the maximum likelihood assignment of states (i.e., labels) to unobserved variables (i.e., nodes) that will maximize our objective function above.

Task Definition Having introduced our graph-based classification task and objective formulation, we define our problem more formally.
Given

- a connotation graph $G = (V, E)$ of words and synsets connected with *typed* edges,

- *prior* knowledge (i.e., probabilities) of (some or all) graph objects belonging to each class,

- *compatibility* of two objects with a given pair of labels being connected to each other;

Classify the probability of the graph objects $Y_i \in \mathcal{Y}$ into one of two classes; $\mathcal{L} = \{+, -\}$, such that the class assignments y_i maximize our objective in Equation (2.21). Note that the above formulation enables us to also *rank* the network objects by the magnitude, or probability, of their connotation sentiment.

2.10.3 Induction using Belief Propagation

We formulate the problem of finding the best assignments to unobserved variables in our objective function as an inference problem. The brute force approach through enumeration of all possible assignments is exponential and thus intractable. In general, exact inference is known to be NP-hard and there is no known algorithm which can be theoretically shown to solve the inference problem for general MRFs. Therefore in this work, we employ a computationally tractable (in fact linearly scalable with network size) approximate inference algorithm called Loopy Belief Propagation (LBP) (Yedidia et al. (2003)), which we extend to handle typed graphs like our connotation graph.

Our inference algorithm is based on iterative message passing and the core of it can be concisely expressed as the following two equations:

$$m_{i \to j}(y_j) = \alpha \sum_{y_i \in \mathcal{L}} \Bigg(\psi_{ij}^t(y_i, y_j)\ \psi_i(y_i) \prod_{Y_k \in \mathcal{N}_i \cap \mathcal{Y} \setminus Y_j} m_{k \to i}(y_i) \Bigg), \quad \forall y_j \in \mathcal{L} \tag{2.22}$$

$$b_i(y_i) = \beta\ \psi_i(y_i) \prod_{Y_j \in \mathcal{N}_i \cap \mathcal{Y}} m_{j \to i}(y_i), \forall y_i \in \mathcal{L} \tag{2.23}$$

A message $m_{i \to j}$ is sent from node i to node j and captures the belief of i about j, which is the probability distribution over the labels of j; i.e. what i "thinks" j's label is, given the current label of i and the *type* of the edge that connects i and j. Beliefs refer to marginal probability distributions of nodes over labels; for example $b_i(y_i)$ denotes the *belief* of node i having label y_i. α and β are the normalization constants, which respectively ensure that each message and each set of marginal probabilities sum to 1. At every iteration, each node computes its belief based on messages received from its neighbors, and uses the compatibility mapping to transform its belief into messages for its neighbors. The key idea is that after enough iterations of message passes between the nodes, the "conversations" are likely to come to a consensus, which determines the marginal probabilities of all the unknown variables.

The pseudo-code of our method is given in Algorithm 2. It first initializes all messages to 1 and *prior*s to unbiased (i.e., equal) probabilities for all nodes except the predicate nodes for which the sentiment is known (lines 3-9). It then proceeds by making each $Y_i \in \mathcal{Y}$ communicate messages with their neighbors in an iterative fashion until the messages stabilize (lines 10-14), i.e. convergence is reached. Although convergence is not theoretically guaranteed, in practice LBP converges to beliefs within a small threshold of change (e.g., $\epsilon = 10^{-6}$) fairly quickly with accurate results (Pandit et al. (2007)). At convergence, we calculate the marginal probabilities, that is of assigning Y_i with label y_i, by computing the final beliefs $b_i(y_i)$ (lines 15-17). We use these maximum likelihood label probabilities for classification; for each node i, we assign the label $\mathcal{L}_i \leftarrow \max_{y_i} b_i(y_i)$.

To completely define our algorithm, we need to instantiate the potentials Ψ, in particular the priors and the compatibilities, which we discuss next.

Algorithm 2: CONNOTATION INFERENCE

1 **Input:** Connotation graph G=(V, E), prior potentials ψ_p for predicate words $p \in P$, compatibility potentials ψ_{ij}^t
2 **Output:** Connotation label probabilities for each node $i \in V \backslash P$
3 **foreach** $e(Y_i, Y_j, t) \in E$ **do** // *initialize msg.s*
4 **foreach** $y_j \in \mathcal{L}$ **do**
5 $m_{i \to j}(y_j) \leftarrow 1$
6 **foreach** $i \in V$ **do** // *initialize priors*
7 **foreach** $y_j \in \mathcal{L}$ **do**
8 **if** $i \in P$ **then** $\phi_i(y_j) \leftarrow \psi_i(y_j)$
9 **else** $\phi_i(y_j) \leftarrow 1/|\mathcal{L}|$
10 **repeat** // *iterative message passing*
11 **foreach** $e(Y_i, Y_j, t) \in E$, $Y_j \in \mathcal{Y}^{V \backslash P}$ **do**
12 **foreach** $y_j \in \mathcal{L}$ **do**
13 Use Equation (2.22)
14 **until** all messages stop changing
15 **foreach** $Y_i \in \mathcal{Y}^{V \backslash P}$ **do** // *compute final beliefs*
16 **foreach** $y_i \in \mathcal{L}$ **do**
17 Use Equation (2.23)

Priors The *prior* beliefs ψ_i of nodes can be suitably initialized if there is any prior knowledge for their connotation sentiment (e.g., `enjoy` is positive, `suffer` is negative). As such, our method is flexible to integrate available side information. In case there is no prior knowledge available, each node is initialized equally likely to have any of the possible labels, i.e., $\frac{1}{|\mathcal{L}|}$ as in Algorithm 2 (line 9).

Compatibilities The *compatibility* potentials can be thought of as matrices, with entries $\psi_{ij}^t(y_i, y_j)$ that give the likelihood of a node having label y_i, given that it has a neighbor with label y_j to which it is connected through a type t edge. A key

difference of our method from earlier models is that we use clique potentials that are based on different edge types, as the connotation graph is heterogeneous. This is exactly because the compatibility of class labels of two adjacent nodes depends on the type of the edge connecting them: e.g., $+ \xrightarrow{syn\text{-}arg} +$ is highly compatible, whereas $+ \xrightarrow{syn\text{-}syn} +$ is unlikely; since $syn\text{-}arg$ edges capture synonymy while $syn\text{-}syn$ edges depict antonym.

A sample instantiation of the compatibilities is shown in Table 2.18. Entry $\psi_{ij}^{t}(y_i, y_j)$ is the compatibility of a node with label y_j having a neighbor labeled y_i, given the edge between i and j is type t, for small ϵ. Notice that the potentials for $pred\text{-}arg$, $arg\text{-}arg$, and $syn\text{-}arg$ capture homophily, i.e., nodes with the same label are likely to connect to each other through these types of edges. $arg\text{-}arg$ edges are based on co-occurrence (see Section 2.10.1), which does not carry as strong of an indication of same connotation as e.g., synonymy. Thus, we enforce less homophily for nodes connected through edges of $arg\text{-}arg$ type.

On the other hand, $syn\text{-}syn$ edges connect nodes that are antonyms of each other, and thus the compatibilities capture the reverse relationship among their labels.

Complexity analysis The computational complexity of our proposed connotation sentiment inference algorithm is is linear in the graph size, therefore it is scalable to large datasets. In particular, the most demanding component of the algorithm is the iterative message passing over the edges (lines 10-14 in Algorithm 2), that has time complexity $O(ml^2r)$, where $m = |E|$ is the number of edges in the connotation graph, $l = |\mathcal{L}|$ is the number of classes, and r is the number of iterations until convergence. Often, l is quite small (in our case, $l = 2$) and

t: t_1	**A**	
P	$+$	$-$
$+$	1-ϵ	ϵ
$-$	ϵ	1-ϵ

(t_1) $pred\text{-}arg$

t: t_2	**A**	
A	$+$	$-$
$+$	1-4ϵ	4ϵ
$-$	4ϵ	1-4ϵ

(t_2) $arg\text{-}arg$

t: t_3	**A**	
S	$+$	$-$
$+$	1-ϵ	ϵ
$-$	ϵ	1-ϵ

(t_3) $syn\text{-}arg$

t: t_4	**S**	
S	$+$	$-$
$+$	ϵ	1-ϵ
$-$	1-ϵ	ϵ

(t_4) $syn\text{-}syn$

Table 2.18: Instantiation of compatibility potentials.

$r \ll m$; thus the running time grows linearly with the number of edges.

2.10.4 Evaluations

In this section, we present comprehensive evaluations, including intrinsic evaluation based on labeled corpus (conventional sentiment lexicon and human labels), along with extrinsic evaluation with sentiment classification tasks.

Evaluation I: Comparison against Sentiment Lexicon

ConnotationWordNet is expected to be the superset of a sentiment lexicon, as it is highly likely for any word with positive/negative sentiment to carry connotation of the same polarity. Thus, we also compare *ConnotationWordNet* with conventional sentiment lexicons as described in Section 2.5, as surrogates to measure the performance of our inference algorithm.

In addition to the comparison with sentiment lexicons, we also specifically address the following questions,

- How does the construction of the graph structure affect the performance?
- How sensitive is the performance to the choice of parameter ϵ?

Variants of Graph Construction The construction of the connotation graph, denoted by $G^{\text{WORD+SENSE}}$, which includes words and synsets, has been described in Section 2.10.1. In addition to this graph, we tried several other graph constructions, the first three of which have previously been used in Feng et al. (2013). We briefly describe these graphs below, and compare performance on all the graphs in the proceeding.

G^{WORD} W/ PRED-ARG: This is a (bipartite) subgraph of $G^{\text{WORD+SENSE}}$, which only includes the connotative predicates and their arguments. As such, it contains only type t_1 edges.The edges between the predicates and the arguments can be weighted by their Point-wise Mutual Information (PMI) based on the Google Web 1T corpus.

G^{WORD} W/ OVERLAY: The second graph is also a proper subgraph of $G^{\text{WORD+SENSE}}$, which includes the predicates and all the argument words. Predicate words are connected to their arguments as before. In addition, argument pairs (a_1, a_2) are connected if they occurred together in the "a_1 and a_2" or "a_2 and a_1" coordination Hatzivassiloglou and McKeown (1997); Pickering and Branigan (1998). This graph contains both type t_1 and t_2 edges. The edges can also be weighted based on the distributional similarities of the word pairs.

G^{WORD}: The third graph is a super-graph of G^{WORD} W/ OVERLAY, with additional edges, where argument pairs in synonym and antonym relation are con-

nected to each other. Note that unlike the connotation graph $G^{WORD+SENSE}$, it does *not* contain any synset nodes. Rather, the words that are synonyms or antonyms of each other are directly linked in the graph. As such, this graph contains all edge types t_1 through t_4.

$G^{WORD+SENSE}$ W/ SYNSIM: This is a super-graph of our original $G^{WORD+SENSE}$ graph; that is, it has all the predicate, arguments, and synset nodes, as well as the four types of edges between them. In addition, we add edges of a fifth type t_5 between the synset nodes to capture their similarity. To define similarity, we use the glossary definitions of the synsets and derive three different scores. Each score utilizes the $\texttt{count}(s_1, s_2)$ of overlapping nouns, verbs, and adjectives/adverbs among the glosses of the two synsets s_1 and s_2.

$G^{WORD+SENSE}$ W/ SYNSIM1: We discard edges with `count` less than 3. The weighted version has the `counts` normalized between 0 and 1.

$G^{WORD+SENSE}$ W/ SYNSIM2: We normalize the `counts` by the length of the gloss (the avg of two lengths), that is, $p = \texttt{count / avg(len\_gloss}(s_1)\texttt{,}$ $\texttt{len\_gloss}(s_2)\texttt{)}$ and discard edges with $p < 0.5$. The weighted version contains p values as edge weights.

$G^{WORD+SENSE}$ W/ SYNSIM3: To further sparsify the graph we discard edges with $p < 0.6$. To weigh the edges, we use the cosine similarity between the gloss vectors of the synsets based on the TF-IDF values of the words the glosses contain.

Note that the connotation inference algorithm, as given in Algorithm 2, remains exactly the same for all the graphs described above. The only difference is the set of parameters used; while G^{WORD} W/ PRED-ARG and G^{WORD} W/ OVERLAY contain one and two edge types, respectively and only use compatibilities

(t_1) and (t_2), G^{WORD} uses all four as given in Table 2.18. The $\text{G}^{\text{WORD+SENSE}}$ w/ SYNSIM graphs use an additional compatibility matrix for the synset similarity edges of type t_5, which is the same as the one used for t_1, i.e., similar synsets are likely to have the same connotation label. This flexibility is one of the key advantages of our algorithm as new types of nodes and edges can be added to the graph seamlessly.

Agreement with Sentiment Lexicons To compute the agreement between *ConnotationWordNet* sentiment lexicons, we first compare the performance of our connotation graph $\text{G}^{\text{WORD+SENSE}}$ to graphs that do not include synset nodes but only words. Then we analyze the performance when the additional synset similarity edges are added.

As shown in Table 2.19 (top), we first observe that including the synonym and antonym relations in the graph, as with G^{WORD} and $\text{G}^{\text{WORD+SENSE}}$, improve the performance significantly, almost by an order of magnitude, over graphs G^{WORD} w/ PRED-ARG and G^{WORD} w/ OVERLAY that do not contain those relation types. Furthermore, we notice that the performances on the $\text{G}^{\text{WORD+SENSE}}$ graph are better than those on the word-only graphs. This shows that including the synset nodes explicitly in the graph structure is beneficial. What is more, it gives us a means to obtain connotation labels for the synsets themselves, which we use in the evaluations in the next sections. Finally, we note that using the unweighted versions of the graphs provide relatively more robust performance, potentially due to noise in the relative edge weights.

Next we analyze the performance when the new edges between synsets are introduced, as given in Table 2.19 (bottom). We observe that connecting the synset

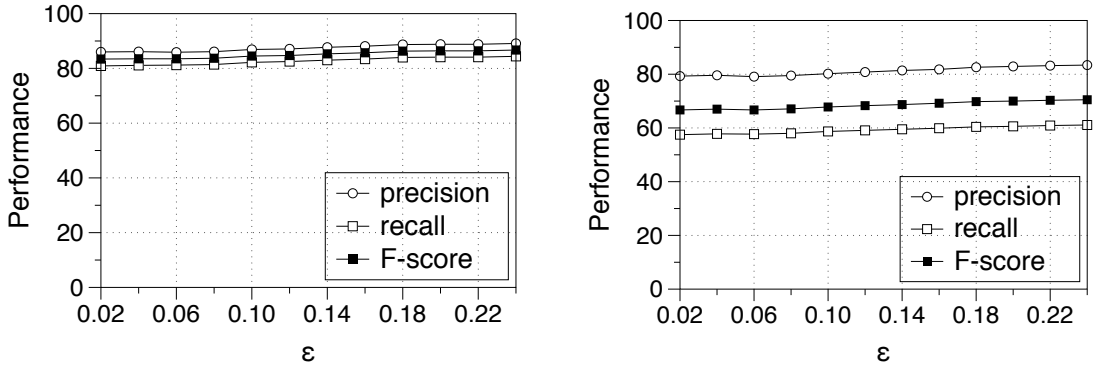
	GENINQ			MPQA
	Precision	**Recall**	**F1-score**	**F**
Variations of G^{WORD}				
W/ PRED-ARG	88.0	67.6	76.5	57.3
W/ PRED-ARG-W	84.9	68.9	76.1	57.8
W/ OVERLAY	87.8	70.4	78.1	58.4
W/ OVERLAY-W	82.2	67.7	74.2	54.2
G^{WORD}	88.5	83.1	85.7	69.7
G^{WORD}-W	75.5	71.5	73.4	53.2
Variations of $G^{WORD+SENSE}$				
$G^{WORD+SENSE}$	**88.8**	**84.1**	**86.4**	**70.0**
$G^{WORD+SENSE}$-W	76.8	73.0	74.9	54.6
W/ SYNSIM1	87.2	83.3	85.2	67.9
W/ SYNSIM2	83.9	80.8	82.3	65.1
W/ SYNSIM3	86.5	83.2	84.8	67.8
W/ SYNSIM1-W	88.0	**84.3**	86.1	69.2
W/ SYNSIM2-W	86.4	83.7	85.0	68.5
W/ SYNSIM3-W	86.7	83.4	85.0	68.2

Table 2.19: Connotation inference performance on various graphs. '-W' indicates weighted versions.

nodes by their gloss-similarity (at least in the ways we tried) does not yield better performance than on our original $G^{WORD+SENSE}$ graph. Different from earlier, the weighted versions of the similarity based graphs provide better performance than their unweighted counterparts. This suggests that glossary similarity would be a more robust means to correlate nodes; we leave it as future work to explore this direction for predicate-argument and argument-argument relations.

Parameter Sensitivity Our belief propagation based connotation sentiment inference algorithm has one user-specified parameter ϵ (see Table 2.18). To study the sensitivity of its performance to the choice of ϵ, we reran our experiments for

$\epsilon = \{0.02, 0.04, \ldots, 0.24\}$. Note that for $\epsilon > 0.25$, compatibilities of ψ^{t_2} in Table 2.18 are reversed, hence the maximum of 0.24. and report the accuracy results on our $\text{G}^{\text{WORD+SENSE}}$ in Figure 2.6 for the two lexicons. The results indicate that the performances remain quite stable across a wide range of the parameter choice.



(a) GENINQ EVAL (b) MPQA EVAL

Figure 2.6: Performance is stable across various ϵ.

Evaluation II: Extrinsic Evaluation via Sentiment Analysis

To show the utility of the resulting lexicon in the context of a concrete sentiment analysis task, we perform lexicon-based sentiment analysis. We experiment with SemEval dataset Strapparava and Mihalcea (2007b) that includes the human labeled dataset for predicting whether a news headline is a *good news* or a *bad news*, which we expect to have a correlation with the use of *connotative* words that we focus on in this paper. For comparison, we also test the connotation lexicon from Feng et al. (2013) and the combined sentiment lexicon GENINQ+MPQA.

Note that there is a difference in how humans judge the orientation and the degree of connotation for a given word out of context, and how the use of such

words in context can be perceived as *good/bad* news. In particular, we conjecture that humans may have a bias toward the use of positive words, which in turn requires calibration from the readers' minds Pennebaker and Stone (2003). That is, we might need to tone down the level of positiveness in order to correctly measure the actual intended positiveness of the message.

With this in mind, we tune the appropriate calibration from a small training data, by using 1 fold from N fold cross validation, and using the remaining $N-1$ folds as testing. We simply learn the mixture coefficient λ to scale the contribution of positive and negative connotation values. We tune this parameter λ. What is reported is based on $\lambda \in \{20, 40, 60, 80\}$. More detailed parameter search does not change the results much. for other lexicons we compare against as well. Note that due to this parameter learning, we are able to report better performance for the CONNOTATION (CO) of Feng et al. (2013) than what the authors have reported in their paper (labeled with *) in Table 2.20.

