

# **From Language to the Real World: Entity-Driven Text Analytics**

**Boyi Xie**

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

**COLUMBIA UNIVERSITY**

2015

©2015

Boyi Xie

All Rights Reserved

# **ABSTRACT**

## **From Language to the Real World: Entity-Driven Text Analytics**

**Boyi Xie**

This study focuses on the modeling of the underlying structured semantic information in natural language text to predict real world phenomena. The thesis of this work is that a general and uniform representation of linguistic information that combines multiple levels, such as semantic frames and roles, syntactic dependency structure, lexical items and their sentiment values, can support challenging classification tasks for NLP problems. The hypothesis behind this work is that it is possible to generate a document representation using more complex data structures, such as trees and graphs, to distinguish the depicted scenarios and semantic roles of the entity mentions in text, which can facilitate text mining tasks by exploiting the deeper semantic information. The testbed for the document representation is entity-driven text analytics, a recent area of active research where large collection of documents are analyzed to study and make predictions about real world outcomes of the entity mentions in text, with the hypothesis that the prediction will be more successful if the representation can capture not only the actual words and grammatical structures but also the underlying semantic generalizations encoded in frame semantics, and the dependency relations among frames and words.

The main contribution of this study includes the demonstration of the benefits of frame semantic features and how to use them in document representation. Novel tree and graph structured representations are proposed to model mentioned entities by incorporating different levels of linguistic information, such as lexical items, syntactic dependencies, and semantic frames and roles. For machine learning on graphs, we proposed a Node Edge Weighting graph kernel that allows a recursive computation on the substructures of graphs, which explores an exponential number of subgraphs for fine-grained feature engineering. We demonstrate the effectiveness of our model to predict price movement of companies in different market sectors solely based on financial news. Based on a comprehensive comparison between different structures of document representation and their cor-

responding learning methods, e.g. vector, tree and graph space model, we found that the application of a rich semantic feature learning on trees and graphs can lead to high prediction accuracy and interpretable features for problem understanding.

Two key questions motivate this study: (1) Can semantic parsing based on frame semantics, a lexical conceptual representation that captures underlying semantic similarities (scenarios) across different forms, be