Table 2.20 shows the results for N=15, where the new lexicon consistently outperforms other competitive lexicons. In addition, Figure 2.7 shows that the performance does not change much based on the size of training data used for parameter tuning (N=$\{5, 10, 15, 20\}$).

Evaluation III: Intrinsic Evaluation via Human Judgment

In this section, we present the result of human evaluation we executed using Amazon Mechanical Turk (AMT). We collect two separate sets of labels: a set of labels at the word-level, and another set at the sense-level. We first describe the labeling process of sense-level connotation: We selected 350 polysemous words and one of their senses, and each Turker was asked to rate the connotative polarity of a

Lexicon	SemEval Threshold			
	20	40	60	80
Instance Size	955	649	341	86
CONNOTATION (CO)	71.5	77.1	81.6	90.5
GENINQ+MPQA	72.8	77.2	80.4	86.7
$G^{\text{WORD+SENSE}}$(95%)	74.5	**79.4**	86.5	**91.9**
$G^{\text{WORD+SENSE}}$(99%)	**74.6**	**79.4**	**86.8**	**91.9**
E-$G^{\text{WORD+SENSE}}$(95%)	72.5	76.8	82.3	87.2
E-$G^{\text{WORD+SENSE}}$(99%)	72.6	76.9	82.5	87.2
CONNOTATION (CO)	70.8	74.6	80.8	93.5
GENINQ+MPQA*	64.5	69.0	74.0	80.5

Table 2.20: SemEval evaluation results, for N=15

given word (or of a given sense), from -5 to 5, 0 being the neutral.[7] For each word, we asked 5 Turkers to rate and we took the average of the 5 ratings as the connotative intensity score of the word. We labeled a word as *negative* if its intensity score is less than 0 and *positive* otherwise. For word-level labels we apply similar procedure as above.

Word-Level Evaluation We first evaluate the word-level assignment of connotation, as shown in Table 2.21. The agreement between the new lexicon and human judges varies between 84% and 86.98%. Sentiment lexicons such as SentiWordNet (Baccianella et al. (2010)) and MPQA (Wiebe et al. (2005)) show low agreement rate with human, which is somewhat as expected: human judges in this study are labeling for subtle connotation, not for more explicit sentiment.

[7]Because senses in WordNet can be tricky to understand, care should be taken in designing the task so that the Turkers will focus only on the corresponding sense of a word. Therefore, we provided the part of speech tag, the WordNet gloss of the selected sense, and a few examples as given in WordNet. As an incentive, each Turker was rewarded $0.07 per hit which consists of 10 words to label.

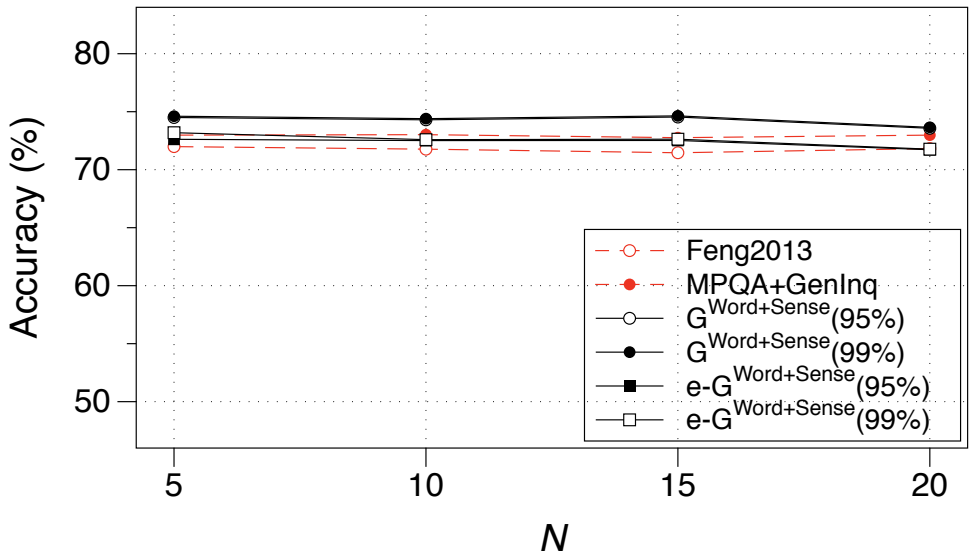


Figure 2.7: Trend of SemEval performance over N, the number of CV folds

MPQA's low agreement rate was mainly due to the low hit rate of the words (successful look-up rate, 33.43%). CONNOTATION (CO) is the lexicon presented in Feng et al. (2013) and it showed a relatively higher 72.13% hit rate.

Note that belief propagation was run until 95% and 99% of the nodes were converged in their beliefs. In addition, the seed words with known connotation labels originally consist of 20 positive and 20 negative predicates. We also extended the seed set with the sentiment lexicon words and denote these runs with E- for 'EXtended'.

Sense-Level Evaluation We also examined the agreement rates on the sense-level. Since MPQA and CONNOTATION (CO) do not provide the polarity scores at the sense-level, we excluded them from this evaluation. Because sense-level polarity assignment is a harder (more subtle) task, the performance of all lexicons

Lexicon	Word-level	Sense-level
SentiWordNet	27.22	14.29
MPQA	31.95	-
CONNOTATION (CO)	62.72	-
$G^{\text{WORD+SENSE}}$(95%)	84.91	83.43
$G^{\text{WORD+SENSE}}$(99%)	84.91	83.71
E-$G^{\text{WORD+SENSE}}$(95%)	**86.98**	**86.29**
E-$G^{\text{WORD+SENSE}}$(99%)	86.69	85.71

Table 2.21: Word-/Sense-level evaluation results

decreased to some degree in comparison to that of word-level evaluations.

A notable goodness of our induction algorithm is that the outcome of the algorithm can be interpreted as an *intensity* of the corresponding connotation. But are these values meaningful? We answer this question in this section. We formulate a pair-wise ranking task as a binary decision task as follows: given a pair of words, we ask which one is more positive (or more negative) than the other. Since we collect human labels based on *scales*, we already have this information at hand. Because different human judges have different notion of scales however, subtle differences are more likely to be noisy. Therefore, we experiment with varying degrees of differences in their scales, as shown in Figure 2.8. Threshold values (ranging from 0.5 to 3.0) indicate the minimum differences in scales for any pair of words, for the pair to be included in the test set. As expected, we observe that the performance improves as we increase the threshold (as pairs get better separated). Within range [0.5, 1.5] (249 pairs examined), the accuracies are as high as 68.27%, which shows that even the subtle differences of the connotative intensities are relatively well reflected in the new lexicons.

The results for pair-wise intensity evaluation (threshold=2.0, 1,208 pairs) are

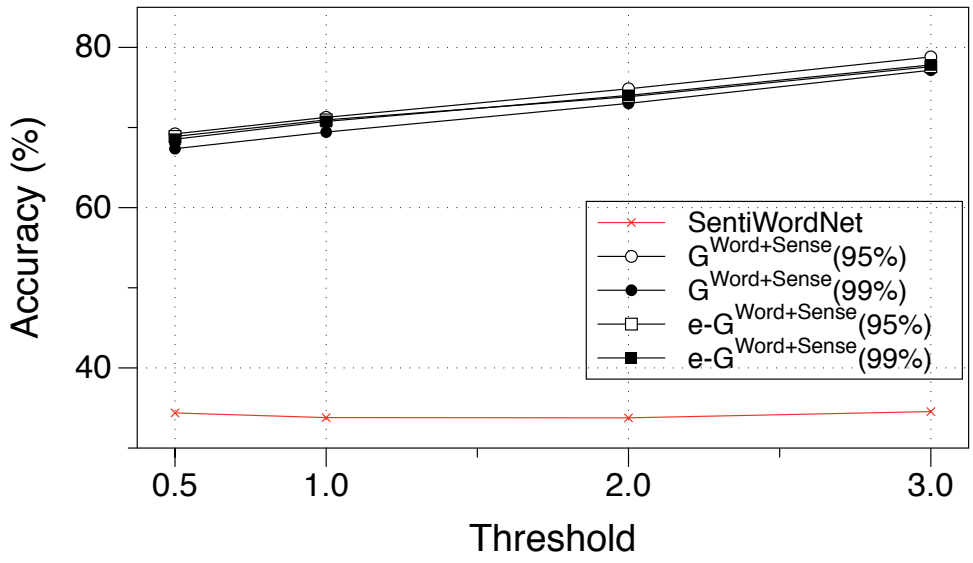


Figure 2.8: Trend of accuracy for pair-wise intensity evaluation over threshold

given in Table 2.22. Despite that intensity is generally a harder property to measure (than the coarser binary categorization of polarities), our connotation lexicons perform surprisingly well, reaching up to 74.83% accuracy. Further study on the incorrect cases reveals that SentiWordNet has many pair of words with the same polarity score (23.34%). Such cases seems to be due to the limited score patterns of SentiWordNet. The ratio of such cases are accounted as *Undecided* in Table 2.22.

2.10.5 Conclusion

We have introduced a novel formulation of lexicon induction operating over both words and senses, by exploiting the innate structure between the words and senses as encoded in WordNet. In addition, we introduce the use of loopy belief prop-

Lexicon	Correct	Undecided
SentiWordNet	33.77	23.34
$G^{\text{WORD+SENSE}}(95\%)$	74.83	0.58
$G^{\text{WORD+SENSE}}(99\%)$	73.01	0.58
E-$G^{\text{WORD+SENSE}}(95\%)$	73.84	1.16
E-$G^{\text{WORD+SENSE}}(99\%)$	74.01	1.16

Table 2.22: Results of pair-wise intensity evaluation, for intensity difference threshold = 2.0

agation over *pairwise*-Markov Random Fields as an effective lexicon induction algorithm. A notable strength of our approach is its expressiveness: various types of prior knowledge and lexical relations can be encoded as node potentials and edge potentials. In addition, it leads to a lexicon of better quality while also offering faster run-time and easiness of implementation. The resulting lexicon, called *ConnotationWordNet*, is the first lexicon that has polarity labels over both words and senses.

2.11 Related Work

2.11.1 Connotation v.s. Sentiment

The study of connotation is broadly related to the study of sentiment and emotion. In the work of Osgood et al. (1957), it has been discussed that connotative meaning of words can be measured in multiple scales of semantic differential, for example, the degree of "goodness" and "badness". Our work presents statistical approaches that measure one such semantic differential automatically. Our graph construction to capture word-to-word relation is analogous to that of Collins-Thompson and Callan (2007), where the graph representation was used to model

more general definitions of words. There is rich research literature on identifying and extracting subjective information in source materials (e.g., Pang and Lee (2008); Liu and Zhang (2012)). In particular, there has been a growing research interest in investigating more fine-grained aspects of lexical sentiment beyond positive and negative sentiment. For example, Mohammad and Turney (2010) study the affects words can *evoke* in people's minds, while Bollen et al. (2011) study various moods, e.g., "tension", "depression", beyond simple dichotomy of positive and negative sentiment. Our work Feng et al. (2011), Feng et al. (2013) and Kang et al. (2014) share this spirit by targeting more subtle, nuanced sentiment even from those words that would be considered as objective in early studies of sentiment analysis.

2.11.2 Sentiment Lexicon

(Wiebe et al. (2005)) first introduced the sentiment lexicon, spawning a great deal of research thereafter. At the beginning, sentiment lexicons were designed to include only those words that *express* sentiment, that is, *subjective* words. However in recent years, sentiment lexicons started expanding to include some of those words that simply associate with sentiment, even if those words are purely objective (e.g., Velikovich et al. (2010); Baccianella et al. (2010)). This trend applies even to the most recent version of the lexicon of Wiebe et al. (2005).

There exist many previous works that deal with the creation of sentiment lexicons, among which we focus on a few popular lexicons. General Inquirer (Stone and Hunt (1963)) can be considered, among other things, the first sentiment lexicon. It is a hand-made lexicon constituted by lemmas. Lemmas are semantic units that can appear in multiple lexicalized forms, e.g. the verb approve is a lemma

that can be found in texts with different inflections, like approved or approving. General Inquirer includes a great amount of information (syntactic, semantic and pragmatic) related to each lemma. Among all this information, there are 4206 lemmas which are tagged as positive or negative. In spite of its age, General Inquirer is still widely used in many works on Sentiment Analysis. MPQA Subjectivity Lexicon (Wiebe et al. (2005)) is an example of a piece of work based on General Inquirer. In particular, it is a lexicon which comprises, in addition to the positive and negative words from General Inquirer, a set of automatically compiled subjective words (Riloff and Wiebe (2003)) and also other terms obtained from a dictionary and a thesaurus. The size of MPQA is significantly greater than General Inquirer but it contains both lemmas and inflections.

We conjecture that this trend of broader coverage suggests that such lexicons are practically more useful than sentiment lexicons that include only those words that are strictly subjective. In this work, we make this transition more explicit and intentional, by introducing a novel *connotation lexicon*. Mohammad and Turney (2010) focussed on emotion *evoked* by common words and phrases. The spirit of their work shares some similarity with ours in that it aims to find the emotion *evoked* by words, as opposed to *expressed*. Two main differences are: (1) our work aims to discover even more subtle association of words with sentiment, and (2) we present a nearly unsupervised approach, while Mohammad and Turney (2010) explored the use of Mechanical Turk to build the lexicon based on human judgment.

Sense-level Sentiment There have been recent studies that address word sense disambiguation issues for sentiment analysis. SentiWordNet (Esuli and Sebastiani

(2006)) was the very first lexicon developed for sense-level labels of sentiment polarity. In recent years, Akkaya et al. (2009) report a successful empirical result where WSD helps improving sentiment analysis, while Wiebe and Mihalcea (2006) study the distinction between objectivity and subjectivity in each different sense of a word, and their empirical effects in the context of sentiment analysis. There are also some recently address the sense-level sentiment of multi-lingu (e.g., Banea et al. (2014); Cruz et al. (2014)). Our work shares the high-level spirit of accessing the sense-level polarity, while also deriving the word-level polarity. There have been a number of previous studies that aim to construct a word-level sentiment lexicon (Wiebe et al. (2005); Qiu et al. (2009)) and a sense-level sentiment lexicon (Esuli and Sebastiani (2006)). But none of these approaches considered to induce the polarity labels at both the word-level and sense-level. Although we focus on learning connotative polarity of words and senses in this paper, the same approach would be applicable to constructing a sentiment lexicon as well.

It is worthy to note the difference between word-level and lemma-level lexicons, like General Inquirer, MPQA Subjectivity Lexicon or Bing Lius Opinion Lexicon, and the synset-level lexicons like SentiWordNet. The first ones are formed by terms with semantic ambiguity due to the polysemy of many words. On the contrary, the synset-level lexicons do not have this problem because their basic units uni-vocally represent one meaning. Nevertheless, the use of this kind of lexicons makes it necessary to pre-process the texts with a Word Sense Disambiguation tool, which has a relatively low accuracy nowadays.

2.11.3 Graph Representation

When automatically collecting words with connotation, previous literature mainly refers to two types of resources to learn the relations between words or construct the word graph: dictionary-based and corpora-based. Dictionary-based resource can be a good start point to learn a lexicon, the contents of which can subsequently be associated with a sentiment score. A widely used semantic lexical resource is WordNet. It enables the distinction between different word forms and meanings. The semantic relations expressed in WordNet can be exploited to generate a sentiment lexicon. A typical approach is to start with a seed set of words and their associated sentiment and to subsequently traverse the WordNet relation while propagating the sentiment (e.g., Kim and Hovy (2004); Hu and Liu (2004); Lerman et al. (2009)). Much of previous research investigated the use of dictionary network (e.g., WordNet) for lexicon induction (e.g., Kamps et al. (2004); Takamura et al. (2005); Adreevskaia and Bergler (2006); Esuli and Sebastiani (2006); Su and Markert (2009); Mohammad et al. (2009)), while relatively less research investigated the use of web documents (e.g., Kaji and Kitsuregawa (2007); Velikovich et al. (2010))).

Another widely explored approach is the use of co-occurrence statistics in web documents has the strength of discovering words that are not in the dictionary, it is practically challenging for academic organizations to collect a substantially large amount of data to enable effective learning. Such concern is not an issue in this work, as we present a novel use of Google Web 1T data (Brants and Franz (2006)) for lexicon induction. The Web 1T data is simple n-gram counts, hence we are able to exploit the statistics of web-scale data without having to collect and process such a large amount of data explicitly.

Rather than by either traversing WordNet or replying on lexical similarity, we utilize both type of resources and explore different lexical relations and different type co-locational statistical information to build the network of words. In addition, we introduce the edge between lemma and senses the when building the graph.

2.11.4 Lexicon Induction Techniques

Several techniques have been developed for sentiment lexicon induction. Graph based approaches have been used in many previous research for lexicon induction. The technique named *label propagation* (Zhu and Ghahramani (2002)) has been used by previous literature (e.g., Raghavan et al. (2010); Velikovich et al. (2010); Chen and Skiena (2014), while random walk based approaches, PageRank in particular, have been used by Esuli and Sebastiani (2007). In our work, we explore the use of both HITS (Kleinberg (1999)) and PageRank (Page et al. (1999)) and present systematic comparison of various options for graph representation and encoding of prior knowledge. We are not aware of any previous research that made use of HITS algorithm for connotation or sentiment lexicon induction. Velikovich et al. (2010) use graph propagation algorithms for constructing a web-scale polarity lexicon for sentiment analysis. Although we employ the same graph propagation algorithm, our graph construction is fundamentally different in that we integrate stronger inductive biases into the graph topology and the corresponding edge weights. As shown in our experimental results, we find that judicious construction of graph structure, exploiting multiple complementing linguistic phenomena can enhance both the performance and the efficiency of the algorithm substantially.

Other interesting approaches include one based on min-cut (Dong et al. (2012))

or LDA (Xie and Li (2012)). Our proposed approaches are more suitable for encoding a much diverse set of linguistic phenomena however. But our work use a few seed predicates with selectional preference instead of relying on word similarity. Some recent work explored the use of constraint optimization framework for inducing domain-dependent sentiment lexicon (e.g., Choi and Cardie (2009); Lu et al. (2011)). Our work differs in that we provide comprehensive insights into different formulations of ILP and LP, aiming to learn the much different task of learning the general connotation of words.

Our work also introduces the use of loopy belief propagation over pairwise-MRF as an alternative solution to these tasks. At a high-level, both approaches share the general idea of propagating confidence or belief over the graph connectivity. The key difference, however, is that in our MRF representation, we can explicitly model various types of word-word, sense-sense and word-sense relations as edge potentials. In particular, we can naturally encode relations that encourage the same assignment (e.g., synonym) as well as the opposite assignment (e.g., antonym) of the polarity labels. Note that integration of the latter is not straightforward in the graph propagation framework.

2.12 Conclusions and Future Work

We presented our work on learning the general connotation based on Web-based and dictionary-based linguistic resources. In order to attain broad-scale coverage while maintaining good precision, we guided the induction algorithms with multiple, carefully selected linguistic insights: (1) semantic prosody; (2) semantic parallelism of coordination; (3) distributional similarity; in addition, we exploited

existing lexical resources as additional inductive bias. For the lexicon inference, we explored several distinct types of inference algorithms, including (i) Random Walk, (ii) Label/Graph Propagation, (iii) Constraint Optimization and (iv) Belief Propagation. Via comprehensive evaluations, we obtain promising results and provide empirical insights into the algorithms.

2.12.1 Summary of Results and Contributions

This work proposed a new task - learning the general connotation. We presented the first broad-coverage connotation lexicon that determines the subtle nuanced sentiment of even those words that are objective on the surface. The resulting connotation lexicon also included general connotation of real-world named entities as an interesting by-product. For constructing the network of words, we exploited various linguistic resources and experimented with several distinctive algorithms, and then proposed the ones with good precision, coverage, and efficiency. In particular, to discover the general connotation at both word-level and sense-level, we introduced a novel formulation of lexicon induction operating over both words and senses, by exploiting the innate structure between the words and senses as encoded in WordNet. Through a series of comprehensive evaluations, we verified that the algorithms we proposed were effective in learning the connotation lexicon, we also provided empirical insights into the practical use of the connotation lexicon. In addition to the connotation lexicon constructed automatically, we created a gold standard that consists of human labeled connotative words. We made both connotation lexicons publicly available for research and practical use.

2.12.2 Future Work

This paper lays the ground-work for learning the general connotation, the insights gained from this work prepare us to pursue several directions that are practice-oriented and with algorithmic support.

Expanding the Coverage of Connotation Lexicon

There are certain limitations that refrain us to further broaden the coverage of connotation lexicon, we will consider address some of them in the future work.

Scarcity of Data When we try to extract word relations from the Web resources, the accuracy tends to suffer if we include the edges with relatively small weight. For this, we only applied some simple strategies such as setting empirical thresholds to filter out "thin" edges in this work. Our future work may seek more effective graph-based algorithms to enhance the (weakly) unsupervised approaches with a specific focus on alleviating data sparsity, for instance, considering graph transition in mining the frequent patterns in sparse graph.

Connotative Predicates Another future direction is to probe the feasibility of learning the connotative predicates automatically while maintaining the good quality. In addition, we would also consider to include different type of predicates (e.g., phrases "break away from", "is traumatized by") in addition to verbs, preferably detect the phrases or syntactic patterns systematically.

Multi-Word Expressions We also would like to include the multi-word expressions in the connotation lexicon. Sometimes it is necessary to differentiate the

singleton lexemes and multi-word expressions while modelling the word relations. For instance, “fearless leader” (for Kim Jong-Il), “Inglourious Basterds” (for the title of a popular movie) have the opposite polarity of connotation compared to the individual word in the phrase. Therefore, it will necessarily help improve the quality (accuracy and coverage) of the connotation lexicon by including multi-word expressions.

Value-laden Connotation

The labels of the general connotation concerned in this work are positive, negative or neutral. For future work, we will consider one step beyond the simplified connotation (as positive, negative, neutral) - learning “value-laden” connotation. The idea is based on the fact that many words or phrases may associate with different dimensions in terms of human values (Schwartz et al. (2001)) or social concepts, for instance, the connotation of “Wall Street” may include the facet of “wealth” (positive) as well as the facet of “greedy” (negative) depending. Therefore, the connotation of the term “Wall Street” depends on which value (“personal wealth” or “morality”) to identify the connotation for. Such task can potentially enable deeper and more accurate understanding of opinions, for instance, help recognizing the public perspective toward certain named entities from different aspects.

Directionality of Connotation

In certain cases, knowing the polarity of the connotative word itself may not necessarily helps predicting the fine-grained positiveness or negativeness toward an entity or a topic. For instance, “A is amazed by B”, there is positive connotation or positive opinion directed to “B”, while “ambiguous” connotation towards “A”.

Similarly, as in sentence “A attacked B”, there is negative connotation or negative opinion towards “B” while “B” is associated with negativeness but the polarity of the opinion towards “B” is ambiguous. We are interested in developing a systematic approach for detect the directionality of connotation.

Chapter 3

Identifying the Intent to Deceit in the Writings

3.1 Introduction

Telling lies often requires creating a story about an experience or attitude that does not exist. Often times, when people intend to deceive in their writings, they would try create a convincing story and present it in a style that appears sincere (Friedman and Tucker (1990)).

Here's an example of review posted on a review website (unimportant details are omitted to protect privacy).

> "I have been shopping at XXX (name of a retailer) since I was very young. My dad used to take me when we were young to the original store down the hill. I also remember when everything was made in America. I recently bought gloves for my wife that she loves. More recently I bought the same gloves for myself and I can honestly say, "I am totally disappointed"! I will

be returning the gloves. My gloves ARE NOT WATER PROOF !!!! They are not the same the same gloves!!! Too bad." .

As we can see from the example, there are several cues that reveal the intent of the author, such as the mention of the name of the retailer, telling unrelated story about his/her family, usage of multiple exclamation marks and so forth. As a matter of fact, imagined experiences are qualitatively different from stories based on real experiences (Johnson and Raye (1981); Vrij (2000)). In fact, several studies indeed have shown that the deceiver often exhibits behavior that belies the content of communication, thus providing cues of deception to an observer. These include linguistic dues (e.g., Newman et al. (2003); Hancock et al. (2004)), paralinguistic cues (e.g., Ekman and O'Sullivan (1991); DePaulo et al. (2003)).

The exponential increase of the availability of online reviews and recommendations make detecting deception an interesting topic for both academic and industrial research. In addition, the need for a robust system is integral in keeping the online community honest and clean. Hence, we focus our study on identifying the intent of deceiving in the natural language texts. We consider the task as a text categorization problem, i.e. differentiating the deceptive writings from the truthful ones. In search of intuitive insights for deception in one's writing, we also explore various linguistically motivated features. Previous studies on deception detection using computational linguistic techniques have mainly relied on shallow lexico-syntactic cues (e.g., Hancock et al. (2007); Vrij et al. (2007); Mihalcea and Strapparava (2009); Ott et al. (2011)). In parallel to the shallow lexical patterns, we particularly explore on deep syntactic structures that are lurking in deceptive writing.

In support of the stylometry analysis on deceptive / truthful writings, it is

necessary to obtain a reference point constituted by sufficient examples of each category. The particular task can be rather challenging as it usually requires psychological knowledge in the social context. One way is to have a dataset annotated by human being. However, the size of the data can be rather limited and it is difficult to control the quality of the data as previous study has shown that human is not very good at recognizing deceptive writing (Ott et al. (2011)). In order to address these issues and conduct generalizable study on deception detection, we therefore investigate on how to attain the deceptive text in an unsupervised manner. We target at the review websites such as `tripadvisor.com` and `amazon.com` as it is known that the deceptive reviews are prevalent on those websites (Ott et al. (2012)).

In the following sections, we will engage our study with the following aspects,

- Construct a corpus (as a pseudo gold standard) that consists of truthful and deceptive writings in an automatic manner.
- Explore the predictive power of stylometry based on truthful and deceptive writing with a special focus on deep syntactic features and providing the insights on the correlation between stylometry and the intent to deceit.
- Investigate how the demographic differences between crowdsourcing and online reviewing might affect the validity of the manufactured datasets.

3.2 Related Work

Deception detection has long been of interest across a broad range of contexts and has been studied in a number of fields, including psychology, communication, and

law enforcement. A lot of effort has been put to automate the process to detect deception. In recent year, there has been a burgeoning research in that uncovers various cues and anomalous patterns for detecting deceptive writings in online review communities and attains promising result. In this section, we will discuss the related work and compare them with our study.

3.2.1 Deception Detection of Online Resource

Some previous research focuses on general spam review detection (e.g., Jindal and Liu (2008); Lim et al. (2010); Jindal et al. (2010)). For example, the work by Jindal and Liu (2008) detects spam reviews by supervised learning using a set of features based on duplicate review text, meta info of reviewers, and products. The proposed method is primarily effective in detecting duplicate spam reviews. Lim et al (Lim et al. (2010)) propose to detect spam reviews based on the pattern of rating behaviour. They propose several heuristics based on multiple reviews on a single product or a group of products within certain time span, and rating deviations etc. The approach practically relies on multiple reviews from the same reviewer targeting the same item or item group. There are some other research work particularly targeting spammers who post multiple reviews. Our work also mainly study the deceptiveness of prolific multi-time reviewers. Mukherjee et al. (2011, 2012) target at a special type of spammers who work in groups and write fake reviews for multiple products. Mukherjee et al. (2011) applies frequent pattern mining to identify group spammers. Mukherjee et al. (2012) applies graph model approach based on behavioural/rule features to detect spammers. Such approaches can only capture the spammers who work in groups, which can be rare in certain domains. Wang et al. (2012) proposes a more general graph-based algorithm to

detect spammers. They create a heterogeneous review graph including the relation between reviews, reviewers and products. However, similar to Jindal et al. (2010); Mukherjee et al. (2011, 2012), such approach is only applicable if a review graph is sufficiently connected. In addition to rule-based approaches, some research work Zhou and Zhang (2008); Toma and Hancock (2010); Ott et al. (2011); Lau et al. (2012) investigates the linguistic cues of deceptive reviews. Ott et al. (2011) first constructs a gold standard dataset of hotel reviews. They obtain deceptive reviews via Amazon Mechanic Turk, truthful reviews from `TripAdvisor.com` website. Then they study the simple lexico-syntactic features that differentiate truthful and deceptive reviews. Feng et al. (2012c) further investigates deep syntactic feature in deception detection in several domains. Both work shows that the machine learning classifier achieves high accuracy based on gold-standard data while human judges can only perform slightly better than chance. This provides one more reliable option to evaluation deception detection techniques other than human annotation. The evaluation method of most previous work rely on human annotation or a limit number of case study (e.g., Jindal and Liu (2008); Jindal et al. (2010); Lim et al. (2010); Wang et al. (2012); Mukherjee et al. (2012, 2011); Xie et al. (2012); Zhang et al. (2012)). The work Feng et al. (2012c) introduce a novel way for evaluation based on gold standard dataset. They first propose unsupervised approach to construct domain-specific pseudo-gold standard dataset for detecting deceptive reviews and employ gold-standard data to indirectly evaluate the proposed approach. The evaluation method of our work is mainly inspired by Ott et al. (2011) and Feng et al. (2012c).

3.2.2 Writing Styles

Previous studies in computerized deception detection have relied only on shallow lexico-syntactic cues. Most are based on dictionary-based word counting using LIWC (Pennebaker et al. (2007)) (e.g., Hancock et al. (2007); Vrij et al. (2007)), while some recent ones explored the use of machine learning techniques using simple lexico-syntactic patterns, such as n-grams and part-of-speech (POS) tags (e.g., Mihalcea and Strapparava (2009); Ott et al. (2011)). These previous studies unveil interesting correlations between certain lexical items or categories with deception that may not be readily apparent to human judges. For instance, the work of Ott et al. (2011) in the hotel review domain results in very insightful observations that deceptive reviewers tend to use verbs and personal pronouns (e.g., "I", "my") more often, while truthful reviewers tend to use more of nouns, adjectives, prepositions. In parallel to these shallow lexical patterns, we have a special focus on deep syntactic stylometry for deception detection, adding a somewhat unconventional angle to prior literature. Such more sophisticated linguistic cues have been explored as well: syntactic labels from partial parsing (Hirst and Feiguina (2007)), etc. The use of syntactic features from parse trees in authorship attribution was initiated by Baayen et al. (1996), and more recently, syntactic features from PCFG parse trees have also been used for authorship attribution (Raghavan et al. (2010)), gender attribution (Sarawgi et al. (2011a)), genre identification (Stamatatos et al. (2000)), native language identification (Wong and Dras (2011)) and readability assessment (Pitler and Nenkova (2008)). We report for the first time that statistical patterns in deep syntactic structure based on PCFG parse trees can also help discriminating deceptive writing from truthful ones.

Stylometry has been a successful line of research. Most early work in stylom-

etry focus on tokenized text. Early works without the support of modern parsing techniques exploit the characteristics based on untokenized text and all it's substrings. n-gram - A very common research practice is to use the high dimension feature vectors based on lexical features or shallow syntactic features.

3.3 Data

To collect the data for the task of identifying the deceptive intention in the writing is particular challenging due to nature of the task - there is no verified approach to track human vigilance and perception in terms of deceptive behavior online, and rarely any individuals who actually confess their deceptive intent. In this study, our target is the deceptive writings are prevalence rapidly increasing online communities of practice (Ott et al. (2012)) . Hence, we focus on the web resources and explore different approaches in diverse domains. There are several popular techniques to obtain the corpora of the deception detection task.

3.3.1 Overview

Previous literature on deception detection has adopted a number of convectional techniques for gathering truthful and deceptive statements (e.g., Kraut (1978); Porter and Yuille (1996); Newman et al. (2003)), including approaches for achieving gold-standard and non-gold standard creation approaches in psychology experiments (Gokhman et al. (2012)). However those approaches are not suitable in our case for the following reasons: (1) it is not practically possible to conduct those kind of experiments in a large-scale settings. (2) it is not feasible for the context of detecting insidious deception on the Web because individuals who post

deceptive content online are unlikely to confess. Another approach to obtain the gold-standard is to label the cases by the third-party human annotators for online reviews (e.g., Lim et al. (2010); Wu et al. (2010); Li et al. (2011)). However, the ability to detect deception of human judges are known to be poor (Bond and DePaulo (2006)). Recent study confirmed that the performance of human annotators is often no better than random guess (Ott et al. (2011)). In addition, a truth bias or over-trusting nature was observed in most human judges (Vrij (2000)).

Our goal is to verify the efficacy of the different techniques for capturing the deceptive behavior online and provide some insights based on the statistical analysis. Therefore, we first need to collect a corpora with high quality, for this we refer to crowdsourcing; and then we aim to find a way to collect larger scale data and therefore explore more effective way to collect data in an automatic manner, for this we largely refer to the real data posted on the websites. For this study, we aim to carefully design the experiments in a way that is both ethically acceptable and experimentally sound.

Crowdsourcing Similar to the conventional sanctioned deception approaches in psychology, one way to obtain the gold-standard labels is to simply collect gold standard truthful/deceptive content. Crowdsourcing provides a platform to solicit truthful or deceptive content. Therefore, some previous work (e.g., Mihalcea and Strapparava (2009); Ott et al. (2011)) has explored an alternative approach, writing tasks (so the labels of the writings are given), to gather deceptive and truthful writings. So it possible for researchers solicit small sized gold-standard datasets. In this work, we consider the dataset by crowdsourcing as gold-standard.

Automated Data Acquisition One obvious limitation of handcrafting data even via crowdsourcing for deception is lack of scalability and it also can be difficult or impossible to adapt to different domains, even from domains such as hotels to restaurants (Feng et al. (2012b)). Therefore, to generalize the study of deception detection, we explore unsupervised approaches to acquire deceptive and truthful writings. In particular, we conduct an in-depth study on corpus collection based on the popular review websites where deceptive reviews are known to be prevalent. Deceptive review spams are the most common type of deceptive writings seen on the Web (Caspi and Gorsky (2006)) and it is more practically meaningful to target at this type of potentially insidious deception. More details on the automated data collection will be elaborated in Section 3.4.

3.3.2 Four Datasets

For this study, we explore different types of deceptive writings in diverse domains, spanning from product reviews (descriptive writings) to essays (argument writing).

I. TripAdvisor (Gold standard): is a corpus generated in the work of Ott et al. (2011). it contains 400 positive (5-star), gold standard deceptive hotel reviews, as well as 400 (positive) truthful reviews covering the same set of hotels. The truthful reviews were mined directly from `www.tripadviser.com` based on a few heuristics while deception reviews were gathered via Amazon Mechanical Turk by 400 unique Turkers. The corpus has been used to train a learning-based classifier that could distinguish deceptive vs. truthful positive reviews at 90% accuracy levels, which indicates the quality that approximate the gold-standard.

In this work, we consider this dataset as gold-standard.

II. TripAdvisor (Pseudo-gold standard): This dataset contains 400 truthful and 400 deceptive reviews harvested from `www.tripadviser.com`, based on fake review detection heuristics introduced in Feng et al. (2012c). Specifically, using the notation of Feng et al. (2012c), we use data created by STRATEGY-$dist\Phi$ heuristic, with $H_S, \mathcal{S}$ as deceptive and $H'_S, \mathcal{T}$ as truthful. We will further describe how we constructed this dataset in Section 3.4.

III. Yelp (Pseudo-gold standard): This dataset is created based on the review filter at `www.yelp.com` . We select 400 *filtered* reviews (with $\geq$ 80 words and content related to the restaurant) and 400 *displayed* reviews for 35 Italian restaurants with average ratings in the range of [3.5, 4.0]. Class labels are based on the meta data, which tells us whether each review is *filtered* by Yelp's automated review filtering system or not. We expect that *filtered* reviews roughly correspond to deceptive reviews, and *displayed* reviews to truthful ones, but not without considerable noise. We only collect 5-star reviews to avoid unwanted noise from varying degree of sentiment.

IV. Essays (Gold standard): was introduced by the work of Mihalcea and Strapparava (2009). This corpus includes truthful and deceptive essays which were collected via Amazon Mechanic Turk. The participants were ask about their belief to a given topic and then they were asked to convince others that they hold the opposite belief. The corpus contains the following three topics: "Abortion" (100 essays per class), "Best Friend" (98 essays per class), and "Death Penalty" (98 essays per class).

3.4 Automated Data Acquisition

In this section, we present our work (Feng et al. (2012c)) on how to develop an automatic approach to create a corpus for deception detection. As confirmed by numerous previous studies (e.g., Dellarocas (2006); Yoo and Gretzel (2009); Mukherjee et al. (2011)), there has been a lot of speculation and anecdotal evidence about the prevalence of deceptive product reviews, i.e., fictitious customer reviews that are written to sound authentic in order to promote the business. We then target at online review sites such as TripAdvisor and the generated corpus consists of truthful and deceptive reviews directly collected from review website. Since the labels of the reviews are approximate, this corpus is considered as a pseudo gold standard.

3.4.1 Statistical Analysis

We aim to investigate the deceptive activities specific to the online reviews. In order to understand this domain better, we first compute some statistics based on reviews from a popular hotel review website - TripAdvisor. To collect the data for statistical analysis, we identify about 4000 hotels located in the United States that listed on `TripAdvisor.com` and then crawl the entire set of historical reviews of each hotel, amount to over 800,000 reviews. Since the deceptive deception emerges in recently years, we take the time factor into consideration and examine the reviews of most recent years from 2007 to 2011.

Among the information posted on the website, the average rating scores would directly reflect the affect of deceptive reviews. In addition, we differentiate reviewers as single-time or multi-time reviewers. The single-time reviewers are the

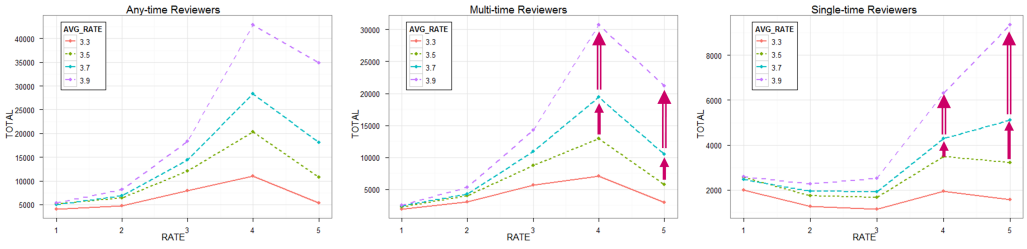


Figure 3.1: Representative distributions of review-ratings for hotels with average rating $\bar{r} \in [3.2, 3.9]$

ones who only post a single review before and multi-time reviewers are the ones, who have posted more than one reviews. Previous research indicates that single-time reviewers are highly suspicious comparing to long-term registered reviewers. Therefore, we calculate the distribution among star rates on a scale from 1 to 5 with 5 being the most positive and 1 being the most negative. We also make sure to calculate the stats based on a set of hotels that are comparable in terms of user ratings. It does not really make sense to compare the rating distribution of hotels with extremely poor ratings and the ones with extremely good ratings. Therefore, we only calculate based on reviews for hotels with the same average rating in range of [3.2, 3.9]. Figure 3.1 shows the representative distributions of the review ratings of hotels with the given average star rating $\bar{r}$ in the range. We see that the review ratings of single time reviewers are relatively more skewed toward extreme opinions: 5-star and 1-star ratings. Similarly as in Figure 1, the distribution of single-time reviewers forms a J-shaped, bi-modal line. However, the distribution of multi-time and any-time reviewers are different, i.e., here we see unimodal graphs with the highest point at rating = 4.

Also notice that if we compare the distribution of reviews written by *single-time* reviewers across different $\bar{r} \in [3.2, 3.9]$, then we see that the number of

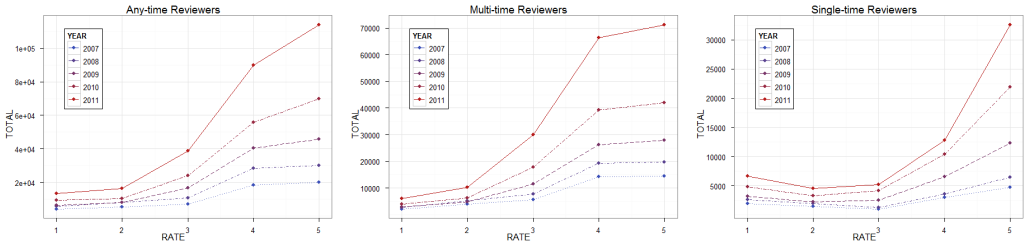


Figure 3.2: Representative distributions of review-ratings for year $y \in [2007, 2011]$

5-star reviews increases faster than the number of 4-star reviews as the average rating goes up, as highlighted by arrows in Figure 3.1. Notice the delta difference in the length of arrows between multi-time and single-time reviewers. In contrast, if we compare the distribution of reviews written by *multi-time* reviewers, then the increase in the number of 4-star and 5-star reviews across different $\bar{r}$ is generally comparable.

This indicates that hotels that are maintaining an average rating as high as 3.9, are substantially supported by an *unnaturally* higher portion of single-time reviewers giving the 5-star reviews, a bulk of which might as well be fakes. Without solid evidence however, such hotels might insist that all those single-time reviewers are genuinely happy customers, who were impressed enough to write a single strongly positive review just for them, just once in their lives. The evaluation presented later in this paper will provide the first quantitative proof to fundamentally challenge such arguments.

We postulate that for a set of hotels of the same average star rating $\bar{r}$, there exists a *natural* distribution of the *truthful* customer ratings. We cannot measure this distribution directly and exactly, because deceptive reviews distort this natural distribution, and it is not possible to identify all of the deceptive reviews. Nonethe-

$\mathcal{S}$	Set of single-time reviewers.
$\mathcal{M}$	Set of multiple-time reviewer.
$\mathcal{T}$	Set of regular reviewers .
$\mathcal{R}^*(h)$	Set of $*$ type reviewers that reviewed h.
$\overline{r}_h$	average rate of hotel h .
$\overline{r}_h^{\mathcal{R}}$	average rate of hotel h based on reviews by $\mathcal{R}$ type of reviewers.
$rv_\lambda^{\mathcal{R}}(h)$	a review with rate λ of hotel h by a reviewer in $\mathcal{R}$.

Table 3.1: Notational Definitions.

less, as will be shown, the notion of the natural distribution helps us identifying the distributional footprints of deceptive reviews.

3.4.2 Deception Detection Strategies

In this section, we introduce several strategies guided by statistics that are suggestive of distributional anomaly. Our detection strategies are *content independent*, in that it will rely only on the meta data, such as, the rating distribution of a hotel, or the historic rating distribution of a reviewer. In other words, we assume that deceptive/genuine reviews can be tracked down by studying the characteristics of corresponding reviewers or reviewees and their historical correlative behavior. Nevertheless, evaluations of the strategies are via classifications based on textual features of reviews, will be show in next secion.

Committee of Truthful Reviewers $\mathcal{T}$ We first begin by collecting the *"committee of truthful reviewers"*, which will become handy in some of the deception detection strategies, as well as evaluation setup. We conjecture that reviewers

with a long history of reviews are more likely to be trustworthy. We collect a set of reviewers who have written more than 10 reviews. One thing regular reviewers hardly do is to post several reviews in a very short time interval (Lim et al. (2010)). We therefore discard any reviewer who has written more than 1 review within 2 consecutive days, as such reviewers might be engaged in deceptive activities. Finally, we only keep those reviewers whose rating trends are not outrageous. For instance, we discard reviewers whose ratings are always far away ($\delta = r(h) - \overline{r}_h, |\delta| \geq 1$) from the the average ratings of *all* the reviewees (i.e., hotels). Such reviewers who are consistently far off from the average might not be necessarily deceptive, but nonetheless do not reflect the general sentiment of the crowd but still reasonably close to the average rating of the reviewees (i.e., hotel).

For TripAdvisor data, we set the constraint as

$$\delta = r(h) - \overline{r}_h, |\delta| \leq 1$$

Note that we purposefully apply somewhat conservative criteria to the committee of truthful reviewers, as our primary goal in this study is not to identify genuine or fake reviewers, but to approximate the actual average rate of a hotel as accurately as possible. The resulting committee has 42766 reviewers as its trustworthy member, which we denote as $\mathcal{T}$. We refer to this strategy of identifying truthful reviewers as

Identifying Deceptive Business Entities Next we present several different strategies for identifying deceptive hotels. Another way, not as sensitive to time factor as the previous way, is to model how close (or deviated) the presented rating is to actual rating in general. If a hotel doesn't hire fake reviewers to increase its rate,

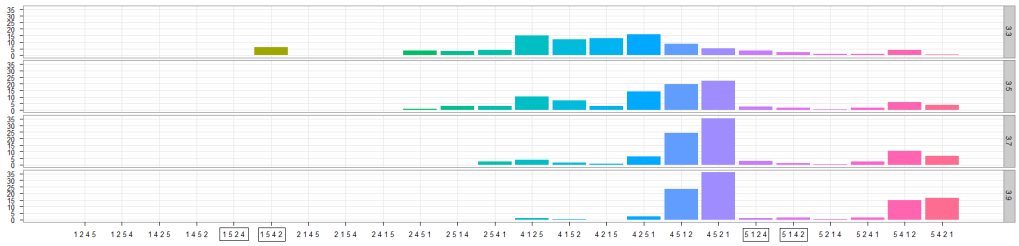


Figure 3.3: *Distribution of distribution* of review-ratings by *any-time* reviewers.

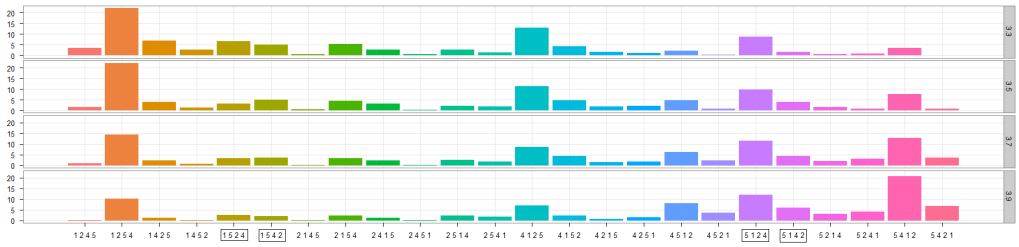


Figure 3.4: *Distribution of distribution* of review-ratings by *single-time* reviewers.

then the presented average rate should be close to its actual quality level. Here we employ two metrics to model the closeness or deviation.

[1] STRATEGY-$avg\Delta$

This strategy is based on the insights we gained from Figure 3.1. For a hotel h, we calculate the discrepancy between the average rating by the committee of truthful reviewers ($\mathcal{T}$) and the average rating by single-time reviewers $\mathcal{S}$:

More formally,

$$\delta_h = \overline{r}_h^{\mathcal{S}} - \overline{r}_h^{\mathcal{T}}$$

After sorting the hotels by δ in a *descending* order, hotels ranked at top are assumed to be more suspicious and hotels ranked at bottom are assumed to be credible as shown in Table 3.3.

[2] STRATEGY-$dist\Phi$

This strategy is based on the insights we gained from Figure 3.3 and 3.4. The row indexes the average rating of the corresponding products, and the column indexes a particular ordering of ratings sorted by corresponding review counts (i.e., each column represents a particular shape of the distribution of review-ratings). The length of each bar is proportionate to the number of products with the corresponding shape of the review distribution. For suspicious hotels, the portion high-rate reviews by single-time reviewers would be likely higher than that by multi-time reviewers. Remind that the percentage of the distribution ($5 \succ 1 \succ 2 \succ 4$) with respect to single-time reviewers in Figure 3.4 is substantially higher than that of any-time reviewers in Figure 3.3. Therefore, we first calculate the ratio of the number of strongly positive reviews to the number of strongly negative reviews among different groups of reviewers, i.e. $\mathcal{S}$ and $\mathcal{M}$.

$$\tau_h^{\mathcal{R}} = \frac{|rv_\lambda^{\mathcal{R}}(h), \lambda \geq \lambda_{high}|}{|rv_\lambda^{\mathcal{R}}(h), \lambda \leq \lambda_{low}|}$$

For suspicious hotels, we pick those with bigger r_h: We set $\lambda_{high} = 5$ and $\lambda_{low} = 2$.

$$r_h = \frac{\tau_h^{\mathcal{S}}}{\tau_h^{\mathcal{M}}}$$

For trustful hotels, we pick those with the smaller r'_h:

$$r'_h = \frac{max(\tau_h^{\mathcal{S}}, \tau_h^{\mathcal{M}})}{min(\tau_h^{\mathcal{S}}, \tau_h^{\mathcal{M}})} - 1$$

[3] STRATEGY-$peak \uparrow$

A sudden burst in the reviewing activity can be a sign for deceptive activities (e.g., Jindal et al. (2010)). We therefore translate this idea into a strategy so that we can compare our proposed strategies with the heuristics in the previous work. Specifically, if $\overline{r}(h, M)$ among reviews posted in month M for h is greater than the average rating among reviews posted within the two months before and after M, then we assume the corresponding hotel is suspicious.

3.4.3 Strategy Evaluation

We want to measure the quality of deception detection strategies introduced earlier, but there is no direct and straightforward method to do so. One might wonder whether we could perform human judgment study on our proposed strategies, but there are two major problems: first, it has been shown in prior literature that human are not good at detecting deceptions (Vrij et al. (2007)), including detecting fake reviews (Ott et al. (2011)). Second, because our strategies are essentially developed based on our own human judgment guided by relevant statistics, human judgment study guided by the same set of statistics is likely to lead to the conclusion that might be overly favorable for this study.

Therefore, we introduce an alternative approach to evaluation that can directly measure the utility of deception detection strategies by employing machine learning framework. More specifically, we exploit the gold standard dataset created by Ott et al. (2011), which includes 400 deceptive reviews that are written by hired people, and contrastive 400 truthful reviews that are gathered from TripAdvisor, modulo filtering rules to reduce incidental inclusion of deceptive reviews. Henceforth, we refer to this dataset as the *gold standard data*, as this is the only dataset

publicly available with true gold standard in the product review domain.

For all our strategies, we mix and match the gold standard data and the pseudo-gold standard data in three different combinations as follows:

(C1) $rule, gold$: Train on the dataset with pseudo gold standard determined by one of the strategies, and test on gold standard dataset of Ott et al. (2011).

(C2) $gold, rule$: Train on gold standard dataset and test on pseudo gold standard dataset.

(C3) $rule, rule$: Train and test on the pseudo gold standard dataset (of different split).

The purpose of the above variations is in order to probe whether a high performance in (C1) and/or (C2) correlate with (C3) empirically. If it does, then it would be suggestive that one could resort to the experiment in the (C3) configuration alone, when the gold standard dataset is not readily available.

Experiment Settings Whenever possible, the dataset with the pseudo-gold standard determined by one of our strategies will include 400 reviews per class, where 80% is used for training, and 20% is used for testing for 5-fold cross validation. Note that for certain variations of strategies, it might be impossible to find as many as 400 reviews for each class. In those cases, the number of training and test instances are given in the parenthesis in Table 3.3 and 3.4. To avoid overlap between the pseudo-gold standard determined by our strategies and the gold standard data, we exclude all those reviews for the 20 hotels that are selected by Ott et al. (2011). We also truncate each review at 150 tokens, to balance the length with the gold standard data. We exclude hotels with less than 20 reviews per year, assuming

deceptive hotels are likely to be much more productive than generating only a handful reviews per year. We use the LIBSVM (Chang and Lin (2011)) classifier and feature values are term frequencies scaled with respect to the document length.

Notational Definitions In Table 2 – 6, the pseudo gold standard dataset is defined using notations of the following format: $(H, \mathcal{R})$, where H corresponds to the set of hotels, and $\mathcal{R}$ corresponds to the set of reviewers. $\mathcal{R}$ can be any of the top three notations in Table 1. H can be one of the following three options:

- H_S denotes the set of hotels selected by strategy S.
- H'_S denotes the set of hotels randomly selected from the complement set of H_S, so that $H_S \cap H'_S = \emptyset$.
- H^* stands for a set of randomly selected hotels.

Baselines Next we define three different pseudo gold standard datasets that correspond to baselines, using notations defined above. These baseline datasets will contrast the quality of other pseudo gold standard dataset created by deception detection strategies discussed earlier.

- BASELINE-1: (DECEPTIVE = $*, *$ TRUTH = $*, *$)

  Both hotels and reviews are randomly selected.

- BASELINE-2: (DECEPTIVE = $H^*, \mathcal{S}$ TRUTH = $H^*, \mathcal{M}$)

  First a set of hotels are randomly selected, then reviews written by $\mathcal{S}$ for the corresponding set of hotels H^* are considered as deceptive reviews, and

DECEPTIVE	TRUTH	TRAIN	TEST	ACCURACY (%)
$*, *$	$*, *$	$rule$	$gold$	43.5
		$gold$	$rule$	42.0
		$rule$	$rule$	48.4
$H^*, \mathcal{S}$	$H^*, \mathcal{T}$	$rule$	$gold$	50.0
		$gold$	$rule$	58.1
		$rule$	$rule$	61.3
$H^*, \mathcal{S}$	$H^*, \mathcal{M}$	$rule$	$gold$	38.5
		$gold$	$rule$	44.0
		$rule$	$rule$	55.0

Table 3.2: Classification on 5-star reviews: BASELINES

DECEPTIVE	TRUTH	TRAIN	TEST	ACCURACY (%)
$H_S, \mathcal{S}$	$H'_S, \mathcal{T}$	$rule$	$gold$	65.7
		$gold$	$rule$	65.1
		$rule$	$rule$	67.1
$H_S, \mathcal{S}$	$H_S, \mathcal{T}$	$rule$	$gold$	**70.0**
		$gold$	$rule$	66.3
		$rule$	$rule$	65.0
$H_S, \mathcal{S}$	$H_S, \mathcal{M}$	$rule$	$gold$	58.3
		$gold$	$rule$	45.6
		$rule$	$rule$	43.1

Table 3.3: Classification on 5-star reviews: STRATEGY-$avg\Delta$

reviews written by $\mathcal{M}$ are considered as truthful reviews. Note that the same set of hotels are used by both deceptive and truthful class.

- BASELINE-3: (DECEPTIVE = $H^*, \mathcal{S}$ TRUTH = $H^*, \mathcal{T}$)

  First randomly select a set hotels, then reviews by $\mathcal{S}$ are considered as deceptive, and reviews by $\mathcal{T}$ are considered as truthful. Again, the same set of hotels are used by both deceptive and truthful class.

DECEPTIVE	TRUTH	TRAIN	TEST	ACCURACY (%)
$H_S, \mathcal{S}$	$H'_S, \mathcal{T}$	$rule$	$gold$	**72.5**
		$gold$	$rule$	**73.8**
		$rule$	$rule$	**74.4**
$H_S, \mathcal{S}$	$H_S, \mathcal{T}$	$rule$	$gold$	60.3 (160/40)
		$gold$	$rule$	62.0
		$rule$	$rule$	63.2 (160/40)
$H_S, \mathcal{S}$	$H_S, \mathcal{M}$	$rule$	$gold$	36.9
		$gold$	$rule$	45.6
		$rule$	$rule$	58.0

Table 3.4: Classification on 5-star reviews: STRATEGY-$dist\Phi$.

DECEPTIVE	TRUTH	TRAIN	TEST	ACCURACY (%)
$H_S, \mathcal{S}$	$H'_S, \mathcal{T}$	$rule$	$gold$	54.1 (200/50)
		$gold$	$rule$	64.4
		$rule$	$rule$	60.4 (200/50)
$H_S, \mathcal{S}$	$H_S, \mathcal{T}$	$rule$	$gold$	53.8 (200/50)
		$gold$	$rule$	**72.0**
		$rule$	$rule$	61.0 (200/50)
$H_S, \mathcal{S}$	$H_S, \mathcal{M}$	$rule$	$gold$	40.2 (200/50)
		$gold$	$rule$	40.5
		$rule$	$rule$	56.6 (200/50)

Table 3.5: Classification on 5-star reviews: STRATEGY-$peak \uparrow$.

3.4.4 Constructing Datasets

Lastly, we select truthful reviews based on the hypothesis (in Section 3.4.2) that the reviews from trusted members are truthful. For deceptive reviews, we first identify the suspiciously business entities at `www.tripadvisor.com` using the strategies proposed in Section 3.4.2 and the select reviews by single-time reviewers into the set of deceptive reviews. For the rest of the study, we employ datasets based best performing approach reported based on the evaluation results.

3.5 Deceptive Writing Style Analysis

Previous work on deception detection based on stylometry analysis has shown that some lexico or shallow syntactic features can be strong indicators for the intent of deceit in the writings. For instance, recently work by Ott et al. (2011) find that deceptive reviews frequently use personal pronouns while truthful reviews contains more proper nouns and number that indicates factoids. Once again confirms that the use of stylometry, authorship recognition through purely linguistic means, has contributed to the deception detection. Some other recent work Brennan et al. (2012); Harris (2012), however, brings it to the attention that the stylistic characteristics might be challenged if the authors attempt to disguise their linguistic writing style based on the statistical information at lexical or shallow syntactic level, for instance, the spammers could deliberately avoid using personal pronouns or include more numbers and proper nouns when preparing the deceptive writings. Therefore, this work further explore deep syntactic features, which is potentially not as easy to counterfeit comparing to lexical and shallow syntactic style that investigated in the previous work.

One focus of our work is to uncover the characteristics of the writing style of an author when he/she intents to deceit so as to provide some insights on the quantitative yet interpretable assessment of the style (statistical implication or evidence). In this study, we first formulate the task for identifying deception as text classification problem. We develop a learning model to differentiate the truthful texts from the deceptive ones based on the stylometric features. It connects the statistical properties of text-classification tasks with the generalization performance of a classifier based on quantitative indicators. Unlike conventional approaches to learning text classifiers, which rely primarily on empirical evidence, we aim to

explain why and when the classifier encoded with certain features perform well for the classification. In particular, it addresses the following questions: How powerful is various stylometric features in differentiating deceptive writings from truthful ones? In parallel to these shallow lexical patterns, might there be deep syntactic structures that are lurking in deceptive writing? How robust are the syntactic cues in the cross topic setting?

3.5.1 Stylometry Analysis

The study of stylometry statistically quantifies the linguistic features, which show as subtle but regular and discernible differences between texts. Some earliest work by Mendenhall (1887), Yule (1938), Yule and Yule (1944) dates back to 19th centry. In the research to date, it belongs to the core task of text categorization like authorship identification (e.g., Raghavan et al. (2010); Feng et al. (2012a)), gender attribution (Sarawgi et al. (2011b)), native language identification (Wong and Dras (2011)) and so forth. It has legal as well as academic and literary applications, ranging from the question of the authorship of novels to forensic linguistics. There is a vivid and growing interest in forensic application for stylometry (De Vel et al. (2001)) given that the development of stylometry is mainly reflected in the choices of quantifiable features used as authorial discriminators. When the domain specific labeled data is insufficient for training a classifier, the statistical evidence could be considered a reference for investigating the individual cases.

There are various types of stylistic features using computational methods such as word n-grams, vocabulary richness, character n-gram (fixed and variable length), content or language specific features. In early phrase of the computational stylometry study, the lexical and shallow syntactic features are predominant (e.g.,

Burrows (2002); Argamon et al. (2003); Abbasi and Chen (2005); Monroe et al. (2008)). Syntax features came to the attention of more and more researchers (e.g., Stamatatos et al. (2001); Zhao and Zobel (2005); Argamon et al. (2007)). Some very recent works have shown that PCFG models can detect distributional difference in sentence structure in gender attribution (Sarawgi et al. (2011b)), authorship attribution (Raghavan et al. (2010)), and native language identification (Wong and Dras (2010)). In particular, we aim to discover the intermediate representation built in a hierarchical structure at phrasal or sentential level, which corresponds to the interpretable factors of higher level of grammar units. To explore the stylometry for this new domain - deception detection, we also set forth a comprehensive study expounding the role of various syntactic features.

3.5.2 Features

Given that more advance language parsing tools are readily available, we are able to explore stylistic characteristics based on different language units from part of speech tags of words to hierarchical structures of sentences (e.g., Wong and Dras (2010); Afroz et al. (2012); Feng et al. (2012a)). In particular, we aim to explore if there exists statistical patterns in deep syntactic structures at phrasal and clausal levels that discriminate deceptive writing from truthful ones, adding a somewhat unconventional angle to prior literature.

One of our primary goals is to quantify the linguistic style of text. In addition to the shallow syntax features, we focus on constituent structural information. What we do is to extract features from syntactic parsing result of text by Berkeley PCFG parser (Petrov and Klein (2007)). Figure 3.6 illustrates the output of syntax parsing of a sentence extracted from a hotel review, a concrete syntax tree struc-

ture. Inspired by previous work (e.g., Wong and Dras (2010); Raghavan et al. (2010)), we render production rules as features based on horizontal slices of a parse tree as shown in Figure 3.6. In a parse tree, there are terminals (leaf nodes) and non-terminals (furcation nodes). We take all possible production rules as features based on the entire tree included or excluded the leaf node (lexical nodes, colored in green in Figure 3.6). To verify the predictive power of different characteristics, especially syntactic versus lexical, we generate two sets of production rules: the lexicalized set that include the production rules involve leaf nodes and the unlexicalized set that excludes the production rules involve leaf nodes. In addition, we also experiment a few variations of production rules, which will be introduced in Section 3.5.2 We will validate various feature encoding presented in Table 3.6.

Bag of Words is one of the most simple representation of text and commonly used for text classification task. Previous work has shown that Bag-of-Words are effective in detecting domain-specific deception (e.g., Ott et al. (2011); Mihalcea and Strapparava (2009)). We consider unigrams, bigrams, and the union of the two as features.

Shallow Syntax As has been used in many previous studies in stylometry (e.g., Mihalcea and Strapparava (2009); Ott et al. (2011)), we utilize part-of-speech (POS) tags to encode shallow syntactic information. Note that Ott et al. (2011) found that even though POS tags are effective in detecting fake product reviews, they are not as effective as words. Therefore, we strengthen POS features with unigram features in this study.

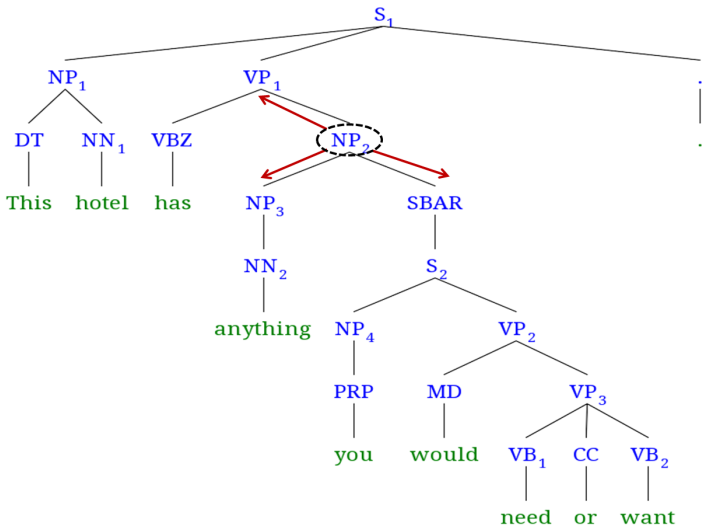


Figure 3.5: Parse Tree (Example I)

Deep Syntactic Characteristics In this work, we obtain the deep syntactic features based on the parsing result. Given a parse tree, we extract rewrite rules by visiting all the treenode N with at least one child node. N corresponds to left side of the production rule and its children corresponds to the right side of the production rule. We form various deep syntactic features by varying the way of traversing parse tree, as illustrated by the red arrows in Figure 3.5. When to capture a bit deeper syntactic structure with a broader text span, we will traverse to upper level of the tree, for instance, to include the parent node, resulting three levels (in vertical direction) involved per production rule. On the other hand, when to extract the syntactic pattern that is not too specific, then we avoid including two many levels.

More specifically, we experiment with four variations of production rules based on the Probabilistic Context Free Grammar (PCFG) parse trees as follows:

- r: the set of regular unlexicalized production rules, e.g., "$S_2 \rightarrow NP_4\ VP_2$ ".

- $\hat{r}$: the set of unlexicalized production rules combined with the grandparent node, e.g., "S_2ˆSBAR $\rightarrow$ NP_4 VP_2 ".
- r_{bi}: the set of unlexicalized production rules , e.g., "S_2ˆADJP $\rightarrow$ RB RB ".
- $r*$: the superset of r, including lexicalized production rules such as "NN$\rightarrow$ *hotel*".
- $\hat{r}*$: the superset of $\hat{r}$, expanding the set by including lexicalized production rules with the grandparent node, e.g. "NNˆNP_1 $\rightarrow$ *hotel*".
- $r_{bi}*$:the superset of r_{bi} including the corresponding lexicalized production rules.

3.5.3 Experiments

In this section, we explore a range of linguistic features to deceptive/truthful reviews classification task for both gold and pseudo-gold standard datasets. For all classification tasks, we use SVM classifier, 80% of data for training and 20% for testing, with 5-fold cross validation. We use LIBLINEAR (Fan et al. (2008)) with L2-regulization, parameter optimized over the training data (3 folds for training, 1 fold for testing). All features are encoded as tf-idf values. We use Berkeley PCFG parser (Petrov and Klein (2007)) to parse sentences. The inflections of the words are normalized and considered as the same words when aggregated for calculating the frequency.

Table 3.6 presents the classification performance using various features across four different datasets introduced earlier. Unlexicalized production rules consider only those production rules that do not lead to terminal nodes (words), focus-

		TripAdvisor		Yelp	Essay		
		Gold	Heur		Abort	BstFr	Death
words	lex_{uni}	*88.4*	**74.4**	59.9	*70.0*	*77.0*	**67.4**
	lex_{bi}	85.8	71.5	**60.7**	71.5	79.5	55.5
	POS_{uni} + lex_{bi}	**89.6**	73.8	60.1	**72.0**	**81.5**	65.5
shallow syntax +words	POS_{uni} + lex_{uni}	87.4	74.0	62.0	70.0	80.0	66.5
	POS_{bi} + lex_{uni}	88.6	74.6	59.0	67.0	82.0	66.5
	POS_{tri} + lex_{uni}	88.6	74.6	59.3	67.0	82.0	66.5
deep syntax	r	78.5	65.3	56.9	62	67.5	55.5
	r_{bi}	75.4	66	58.8	65	70.5	60.0
	$\hat{r}$	74.8	65.3	56.5	58.5	65.5	56.0
	$\hat{r}_{bi}$	73.6	65.1	59	64.5	69.0	62.0
	$r*$	89.4	74.0	64.0	70.1	77.5	66.0
	$r_{bi}*$	89.6	73.4	64.9	68.5	76.5	66.0
	$\hat{r}*$	90.4	75	63.5	71.0	78	67.5
	$\hat{r*_{bi}}$	89.9	73.5	63.6	71.5	76.0	66.0
deep syntax +words	r + lex_{uni}	89.0	74.3	62.3	76.5	82.0	69.0
	r_{bi} + lex_{uni}	88.1	73.5	60.8	78.5	81.5	70.0
	$\hat{r}$ + lex_{uni}	88.5	74.3	62.5	**77.0**	81.5	70.5
	$\hat{r_{bi}}$ + lex_{uni}	88.0	74.2	61.9	75.5	82.5	71.0
	$r*$ + lex_{uni}	90.3	75.4	**64.3**	74.0	**85.0**	**71.5**
	$r_{bi}*$ + lex_{uni}	90.5	74.4	65.3	76.5	83.5	75.5
	$\hat{r}*$ + lex_{uni}	**91.2**	**76.6**	62.1	76.0	84.5	71.0
	$\hat{r_{bi}}*$ + lex_{uni}	89.5	74.2	64.9	76.5	84.5	75.0

Table 3.6: Deception Detection Accuracy (%).

ing only on syntactic structure without exploiting any information on words. In Table 3.6, we vary above deep syntactic features with and without additional unigram features for comparison purposes. In addition, we vary the rewrite rules with respect to how deep the structure is considered (or corresponding text span) by including the parent node. Numbers in *italic* are classification results reported in Ott et al. (2011) and Mihalcea and Strapparava (2009).

3.5.4 Discussion

Over four different datasets spanning from the product review domain to the essay domain, we find that features driven from Context Free Grammar (CFG) parse trees consistently improve the detection performance over several baselines that are based only on shallow lexico-syntactic features. Our results improve the best published result on the hotel review data of Ott et al. (2011), reaching 91.2% accuracy, reducing the error by 14%. We also achieve substantial improvement over the essay data of Mihalcea and Strapparava (2009), obtaining upto 85.0% accuracy. For pseudo gold standard data, we obtain a 2% improvement on accuracy over N-gram lexical model . It is also noted that by using unlexicalized syntax features only, we can obtain a 78.5% and 66% of accuracy respectively. Rewrite rule features perform better in other three datasets as well. Using $rr*$, we are able to achieve 7% or more increase of accuracy of cross-topic classification on lie detection data over the lexicon and shallow syntactic feature set.

Result Analysis

TripAdvisor–Gold We first examine the results for the gold standard hotel reviews shown in Table 3.6. As reported in Ott et al. (2011), bag-of-words features achieve surprisingly high performance, reaching upto 89.6% accuracy. Deep syntactic features, encoded as $\hat{r}*$ slightly improves this performance, achieving 90.4% accuracy. When these syntactic features are combined with unigram features, we attain the best performance of 91.2% accuracy, yielding 14% error reduction over the word-only features.

Given the power of word-based features, one might wonder, whether the PCFG driven features are being useful only due to their lexical production rules. To ad-

dress such doubts, we include experiments with unlexicalized rules, r and $\hat{r}$. These features achieve 78.5% and 74.8% accuracy respectively, which are significantly higher than that of a random baseline ($\sim$50.0%), confirming statistical differences in deep syntactic structures.

Another question one might have is whether the performance gain of PCFG features are mostly from local sequences of POS tags, indirectly encoded in the production rules. Comparing the performance of [shallow syntax+words] and [deep syntax+words] in Table 2, we find statistical evidence that deep syntax based features offer information that are not available in simple POS sequences.

TripAdvisor–Heuristic & Yelp The performance is generally lower than that of the previous dataset, due to the noisy nature of these datasets. These dataset, however, consists of reviews directly from collected from the review site, which is a implementation to the sanctioned gold-stand. Nevertheless, we find similar trends as those seen in the TripAdvisor–Gold dataset, with respect to the relative performance differences across different approaches. The significance of these results comes from the fact that these two datasets consists of real (fake) reviews in the wild, rather than manufactured ones that might invite unwanted signals or bias that can unexpectedly help with classification accuracy. In sum, these results indicate the existence of the statistical signals hidden in deep syntax even in real product reviews with noisy gold standards.

Essay Finally in Table 3.6, the last dataset, a different type of writings - essays, the performance on this dataset confirms the similar trends again, that the deep syntactic features consistently improve the performance over several baselines based only on shallow lexico-syntactic features. The final results, reaching

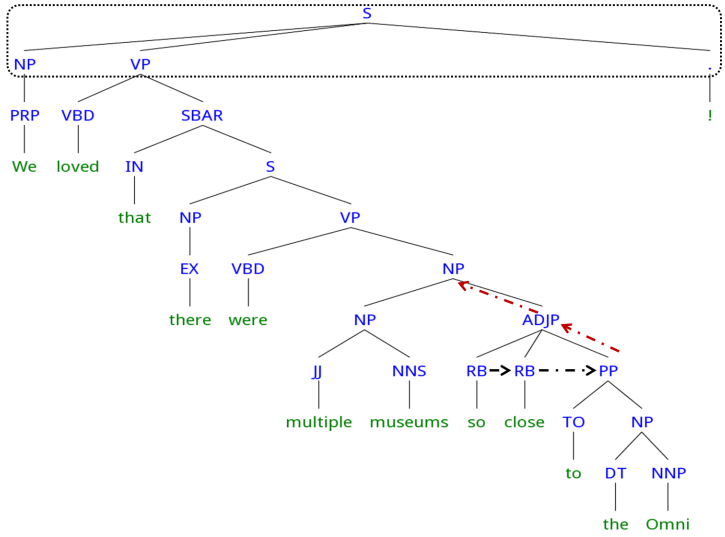


Figure 3.6: Parse Tree (Example II)

TRAINING:	**A** & **B**	**A** & **D**	**B** & **D**
TESTING:	**D**eathPen	**B**estFrn	**A**bortion
M&S 2009	58.7	58.7	62.0
$r*$	66.8	70.9	69.0

Table 3.7: Cross topic deception detection accuracy: Essay data

accuracy as high as 85%, substantially outperform what has been previously reported in Mihalcea and Strapparava (2009). How robust are the syntactic cues in the cross topic setting? Table 3.7 compares the results of Mihalcea and Strapparava (2009) and ours, demonstrating that syntactic features achieve substantially and surprisingly more robust results.

Discriminative Production Rules

In search of more tangible insights, we select the most discriminative deep syntactic features (augmented with the grandparent node) for each hotel review datasets.

	DECEPTION (# 913)	TRUTHFUL (# 1032)
1	NP → DT NNP NNP	NP → $ CD
2	NP → PRP$ NN	S → VP .
3	VP → VBG PP	NP → CD NNS
4	NP → NP SBAR	NP → NNP
5	SBAR → S	NP → QP
6	ADJP → RBS JJ	NP → JJ NN
7	VP → TO VP	NP → QP NNS
8	VP → VB NP PP	S → S : S .
9	NP → PRP$ NNS	NP → NP PP .
10	VP → MD ADVP VP	VP → VBD NP PP

Table 3.8: Most discriminative production rules in gold standard data

	DECEPTION (# 1130)	TRUTHFUL (# 1137)
1	NP → PRP$ NN	VP → VBZ NP
2	NP → DT NNP	NP → EX
3	NP → PRP$ JJ NN	NP → DT JJ NN NN
4	PP → TO NP	NP → DT JJ NN
5	VP → VBD VP	ADJP → RB JJ
6	NP → PRP$ NNS	VP → VBZ UCP
7	VP → VB VP	S → NP , NP VP
8	VP → TO VP	VP → VBZ ADJP
9	ADVP → RB	NP → DT ADJP NN
10	VP → VB SBAR	ADVP → IN RB

Table 3.9: Most discriminative production rules in pseudo gold standard data

The way we select the most discriminative features is the following. We order the rules based on the feature weights assigned by LIBLINEAR classifier. Notice that the two production rules in bolds — [SBARˆNP → S] and [NPˆVP → NP SBAR] — are parts of the parse tree shown in Figure 3.6, the corresponding sentence for parsing is taken from an actual fake review.

Production Rules We list top 10 most discriminative production rules in gold standard dataset as seen in 3.8 and pseudo gold standard dataset as seen in 3.9. For brevity, we only examine one version of production rules, which are specified as unlexicalized without parent node. Much information carried by these production rules correlates with the finding in Ott et al. (2011). Such as there are more facts in numbers ("NP→$CD") included in truthful text and more pronouns ("NP→PRP$ NN") used in deceptive text. Such finding is again confirmed by the results shown in Table 3.9 (row 2 of "DECEPTION" column and row 1 of "TRUTHFUL" column).

The number next to column name is total of unique production rules in corresponding dataset. As we can see, the rule of truthful dataset is only slightly greater than deceptive dataset or comparable for pseudo gold standard (all online reviews) dataset. One the other hand, the difference is greater for the gold standard. We will further investigate whether such bias is introduced by the demographic difference of the two datasets in the separate section.

Sentence Outline We notice that a subset of production rules as that gives us the insight how a sentence is outlined. The rules are positioned at the top-level of a parse tree as marked by a dotted rectangular in Figure 3.6. In particular, such patterns characterize the outline of sentences: beginning part of a sentence, ending part of a sentence, and the order of different modules of a sentence (bigram sequence). For instance, in Figure 3.6, the beginning part of the sentence is denoted as "S $\rightarrow$ SBAR", the end part of the sentence is denoted as "S $\rightarrow$ VP", the sequence of different module is denoted as "SBAR $\rightarrow$ ADVP", "ADVP $\rightarrow$ NP", "NP $\rightarrow$ VP". For analysis purpose, this work also examines the discriminative features of sentence outlines as shown in Table 3.10 and 3.11.

	DECEPTION (# 89)	TRUTHFUL (# 110)
1	S → ADVP , NP VP	S → VP
2	S → SBAR , NP VP	S → S : S
3	S → PP NP VP	NP → NP PP
4	S → NP ADVP VP	S → NP VP
5	S → PP , NP VP	S → CC NP VP
6	S → S , S , CC S	S → NP , NP VP
7	S → S , CC S	S → NP NP VP
8	S → S , S	S → S , NP VP
9	S → S VP	S → S , S CC S
10	S → S CC S	S → ADVP NP VP

Table 3.10: Most discriminative sentence outlines of gold standard data.

As shown in Table 3.10, among the most discriminative sentence outlines of gold standard data, rules (2 and 6-10) in deceptive text indicate that corresponding sentences are complex are is much more than those (2, 8, 9) in truthful text. In addition, a casual writing style can be sensed in truthful text with regards to rules of row 1 and 3, one being without subject and one being noun phrases as sentences.

Similarly, in Table 3.11, most of rules (row 1, 4-7, 9, 10) of deceptive text involves complex sentence structures; in contrast, rules (row 1, 4 - 6) indicates more simple sentences are used in truthful text.

Constituent Tags We also investigate a type of less specific syntactic patterns at constituent level. For this, we refer to the phrasal tag (or non-leaf node) of the PCFG parse tree. Table 3.12 shows the most discriminative phrasal tags for each class. Interestingly. We find deceptive reviews contains VP, SBAR (clause introduced by subordinating conjunction) in than truthful reviews. In addition, there are also more frequent occurrence of WHADVP in deceptive reviews than truth-

	DECEPTION (# 110)	TRUTHFUL (# 108)
1	S → PP NP VP	NP → NP PP
2	S → S , NP VP	S → NP VP
3	S → SBAR , NP VP	S → S : S
4	S → NP ADVP VP	S → ADVP NP VP
5	NP → NNP	S → NP , NP VP
6	S → PP , NP VP	S → S , S , CC S
7	FRAG → NP	S → S , CC S
8	S → SBAR NP VP	S → ADVP , NP VP
9	S → SBAR VP	S → NP VP
10	S → SBAR , S CC S	S → VP

Table 3.11: Most discriminative sentence outlines of pseudo gold standard data.

TRIPADVISOR–GOLD		TRIPADVISOR–HEUR	
DECEP	TRUTH	DECEP	TRUTH
VP	PRN	VP	PRN
SBAR	QP	WHADVP	NX
WHADVP	S	SBAR	WHNP
ADVP	PRT	WHADJP	ADJP
CONJP	UCP	INTJ	WHPP

Table 3.12: Most discriminative phrasal tags in PCFG parse trees: TripAdvisor data.

ful reviews. Such more general information at constituent level could potentially provide more concrete guidance in discerning between the truthful and deceptive writing in practice.

3.5.5 Conclusion

We investigated syntactic stylometry for deception detection, adding a somewhat unconventional angle to previous studies. We also report for the first time that

there are statistical patterns in deep syntactic structure that discriminate deceptive writing from truthful ones. Experimental results consistently find statistical evidence of deep syntactic patterns that are helpful in discriminating deceptive writing. over different datasets demonstrate that features driven from Context Free Grammar (CFG) parse trees consistently improve the detection performance over several baselines that are based only on shallow lexico-syntactic features.

3.6 Demographics

Given that the recent studies in automatic deception detections based on crowdsourcing show great potentials (Mihalcea and Strapparava (2009); Potthast (2010); Ott et al. (2011); Feng et al. (2012b); Rubin and Vashchilko (2012)), some concerns are also raised about crowdsourcing for deceptive data, for instance, Mukherjee et al. (2013) pointed out that the crowdsourced fake reviews may not the representative of real-life fake reviews, indicating that it is questionable that the good performance might potentially be due to the possible differences in the demographics between the crowsourcing user base and real online users. Therefore, we aim to investigate the effects of the demographic factors in terms of corresponding constructed deception corpora. In particular, we focus on the two demographics involved in the several recent study on deception detection (Ott et al. (2011); Feng et al. (2012b)): one is the online review community which is a burgeoning platform of deceptive reviews and the other is the Amazon Mechanic Turk, a popular online crowdsourcing system in which tasks are distributed to a population of thousands of anonymous workers for completion. In rest of the section, we are trying to answer the following questions:

- How does the demographics of participants for writing tasks on Amazon Mechanical Turk differs from that of the general users of review sites?
- If there exists distinguishable differences in the demographics, how does it affect the validity and the efficacy of the datasets constructed by the Amazon MTurk?

3.6.1 Challenges

Challenges on "Direct" Demographic Analysis To verify the differences in demographics of the two communities, is there a way to collect such info of the participants of the communities? How about using the common survey method for identifying the users of each community? Not only the expense will beyond most research institutes can afford, it is practically impossible due to the nature of the task of online deceptive review detection. In fact, it is extremely unnatural for human being to self-report their deceptive behavior (Levine et al. (2010)) and especially for the illicit deceptive online behavior. Since it is not practically feasible to directly obtain the demographics information of each community, we decided to exploit a different direction - evaluation (either intrinsic or extrinsic).

Challenges on Intrinsic Evaluation Remind that the evaluation (i.e., dataset with gold standard labels) of deception detection has remained a challenge. Literature has shown that humans are not good at catching deception in general (e.g., Bond and DePaulo (2006)), and it proves to be no exception for online reviews (Ott et al. (2011)). The implication of this observation is that one practically cannot expect human annotators reliably label existing reviews as deceptive or truthful, except for those less common scenarios where non-linguistic contextual

information such as review time stamps, or IP addresses provide undeniable evidence of dubious acts (e.g., Jindal and Liu (2008); Lim et al. (2010); Mukherjee et al. (2012)). As a result, several previous work has attempted to build deception corpora by instructing participants to lie for a given topic (e.g., Newman et al. (2003)), some of which were based on Amazon Mechanical Turk (e.g., Mihalcea and Strapparava (2009); Potthast (2010); Ott et al. (2011); Afroz et al. (2012)).

Challenges on Data Collection In fact, even for collecting truthful writings for some topics, such as hotel reviews, it is not easy to recruit participants who can actually write truthful stories for any given hotel, especially for luxury hotels, as those who do frequent luxury hotels are less likely to be part of the demographics who write reviews for monetary rewards. Furthermore, Amazon Mechanical Turk workers can pretend to have experienced any hotel that they have never been to and write *fake-truthful* reviews, and it is nearly impractical to verify the authenticity of their claimed experience. As a result, in certain domains such as high-end hotel reviews, it would seem rather unavoidable to assume reviews in the wild to be truthful, subject to filtering rules that minimize accidental inclusion of fake ones (Ott et al. (2011)). 18 out of 20 of hotels in the dataset of Ott et al. (2011) correspond to high-end (4 – 4.5 star rating). Then the validity of the resulting corpus relies on the assumption that the real-world reviews are mostly truthful, which is certainly plausible but difficult to validate.The work of Ott et al. (2012) estimates the prevalence of deceptive reviews in the wild, but the estimation is based on the classifier that is trained based on the same assumption that the real reviews taken from the corresponding review site are mostly truthful. Therefore, the resulting estimates might have been biased toward somewhat conservative measures.

Given the challenges described above, we need to explore the direction of extrinsic evaluation of the existence of variance in the demographics, in other words, the indirect verification on how it affects the performance of the classifier built on the corpora created by crowdsourcing comparing to the one based on online reviews.

3.6.2 Deceptive Review Corpora

In this work, we first take a closer look at this challenge of constructing deception datasets. Specifically, we examine the influence of incidental demographic differences if any, between truthful reviewers (e.g., Yelp users) and deceptive reviewers (i.e., Amazon Mechanical Turk workers), on classification performance. This time, we explore an alternative method of constructing deception dataset by soliciting both truthful and deceptive reviews from the same group of online users, i.e., Amazon Mechanical Turk workers (or Turkers). The plan is to build contrastive corpora of truthful and deceptive reviews taken from different demographic communities. For this purpose, we choose to work with restaurant review domain, as we expect it to be easier to recruit Amazon Mechanical Turk workers who can write actual truthful reviews for restaurants that they have actually been to. We study the reviews in two domains: hotels and restaurants. The data of the two domains are collected from two resources respectively: one is online crowdsourcing - Amazon Mechanical Turk; the other is review websites, e.g. `www.yelp.com` and `www.tripadvisor.com`.

Reviews made to order (Amazon Mechanical Turk): We ask Turkers to complete two writing tasks: one truthful positive review for a restaurant of their own

choice; and one fake positive review for a restaurant that they select from a list we provided. The list of restaurants is randomly sampled from Yelp, and we collect at most one review for each restaurant to avoid incidental biases in restaurant selection. When writing the fake review, Turkers are asked to select a restaurant which they either have never visited before or had a bad impression about. These two different scenarios might cause subtle differences in deception cues. We leave investigation of this effect as future research. Note that we deliberately ask reviewers to write both truthful and deceptive reviews back to back to gain insights on the priming effect (Pickering and Branigan (1999)), but in most experiments we only use the first review from each user. The ordering is randomly selected for each user such that each ordering corresponds to 50% of the entire data collected. To facilitate linguistic analysis, we request the reviews to be reasonably long (100 words or more). We also impose time limit of 1 hour for both tasks. Each paired writing tasks are supposed to be finished within an hours by unique Turkers. Since the Turkers are asked to perform two writing tasks back to back, the priming effect (Branigan et al. (1999)) might affect their writings and the quality of the data. Therefore, we alternate the order the tasks and keep 200 tasks of truthful $\rightarrow$ deceptive and 200 tasks of deceptive $\rightarrow$ truthful.

Restaurant Reviews (Yelp) We identify 400 restaurants from `www.yelp.com` and then collect corresponding reviews for the restaurant from the website. We also collect the reviews for the same set of restaurant as the Amazon Mechanical Turk deceptive writing tasks. But this time, we also collect both truthful and deceptive reviews resulting 400 5-star reviews (with 100 words or more). For these reviews we purposefully make the same assumption as the work by Ott et al.

	$\text{Class}_1 \Leftrightarrow \text{Class}_2$	DECEPTION v.s. Truthful?	VARYING demographics?
1	$\text{D}_{\text{AMT}} \Leftrightarrow \text{T}_{\text{YELP}}$	yes	yes
2	$\text{D}_{\text{AMT}} \Leftrightarrow \text{T}_{\text{AMT}}$	yes	no
3	$\text{T}_{\text{AMT}} \Leftrightarrow \text{T}_{\text{YELP}}$	no	yes

Table 3.13: Three Classification Configurations

(2011), i.e., assume everything as truthful reviews in order to quantitatively investigate the influence of demographic differences in constructing deception corpora.

Hotel Reviews For the hotel domain, we reuse the dataset from the work by (Ott et al. (2011)), which consists of 400 deceptive reviews collected via Amazon MTurk and 400 truthful reviews from TripAdvisor. Unlike restaurant domain, we did not collect the truthful reviews via Amazon Mechanical Turk because we are concerned that the real customers of those hotels are unlikely to work as Turkers and then we may end collecting fake reviews.

Notations: Following notations refer to specific portions of the data constructed above:

- $\mathbf{T_{TRIP}}$ **(**$\mathbf{T_{YELP}}$**):** the set of truthful reviews collected from TripAdvisor (Yelp).
- $\mathbf{D_{AMT}}$ **(**$\mathbf{T_{AMT}}$**):** the set of deceptive (truthful) reviews from AMT (first reviews only).

3.6.3 Experimental Results

Next is to approximate the difference of demographics in the different online communities based on the corresponding datasets. There has been a increasing body

	$D_{AMT} \Leftrightarrow T_{YELP}$	$D_{AMT} \Leftrightarrow T_{AMT}$	$T_{AMT} \Leftrightarrow T_{YELP}$
INFLUENCING FACTORS	Deception|Demograph	Deception	Demograph
POS_{uni}	77.5	71.3	77.8
POS_{bi}	74.8	69.8	76.3
Lex_{uni}	87.0	81.0	89.0
Lex_{bi}	80.8	78.8	84.5
Liwc	78.6	69.8	75.6
All	**89.8**	**81.8**	**92.0**

Table 3.14: Classification Accuracy

of work that examines linguistic differences in writing styles of people with distinctively different demographic backgrounds (e.g., Schler et al. (2006); Wong and Dras (2011); Sarawgi et al. (2011b)), which suggest that demographic differences can introduce some amount of unwanted biases. The question is, how much would it be? Although the demographics of two different sites are likely to be different, there might be enough common grounds making the level of noise practically negligible. In this work, we seek an empirical answer to this question with respect to deceptive review corpora.

To measure the extend to which demographic differences can influence the classification performance, we compare three classification setups summarized in Table 3.13. We use 200 reviews per class, 80% of data for training and 20% for testing. We use LIBLINEAR (Fan et al. (2008)) toolkit with default parameters on the BoW features of review text with 5-fold cross validation. The classification results are shown in Table 3.14. We consider n-gram POS and lexical features, as well as the features derived from LIWC. The classification result of $D_{AMT} \Leftrightarrow T_{YELP}$ is nearly 90%, echoing similar performance of e.g., Ott et al. (2011) in the hotel review domain. Strikingly however, the classification accuracy of $T_{AMT} \Leftrightarrow T_{TRIP}$ is equally as high as 92.0%, suggesting how demographic differences can make the

(POS dist 1) vs (POS dist 2)	Pearson's r	Agree%
($D_{AMT} \Leftrightarrow T_{YELP}$) vs ($D_{AMT} \Leftrightarrow T_{AMT}$)	0.801	85.7
($D_{AMT} \Leftrightarrow T_{YELP}$) vs ($T_{AMT} \Leftrightarrow T_{YELP}$)	-0.389	75.0
($D_{AMT} \Leftrightarrow T_{AMT}$) vs ($T_{AMT} \Leftrightarrow T_{YELP}$)	-0.863	57.1

Table 3.15: Similarity of POS distributions between different pairs of classification setups

classification task unexpectedly easier. Fortunately, the performance of $T_{AMT} \Leftrightarrow T_{YELP}$ that removes demographic biases is not too far away, reaching nearly 82% in accuracy, validating that the existence of linguistic signals in deceptive narratives, while also suggesting that the performance of the first classification is likely to be an overly optimistic measure.

Priming: If we repeat the third classification setting by utilizing both first and second reviews written by each Turker (instead of taking only the first), then the performance drops to 75.4% (from 81.8%) despite that the dataset grows twice. We conjecture that this performance drop is due to priming effects, i.e., the second review written by the same person is inherently influenced by the first review, confusing classifiers. Therefore, when building deception corpora, care needs to be taken in designing the review collection procedure in order to avoid unwanted priming effects.

Despite the reviews are written by reviewers from difference sources for different domains, $non\text{-}conclicted\%$ is above 60% and up-to 70%. This indicates that there exists some linguistic cues for deceptive/truthful writings. We also calculate the $non\text{-}conclicted\%$ for randomly paired the datasets from restaurant or hotel domain, the result is around 50% for the LIWC and POS-based features.

Experimental results reveal surprising insights on the extent to which inciden-

tal demographic differences can influence classification accuracy, while confirming the viability of automatic deception detection in previous studies. In particular, even when the demographic biases are removed, we show that it is possible to achieve deception detection accuracy close to 82%. Additionally, our work results in updated insights on deception cues.

3.6.4 Elements of Deceptive Wring Styles

Next we compare the discriminative features for deceptive and truthful reviews collected from different sources across domains. In Table 3.18, we calculate $non\text{-}conclicted\%$ as the percentage of the discriminative features for the same class and $conflicted\%$ as the percentage of the discriminative features for different classes over all features across two pairs of datasets.

To provide comparative insights on linguistic patterns characterizing fake reviews, we examine the distribution of POS across different datasets and domains. The statistical results shown in Table 3.18 & 3.17 coincide with previous studies that have shown that deceptive reviews have distinctively different distribution of them. We quantify the difference in a feature's distribution between one class (c_1) and the other (c_2) classes by σ. For a feature f, we define σ as the standard divination of ($\frac{v_{f,c_1}}{max(v_{f,c_1},v_{f,c_2})}$, $\frac{v_{f,c_2}}{max(v_{f,c_1},v_{f,c_2}}$). We sort the features by σ in descending order to show the most discriminative features first. If $v_{c_1} > v_{c_2}$, we highlight the feature in the column of c_1. We provide the summary of discriminative features of the hotel review domain (Ott et al. (2011)) as a comparison point.

We find a substantial overlap (i.e., agreement) among salient features (marked in **boldface**, and self referencing (*i, me, my*) are pronounced in deceptive reviews. Not all features are listed in Table 3.18 & 3.17, but the overlap (in boldface) is

checked against the entire set of features.) across different demographics and domains. E.g., VERB and PRONOUN. Conflicting cases are marked with *. Interestingly, superlatives (JJS, RBS) are strongly indicative of deception when truthful reviews are taken from actual review sites, while not as much when truthful reviews are collected from AMT. This observation opens up the need for additional research to investigate the use of superlatives in different deception / truthful contexts.

Table 3.19 shows more detailed differences in POS distributions. Positive (negative) values indicate the corresponding tags are predictive for deceptive (truthful) class. Shaded rows highlight the discrepancy of deception cues due to demographic differences. Note that those POS tabs for which the sign of the numbers of the first two columns match correspond to common linguistic patterns of deceptive (truthful) reviews, regardless of demographic biases. Again, most features have matching signs for the first two columns except for a few exceptions such as JJS, RBS, WDT, WRB. Table 3.16 shows what POS and LIWC categories are in agreement / disagreement (in terms of being discriminative toward either deceptive or truthful) across different classification setups with varying demographic differences. We find that the % of disagreement in LIWC and POS categories are about 35% and 21% respectively. Additionally, the third column in Table 3.19 gives us insights about demographic differences in writing styles between AMT and Yelp users.

3.6.5 Conclusion

We present the first report on the effect of demographic factors in fake review detection. Our work reveals the extent to which demographic differences can

Setup-1: $D_{AMT} \Leftrightarrow T_{YELP}$ Setup-2: $D_{AMT} \Leftrightarrow T_{AMT}$	
AGREEMENT (DECEPTIVE / DECEPTIVE)	28.6%
Dic, achieve, adverb, affect, auxverb, cogmech, discrep, family, friend, funct, future, i, incl, insight, motion, nonfl, past, posemo, ppron, preps, verb, we	
AGREEMENT (TRUTHFUL / TRUTHFUL)	36.4%
AllPct, Apostro, Colon, Comma, Dash, OtherP, Period, SemiC, anger, anx, body, cause, conj, feel, filler, health, humans, inhib, ipron, negate, negemo, number, present, sexual, social, space, they, you	
DISAGREEMENT (DECEPTIVE / TRUTHFUL)	15.6%
Sixltr, article, bio, certain, home, ingest, leisure, money, quant, tentat, time, work	
DISAGREEMENT (TRUTHFUL / DECEPTIVE)	19.5%
Exclam, Parenth, QMark, assent, death, excl, hear, percept, pronoun, relativ, relig, sad, see, shehe, swear	
AGREEMENT (DECEPTIVE / DECEPTIVE)	35.7%
CC, DT, JJ, MD, PRP$, RB, VB, VBD, VBG, VBN	
AGREEMENT (TRUTHFUL / TRUTHFUL)	39.3%
CD, IN, JJR, NN, NNP, NNPS, NNS, PDT, VBP, VBZ, WP	
DISAGREEMENT (DECEPTIVE / TRUTHFUL)	10.7%
JJS, RBS, WDT	
DISAGREEMENT (TRUTHFUL / DECEPTIVE)	10.7%
PRP, RBR, WRB	

Table 3.16: LIWC & POS are grouped per their agreement/disagreement between the two classification setups.

influence classification accuracy, yields a more realistic measure of detection performance, provides updated insights on linguistic cues, and last but not least, a new deception corpus. In particular, even when the demographic biases are removed, we show that it is possible to achieve deception detection accuracy close to 82

3.7 Conclusions

In this study, we mainly investigated on how to identify the intent to deceit in ones writing based on plain text. For this, we examined various linguistic features, in particular, we explored deep syntactic stylometry in a range of datasets that consists of deceptive and truthful writings. Experiment results consistently showed statistical evidence of deep syntactic pattern as effective in discriminating deceptive writings from truthful writings.

3.7.1 Summary of Results and Contributions

In this work, we aimed to seek effective approaches to uncover the intent to deceit based on plain text. For this, we investigated on the efficacy of assorted informative cues and provides insights based on web resources using computational linguistic techniques. In particular, our work is the first to apply deep syntactic stylometry for identifying deceptive writing, adding a somewhat unconventional angle to previous literature.

For deception detection, the evaluation has always been a challenge, as it is almost impossible to manually determine whether a writing is truthful or not. In this work, we presented a novel evaluation strategy that exploits existing gold standard, and empirically validated the connection between the performance evaluated using the gold standard and the performance evaluated using only the pseudo gold standard data. Our work on constructing the pseudo gold standard is the first to provide a comprehensive, direct and large-scale analysis on representative distribution of product reviews, accompanying quantitative evaluations that are not based on human judgments.

Previous work proposed to use crowd sourcing to collect deceptive writings to create gold standard, which is a promising alternative to human annotated data. However, it also raises the concern of the bias that might be caused by the demographic difference between crowd sourcing and the online reviewers. We addressed this concern, presenting the first report on the effect of demographic factors in detecting the deceptive reviews. Our work revealed the extent to which demographic differences can influence classification accuracy, yields a more realistic measure of detection performance, provided updated insights on linguistic cues, and last but not least, a new deception corpus.

3.7.2 Future Work

There has been a burgeoning research have been put in the solving the problem of deception detection, the task itself remains challenging. We hypothesis that the writing style of deceptive writings in the digital communication may change over time and distinctively different across domain, therefore we endeavor to develop practice-oriented, computationally effective approaches without relying on human judges. Even though we try to tackle the problem automatically, we would like to further our study on interpreting the statistical evidences so as to unveil the possible intuition or potential connection to psychological insights.

Stylistic Patterns

One interesting question is whether we dig out something more from deep syntactic structures, which can be potentially characterized in a collective way. This could be a starting point to interpret the abstract syntactic differences between deceptive and truthful writing. As an attempt to reduce this gap between modern

statistical parsers and cognitively recognizable stylistic elements, we will explore two complementary approaches:

1. Translate the PCFG parsing results with certain stylistic elements of rhetoric or rhetorical devices. Since rhetoric is the technique that an author applies to convey to the readers a meaning with the goal of persuading him or her towards considering a topic from a different perspective,

2. Mine the frequent patterns extracted from the tree structures of PCFG parsing result. For this, we will systematically compute the variations of the subtrees as syntactic patterns, which may include a pattern mixed with both phrasal node and lexical node as long as the pattern is frequent.

Discourse Analysis

So far, we have processed and analysed sentences in ones writing separately and have not considered the discourse relations between sentences for deception detection. For future exploration, we may consider, for instance, to what extent a text is coherent and what cohesive devices are used to achieve the particular level of coherence of the text, and whether truthful writings are necessarily more cohesive than deceptive ones in terms of usage of words and changes of sentence structures across sentences. We also would like to explore the possible intuitive way to visualize the changes in language usage between text chunks to assist human-beings to review and analyze the statistics.

Collective Evidence in Detecting Deception

Endless effort has been put in the study that uncovers various cues and anomalous behavior that differentiate deceptive writings from the truthful ones in online review communities. However, the majority studies, naturally and rightfully, have focused on the empirical effectiveness of a particular type of deception cues targeted for a specific sub-group of online users in a specific domain. As a result, various findings remain somewhat unconnected, calling for insights into the interplay among different types of strategies in detecting deception . In the future work, we plan to develop an algorithmic framework that can potentially incorporate various sources of information for detecting the intent of deceit. Hence, we aim to find connection between evidences of different types of cues that analyzed at scale: (I) circumstantial patterns (of each user), (II) distributional anomaly of opinions (of each product), and (III) linguistic patterns of deception (of each review). We will consider online hotel review domain as a starting point.

HOTEL		RESTAURANT	
D-AMT	T-TRIP	D-AMT	T-YELP
LIWC (Top10)			
family	**OtherP**	ingest*	**OtherP**
i	**Dash**	**certain**	**Dash**
Exclam*	negemo*	**leisure**	**Apostro**
feel*	filler	**bio**	**AllPct**
ppron	**negate**	**motion**	cause*
you*	we*	past*	**Period**
certain	**AllPct**	posemo	Exclam*
they*	**excl**	we*	**excl**
cause*	**Period**	**auxverb**	ipron*
pronoun	ingest*	**discrep**	they*
future	money*	Sixltr*	**Comma**
see	**home**	**affect**	tentat*
insight	**Apostro**	work*	**space**
friend	**number**	incl*	present*
discrep*	**space**	**i**	conj*
POS (TOP 10)			
RBS	**CD**	**RBS**	WP$
RBR*	**NNPS**	**VBD**	**NNPS**
PRP$	**JJR**	**MD**	**CD**
WRB*	NNS	**PRP$**	RBR*
JJS	CC*	JJ*	**PDT**
VB	VBZ	**JJS**	WP*
MD	JJ*	**VBG**	**JJR**
VBG	**PDT**	**VB**	**VBP**
PRP	DT*	CC*	WRB*
WP*	VBN	DT*	**NNP**
POS^{cate}			
PRONOUN	**CD**	ADJ*	**CD**
WH*	ADJ*	**VERB**	**PRE-DT**
VERB	**PRE-DT**	DT	WH*
ADV*	DT*	**PRONOUN**	**NOUN**
-	**NOUN**	-	**PREP**
-	**PREP**	-	ADV*

Table 3.17: Comparison of Prominent POS features (sorted by σ) — hotel domain v.s. restaurant domain

RESTAURANT		RESTAURANT	
D_{AMT}	T_{YELP}	D_{AMT}	T_{AMT}
LIWC (Top 15)			
ingest*	**OtherP**	**past**	**they**
certain*	**Dash**	**we**	**present**
leisure*	**Apostro**	**i**	**Dash**
bio*	**AllPct**	**discrep**	**you**
motion	**cause**	**achieve**	**cause**
past	**Period**	percept	work*
posemo	Exclam*	**motion**	leisure*
we	excl*	**ppron**	bio*
auxverb	ipron*	**insight**	ingest*
discrep	**they**	pronoun*	**tentat**
Sixltr*	**Comma**	**adverb**	**social**
affect	**tentat**	**posemo**	**Period**
work*	**space**	**incl**	certain*
incl	**present**	**affect**	**Apostro**
i	**conj**	ipron*	**conj**
POS (Top 10)			
RBS*	WP$	**VBD**	**NNPS**
VBD	**NNPS**	**MD**	**VBP**
MD	**CD**	**VBG**	**VBZ**
PRP$	RBR*	PRP	**PDT**
JJ	**PDT**	RBR*	**NNS**
JJS*	**WP**	**VB**	**CD**
VBG	**JJR**	**WRB**	**WP**
VB	**VBP**	**PRP$**	**JJR**
CC	**WRB**	**JJ**	**NNP**
DT	**NNP**	**RB**	RBS*
POS^{cate}			
ADJ	**CD**	**PRONOUN**	**PRE-DT**
DT*	**PRE-DT**	**VERB**	**CD**
VERB	**WH**	**ADJ**	**NOUN**
PRONOUN	PREP*	ADV*	**WH**
-	**NOUN**	-	PREP*
-	ADV*	-	DT*

Table 3.18: Comparison of Prominent POS features (sorted by σ) — real reviews (T_{YELP}) v.s. AMT reviews (T_{AMT})

	$D_{AMT} \Leftrightarrow T_{YELP}$		$D_{AMT} \Leftrightarrow T_{AMT}$	$T_{AMT} \Leftrightarrow T_{YELP}$
INFLUENCE(FACTOR)	DECEPTION	DEMOGRAPH	DECEPTION	DEMOGRAPH
POS Categories				
ADJ	+.0497		+.0122	+.0385
ADV	+.0109		+.0207	-.01
CD	-.268		-.1103	-.2024
DT	+.0278		-.0	+.0278
NOUN	-.0352		-.036	+.0008
PRE-DT	-.1378		-.1591	+.0294
PREP	-.0382		-.031	-.0077
PRONOUN	+.0102		+.0467	-.0372
VERB	+.0326		+.0279	+.0051
WH	-.0728		-.0424	-.0332
Selected POS Tags				
NN	-.0202		-.003	-.0173
NNP	-.0661		-.0474	-.0206
VB	+.0322		+.0203	+.0124
VBD	+.1724		+.3671	-.2972
VBG	+.0294		+.109	-.0846
VBZ	-.093		-.2298	+.1681
VBP	-.1235		-.2211	+.1296
VBN	+.0771		+.0393	+.041
MD	+.1127		+.0432	+.0761
JJ	+.0633		+.0173	+.0476
JJR	-.1657		-.0066	-.1612
JJS	+.075		-.0934	+.1544
RB	+.0228		+.0214	+.0014
RBR	-.2246		+.0918	-.2752
RBS	+.1954		-.0677	+.2366
PDT	-.1307		-.1587	+.0379
CD	-.2635		-.1099	-.1969
WDT	+.0107		-.0536	+.0632
WP	-.1943		-.0926	-.1248
WRB	-.0449		+.0018	-.0465
PRP	-.0		+.0447	-.0447
PRP$	+.0914		+.056	+.0399
CC	+.0351		+.0026	+.0327
DT	+.0369		+.0005	+.0364
IN	-.0292		-.0306	+.0015

Table 3.19: Differences (σ) of the distribution of POS tags on Restaurant Reviews.

Chapter 4

Conclusions

4.1 Conclusions and Future Work

We have presented our study on recognizing the authorial intention embedded in the one's writings, which potentially enable a deeper understanding of human communications in various circumstances. In particular, our exploration has been focused on two aspects: one is learning the connotative meaning; one is detecting the intent to deceit in the writing. For the authorial intention on connotation, we attained the first large scale connotation lexicon, which was learned over an extensive network of words and senses via collective inference approaches based on rich linguistic resources. For the authorial intent on deception, we exploited the predicting power of a range of stylistic elements on assorted corpora in varied domains and also constructed our own datasets for a more thorough comparison. In this work, we primarily exploited linguistically motivated features within the collection of short documents, followed the classification paradigm and finally provided some empirical insight into algorithms. Our evaluation results validated

both the quality of resulting datasets and the effectiveness of our proposed strategies. Multiple datasets generated in this work were made publicly available for the use of research and practice.

In this work, we presented a focus study on the connotation and deception as part of the research of learning the authorial intention in the writings. Thus, we have began to solve small pieces for the hard puzzle. By virtue of of that, there are considerable opportunities remained for additional work on identifying the intended information in the natural language texts.

Given that we obtained promising results on the identifying the polarity of lexical connotation, we are motivated to learn such information based on lexical correlations, which are quantified based on various linguistic insights. For the future work, we aim to "replicate the success" of learning the general connotation when we extend our study to learning other kind of "intented" information carried by words or phrases. In other words, for applications that involved classification based on statistical dependence and graph representations, we could start from the inference framework that we developed for the task of learning the connotation. For instance, if we would like to detect the intention of insulting, we need to differentiate insulting words or phrases from general connotatively negative ones. The number of insulting words is likely to be significantly smaller than the non-insulting words. Therefore, in order to generalize the framework, it needs to be further developed to handle the imbalanced distribution among classes, etc..

Recognizing the intended information in the natural language text is profound to better understanding the human communications. In this work, our study was primarily based on short documents, which demonstrate the communication between writers and readers. We are also interested in other types of communication

such as dialogues or conversations in the text-based setting. This particular type of communication entails the interaction in diverse social contexts; it also naturally provides the consequences of a communication. For instance, the reaction of one conversational participant could be in turn utilized as an indicator of certain type of intention by the other conversational participant in the previous utterance. We hypothesize that for this particular type of communicating, there might be frequent and rich linguistic phenomenons that are coupled with secondary intention. This will create new opportunities for learning the authorial intention. In addition, the dialogical analysis is closely related to the interpretative analysis of spoken or written utterances from linguistic perspective. Therefore, we also could make use of linguistic insights we gained in this work for dialogical analysis.

Bibliography

Ahmed Abbasi and Hsinchun Chen. Applying authorship analysis to extremist-group web forum messages. *Intelligent Systems, IEEE*, 20(5):67–75, 2005.

Alina Adreevskaia and Sabine Bergler. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 209–216, 2006.

Sadia Afroz, Michael Brennan, and Rachel Greenstadt. Detecting hoaxes, frauds, and deception in writing style online. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 461–475. IEEE, 2012.

Cem Akkaya, Janyce Wiebe, and Rada Mihalcea. Subjectivity word sense disambiguation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 190–199. Association for Computational Linguistics, 2009.

Shlomo Argamon, Marin Šarić, and Sterling S Stein. Style mining of electronic messages for multiple authorship discrimination: first results. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 475–480. ACM, 2003.

Shlomo Argamon, Casey Whitelaw, Paul Chase, Sobhan Raj Hota, Navendu Garg, and Shlomo Levitan. Stylistic text classification using functional lexical features. *Journal of the American Society for Information Science and Technology*, 58(6):802–822, 2007.

Harald Baayen, Hans Van Halteren, and Fiona Tweedie. Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution. *Literary and Linguistic Computing*, 11(3):121–132, 1996.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may 2010. European Language Resources Association (ELRA). ISBN 2-9517408-6-7.

Alexandra Balahur, Rada Mihalcea, and Andrés Montoyo. Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications. *Computer Speech & Language*, 28(1):1–6, 2014.

Carmen Banea, Rada Mihalcea, and Janyce Wiebe. Sense-level subjectivity in a multilingual setting. *Computer Speech & Language*, 28(1):7–19, 2014.

Monica Bianchini, Marco Gori, and Franco Scarselli. Inside pagerank. *ACM Trans. Internet Technol.*, 5:92–128, February 2005. ISSN 1533-5399. doi: http://doi.acm.org/10.1145/1052934.1052938. URL `http://doi.acm.org/10.1145/1052934.1052938`.

John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

J. Kathryn Bock. Syntactic persistence in language production. *Cognitive psychology*, 18(3):355–387, 1986.

Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*, 2011.

Charles F Bond and Bella M DePaulo. Accuracy of deception judgments. *Personality and social psychology Review*, 10(3):214–234, 2006.

Gary D Bond and Adrienne Y Lee. Language of lies in prison: Linguistic classification of prisoners' truthful and deceptive natural language. *Applied Cognitive Psychology*, 19(3):313–329, 2005.

Holly P Branigan, Martin J Pickering, and Alexandra A Cleland. Syntactic priming in written production: Evidence for rapid decay. *Psychonomic Bulletin & Review*, 6(4):635–640, 1999.

Thorsten Brants and Alex Franz. {Web 1T 5-gram Version 1}. 2006.

Michael Brennan, Sadia Afroz, and Rachel Greenstadt. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):12, 2012.

John Burrows. delta: A measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, 17(3):267–287, 2002.

Avner Caspi and Paul Gorsky. Online deception: Prevalence, motivation, and emotion. *CyberPsychology & Behavior*, 9(1):54–59, 2006.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Yanqing Chen and Steven Skiena. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 383–389, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P14/P14-2063.

Yejin Choi and Claire Cardie. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 590–598, Singapore, August 2009. Association for Computational Linguistics.

Cindy K Chung and James W Pennebaker. Revealing dimensions of thinking in open-ended self-descriptions: An automated meaning extraction method for natural language. *Journal of Research in Personality*, 42(1):96–132, 2008.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16:22–29, March 1990. ISSN 0891-2017. URL http://portal.acm.org/citation.cfm?id=89086.89095.

Kevyn Collins-Thompson and Jamie Callan. Automatic and human scoring of word definition responses. In *HLT-NAACL*, pages 476–483, 2007.

ILOG CPLEX. High-performance software for mathematical programming and optimization. *U RL http://www.ilog.com/products/cplex*, 2009.

Fermín L Cruz, José A Troyano, Beatriz Pontes, and F Javier Ortega. Building layered, multilingual sentiment lexicons at synset and lemma levels. *Expert Systems with Applications*, 2014.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116. Association for Computational Linguistics, 2010.

Olivier De Vel, Alison Anderson, Malcolm Corney, and George Mohay. Mining e-mail content for author identification forensics. *ACM Sigmod Record*, 30(4): 55–64, 2001.

Chrysanthos Dellarocas. Strategic manipulation of internet opinion forums: Implications for consumers and firms. In *Management Science, Vol. 52, No. 10*, 2006.

Bella M DePaulo, James J Lindsay, Brian E Malone, Laura Muhlenbruck, Kelly Charlton, and Harris Cooper. Cues to deception. *Psychological bulletin*, 129 (1):74, 2003.

Xishuang Dong, Qibo Zou, and Yi Guan. Set-similarity joins based semi-supervised sentiment analysis. In *Neural Information Processing*, pages 176–183. Springer, 2012.

Paul Ekman and Maureen O'Sullivan. Who can catch a liar? *American psychologist*, 46(9):913, 1991.

Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res.(JAIR)*, 22(1):457–479, 2004.

Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*, pages 417–422, 2006.

Andrea Esuli and Fabrizio Sebastiani. Pageranking wordnet synsets: An application to opinion mining. In *Proceedings of the 45th Annual Meeting*

of the Association of Computational Linguistics, pages 424–431. Association for Computational Linguistics, 2007. URL http://www.aclweb.org/anthology/P/P07/P07-1054.pdf.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Geli Fei, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Exploiting burstiness in reviews for review spammer detection. In *ICWSM*, 2013.

Song Feng, Ritwik Bose, and Yejin Choi. Learning general connotation of words using graph-based algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1092–1103. Association for Computational Linguistics, 2011.

Song Feng, Ritwik Banerjee, and Yejin Choi. Characterizing stylistic elements in syntactic structure. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1522–1533. Association for Computational Linguistics, 2012a.

Song Feng, Ritwik Banerjee, and Yejin Choi. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 171–175, Stroudsburg, PA, USA, 2012b. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390665.2390708.

Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. Distributional footprints of deceptive product reviews. In *Proceedings of the 2012 International AAAI Conference on WebBlogs and Social Media, June*, 2012c.

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 (Volume 1: Long Papers)*, pages 1774–1784, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Howard S Friedman and Joan S Tucker. Language and deception. 1990.

John J Furedy and Gershon Ben-Shakhar. The roles of deception, intention to deceive, and motivation to avoid detection in the psychophysiological detection of guilty knowledge. *Psychophysiology*, 28(2):163–171, 1991.

Stephanie Gokhman, Jeff Hancock, Poornima Prabhu, Myle Ott, and Claire Cardie. In search of a gold standard in studies of deception. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*, pages 23–30. Association for Computational Linguistics, 2012.

Stephan Greene and Philip Resnik. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 503–511, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Jeffrey T Hancock, Jennifer Thom-Santelli, and Thompson Ritchie. Deception and design: The impact of communication technology on lying behavior. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 129–134. ACM, 2004.

Jeffrey T Hancock, Lauren E Curry, Saurabh Goorha, and Michael Woodworth. On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. *Discourse Processes*, 45(1):1–23, 2007.

Christopher Harris. Detecting deceptive opinion spam using human computation. In *Workshops at AAAI on AI*, 2012.

Vasileios Hatzivassiloglou and Kathleen McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the Joint ACL/EACL Conference*, pages 174–181, 1997.

Taher H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the Eleventh International World Wide Web Conference*, Honolulu, Hawaii, 2002.

Bas Heerschop, Alexander Hogenboom, and Flavius Frasincar. Sentiment lexicon creation from lexical resources. In *Business Information Systems*, pages 185–196. Springer, 2011.

Graeme Hirst and Olga Feiguina. Bigrams of syntactic labels for authorship discrimination of short texts. *Literary and Linguistic Computing*, 22(4):405–417, 2007.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM. ISBN 1-58113-888-1.

Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the international conference on Web search and web data mining*, WSDM '08, pages 219–230, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-927-2. doi: http://doi.acm.org/10.1145/1341531.1341560. URL `http://doi.acm.org/10.1145/1341531.1341560`.

Nitin Jindal, Bing Liu, and Ee-Peng Lim. Finding unusual review patterns using unexpected rules. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1549–1552, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5.

Marcia K Johnson and Carol L Raye. Reality monitoring. *Psychological review*, 88(1):67, 1981.

Nobuhiro Kaji and Masaru Kitsuregawa. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1075–1083, 2007.

Jaap Kamps, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. Using wordnet to measure semantic orientation of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, pages 1115–1118, 2004.

Jun Seok Kang, Song Feng, Leman Akoglu, and Yejin Choi. Connotationwordnet: Learning connotation over the word+sense network. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1544–1554, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P14/P14-1145`.

Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *COLING '04: Proceedings of the 20th international conference on Computational*

Linguistics, page 1367, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

Ross Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. 1980.

Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *JOURNAL OF THE ACM*, 46(5):604–632, 1999.

Robert E Kraut. Verbal and nonverbal cues in the perception of lying. *Journal of personality and social psychology*, 36(4):380, 1978.

Raymond Y. K. Lau, S. Y. Liao, Ron Chi-Wai Kwok, Kaiquan Xu, Yunqing Xia, and Yuefeng Li. Text mining and probabilistic language modeling for online review spam detection. *ACM Trans. Manage. Inf. Syst.*, 2(4):25:1–25:30, January 2012. ISSN 2158-656X. doi: 10.1145/2070710.2070716. URL `http://doi.acm.org/10.1145/2070710.2070716`.

Semi-Supervised Learning. Cluster kernels for semi-supervised learning. 2003.

Kevin Lerman, Sasha Blair-Goldensohn, and Ryan McDonald. Sentiment summarization: Evaluating and learning user preferences. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 514–522, Athens, Greece, March 2009. Association for Computational Linguistics.

Timothy R Levine, Rachel K Kim, and J Pete Blair. (in) accuracy at detecting true and false confessions and denials: An initial test of a projected motive model of veracity judgments. *Human Communication Research*, 36(1):82–102, 2010.

Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. Learning to identify review spam. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2488, 2011.

Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 939–948, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0099-5. doi: http://doi.acm.org/10.1145/1871437.1871557. URL `http://doi.acm.org/10.1145/1871437.1871557`.

Bing Liu and Lei Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. Springer, 2012.

Bill Louw. Irony in the text or insincerity in the writer. *Text and technology: In honour of John Sinclair*, pages 157–176, 1993.

Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *Proceedings of the 20th international conference on World wide web*, pages 347–356. ACM, 2011.

Thomas Corwin Mendenhall. The characteristic curves of composition. *Science*, (214S):237–246, 1887.

Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.

Rada Mihalcea and Carlo Strapparava. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 309–312. Association for Computational Linguistics, 2009.

Rada Mihalcea and Paul Tarau. Textrank: Bringing order into texts. Association for Computational Linguistics, 2004.

Rada Mihalcea, Carmen Banea, and Janyce Wiebe. Multilingual subjectivity and sentiment analysis. In *Tutorial Abstracts of ACL 2012*, pages 4–4. Association for Computational Linguistics, 2012.

George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38 (11):39–41, 1995. ISSN 0001-0782.

Saif Mohammad and Peter Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 26–34, Los Angeles, CA, June 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W10-0204`.

Saif Mohammad, Cody Dunne, and Bonnie Dorr. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 599–608, Singapore, August 2009. Association for Computational Linguistics.

Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403, 2008.

Arturo Montejo-Ráez, Eugenio Martínez-Cámara, M. Teresa Martín-Valdivia, and L. Alfonso Ureña López. Random walk weighting over sentiwordnet for sentiment polarity detection on twitter. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 3–10, Jeju, Korea, July 2012. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-3703.

Arjun Mukherjee, Bing Liu, Junhui Wang, Natalie Glance, and Nitin Jindal. Detecting group review spam. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 93–94, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0637-9. doi: 10.1145/1963192.1963240. URL http://doi.acm.org/10.1145/1963192.1963240.

Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 191–200, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187863. URL http://doi.acm.org/10.1145/2187836.2187863.

Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie S Glance. What yelp fake review filter might be doing? In *ICWSM*, 2013.

David Newman, Sarvnaz Karimi, and Lawrence Cavedon. External evaluation of topic models. In *Australasian Document Computing Symposium*, pages 11–18, Sydney, December 2009. ISBN 978-1-74210-171-2.

Matthew L Newman, James W Pennebaker, Diane S Berry, and Jane M Richards. Lying words: Predicting deception from linguistic styles. *Personality and Social Psychology Bulletin*, 29(5):665–675, 2003.

Zheng-Yu Niu, Dong-Hong Ji, and Chew Lim Tan. Word sense disambiguation using label propagation based semi-supervised learning. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 395–402, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219889. URL http://dx.doi.org/10.3115/1219840.1219889.

Charles Egerton Osgood, George John Suci, and Percy H Tannenbaum. *The measurement of meaning*, volume 47. Urbana: University of Illinois Press, 1957.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 309–319, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1032.

Myle Ott, Claire Cardie, and Jeff Hancock. Estimating the prevalence of deception in online review communities. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 201–210, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1229-5. doi: 10.1145/2187836.2187864. URL http://doi.acm.org/10.1145/2187836.2187864.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *WWW*, pages 201–210, 2007.

Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1-2):1–135, 2008. ISSN 1554-0669.

James W Pennebaker and Lori D Stone. Words of wisdom: language use over the life span. *Journal of personality and social psychology*, 85(2):291, 2003.

James W Pennebaker, Matthias R Mehl, and Kate G Niederhoffer. Psychological aspects of natural language use: Our words, our selves. *Annual review of psychology*, 54(1):547–577, 2003.

James W Pennebaker, Cindy K Chung, Molly Ireland, Amy Gonzales, and Roger J Booth. The development and psychometric properties of liwc2007. *Austin, TX, LIWC. Net*, 2007.

John P Pestian, Pawel Matykiewicz, Michelle Linn-Gust, Brett South, Ozlem Uzuner, Jan Wiebe, K Bretonnel Cohen, John Hurdle, Christopher Brew, et al. Sentiment analysis of suicide notes: A shared task. *Biomedical Informatics Insights*, 5(Suppl. 1):3, 2012.

Slav Petrov and Dan Klein. Improved inference for unlexicalized parsing. In *Proceedings of NAACL HLT 2007*, pages 404–411, 2007.

Martin J. Pickering and Holly P. Branigan. The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and Language*, 39:633–651, 1998.

Martin J Pickering and Holly P Branigan. Syntactic priming in language production. *Trends in cognitive sciences*, 3(4):136–141, 1999.

Emily Pitler and Ani Nenkova. Revisiting readability: a unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 186–195, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1613715.1613742`.

Stephen Porter and John C Yuille. The language of deceit: An investigation of the verbal clues to deception in the interrogation context. *Law and Human Behavior*, 20(4):443, 1996.

Martin Potthast. Crowdsourcing a wikipedia vandalism corpus. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 789–790. ACM, 2010.

Martin Potthast, Benno Stein, Alberto Barrón-Cedeño, and Paolo Rosso. An evaluation framework for plagiarism detection. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 997–1005, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1944566.1944681`.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 1199–1204, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1661445.1661637.

Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. Authorship attribution using probabilistic context-free grammars. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 38–42. Association for Computational Linguistics, 2010.

Delip Rao and Deepak Ravichandran. Semi-supervised polarity lexicon induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 675–682, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1609067.1609142.

Antonio Reyes, Paolo Rosso, and Davide Buscaldi. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 2012.

Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112. Association for Computational Linguistics, 2003.

Dan Roth and Wen-tau Yih. *A linear programming formulation for global inference in natural language tasks*. Defense Technical Information Center, 2004.

Victoria L Rubin and Tatiana Vashchilko. Identification of truth and deception in text: Application of vector space model to rhetorical structure theory. In *Proceedings of the Workshop on Computational Approaches to Deception Detection*, pages 97–106. Association for Computational Linguistics, 2012.

Ruchita Sarawgi, Kailash Gajulapalli, and Yejin Choi. Gender attribution: tracing stylometric evidence beyond topic and genre. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 78–86, Stroudsburg, PA, USA, 2011a. Association for Computational Linguistics.

Ruchita Sarawgi, Kailash Gajulapalli, and Yejin Choi. Gender attribution: tracing stylometric evidence beyond topic and genre. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 78–86. Association for Computational Linguistics, 2011b.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 199–205, 2006.

Shalom H Schwartz, Gila Melech, Arielle Lehmann, Steven Burgess, Mari Harris, and Vicki Owens. Extending the cross-cultural validity of the theory of basic human values with a different method of measurement. *Journal of cross-cultural psychology*, 32(5):519–542, 2001.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29 (3):93–106, 2008.

John Sinclair. *Corpus, concordance, collocation*. Describing English language. Oxford University Press, 1991. ISBN 9780194371445. URL `http://books.google.com/books?id=L8l4AAAAIAAJ`.

Efstathios Stamatatos, George Kokkinakis, and Nikos Fakotakis. Automatic text categorization in terms of genre and author. *Comput. Linguist.*, 26(4):471–495, 2000.

Efstathios Stamatatos, Nikos Fakotakis, and George Kokkinakis. Computer-based authorship attribution without lexical measures. *Computers and the Humanities*, 35(2):193–214, 2001.

Anatol Stefanowitsch and Stefan Th Gries. Collostructions: Investigating the interaction of words and constructions. *International journal of corpus linguistics*, 8(2):209–243, 2003.

Philip J. Stone and Earl B. Hunt. A computer approach to content analysis: studies using the general inquirer system. In *Proceedings of the May 21-23, 1963, spring joint computer conference*, AFIPS '63 (Spring), pages 241–256, New York, NY, USA, 1963. ACM. doi: http://doi.acm.org/10.1145/1461551.1461583. URL `http://doi.acm.org/10.1145/1461551.1461583`.

Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: affective text. In *SemEval '07: Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74, Morristown, NJ, USA, 2007a. Association for Computational Linguistics.

Carlo Strapparava and Rada Mihalcea. Semeval-2007 task 14: Affective text. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 70–74. Association for Computational Linguistics, 2007b.

Michael Stubbs. Collocations and semantic profiles: on the cause of the trouble with quantitative studies. *Functions of language*, 2(1):23–55, 1995.

Fangzhong Su and Katja Markert. Subjectivity recognition on word senses via semi-supervised mincuts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1–9. Association for Computational Linguistics, 2009. URL `http://www.aclweb.org/anthology/N/N09/N09-1001`.

Hiroya Takamura, Takashi Inui, and Manabu Okumura. Extracting semantic orientations of words using spin model. In *Proceedings of ACL-05, 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, US, 2005. Association for Computational Linguistics.

Catalina L Toma and Jeffrey T Hancock. Reading between the lines: linguistic cues to deception in online dating profiles. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pages 5–8. ACM, 2010.

Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *In EMNLP/VLC 2000*, pages 63–70, 2000.

Peter D. Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the Twelfth European Conference on Machine Learning (ECML-01)*, pages 491–502, Freiburg, Germany, 2001.

Leonid Velikovich, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010.

Aldert Vrij. *Detecting lies and deceit: The psychology of lying and implications for professional practice*. Wiley & Sons, 2000.

Aldert Vrij, Samantha Mann, Susanne Kristen, and Ronald P Fisher. Cues to deception and ability to detect lies as a function of police interview styles. *Law and human behavior*, 31(5):499–518, 2007.

Guan Wang, Sihong Xie, Bing Liu, and Philip S. Yu. Identify online store review spammers via social review graph. *ACM Trans. Intell. Syst. Technol.*, 3(4):61:1–61:21, September 2012. ISSN 2157-6904. doi: 10.1145/2337542.2337546. URL `http://doi.acm.org/10.1145/2337542.2337546`.

Janyce Wiebe and Rada Mihalcea. Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1065–1072. Association for Computational Linguistics, 2006.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation (formerly Computers and the Humanities)*, 39(2/3):164–210, 2005.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. Opinionfinder: a system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

Ellen Winner and Sue Leekam. Distinguishing irony from deception: Understanding the speaker's second-order intention. *British Journal of Developmental Psychology*, 9(2):257–270, 1991.

Sze-Meng Jojo Wong and Mark Dras. Parser features for sentence grammaticality classification. In *Proceedings of the Australasian Language Technology Association Workshop 2010*, pages 67–75, Melbourne, Australia, December 2010. URL `http://www.aclweb.org/anthology/U/U10/U10-1011`.

Sze-Meng Jojo Wong and Mark Dras. Exploiting parse structures for native language identification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1600–1610. Association for Computational Linguistics, 2011.

Guangyu Wu, Derek Greene, Barry Smyth, and Pádraig Cunningham. Distortion as a validation criterion in the identification of suspicious reviews. In *Proceedings of the First Workshop on Social Media Analytics*, pages 10–13. ACM, 2010.

Rui Xie and Chunping Li. Lexicon construction: A topic model approach. In *Systems and Informatics (ICSAI), 2012 International Conference on*, pages 2299–2303. IEEE, 2012.

Sihong Xie, Guan Wang, Shuyang Lin, and Philip S. Yu. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 823–831, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339662. URL `http://doi.acm.org/10.1145/2339530.2339662`.

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring AI in the new millennium*, pages 239–269. 2003.

Kyung-Hyan Yoo and Ulrike Gretzel. Comparison of deceptive and truthful travel reviews. In *Information and Communication Technologies in Tourism*, pages 37–47. Springer Vienna, 2009.

G Udny Yule. A test of tippett's random sampling numbers. *Journal of the Royal Statistical Society*, 101(1):167–172, 1938.

George Udny Yule and G Udny Yule. *The statistical study of literary vocabulary*. Cambridge Univ Press, 1944.

Rong Zhang, ChaoFeng Sha, Minqi Zhou, and Aoying Zhou. Exploiting shopping and reviewing behavior to re-score online evaluations. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 649–650, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1230-1. doi: 10.1145/2187980.2188171. URL `http://doi.acm.org/10.1145/2187980.2188171`.

Ying Zhao and Justin Zobel. Effective and scalable authorship attribution using function words. In *Information Retrieval Technology*, pages 174–189. Springer, 2005.

Lina Zhou and Dongsong Zhang. toma linguistic footprints: Automatic deception detection in online communication. *Communications of the ACM*, 51(9):119–122, 2008.

Xiaojin Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2:3, 2006.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. In *Technical Report CMU-CALD-02-107*. CarnegieMellon University, 2002.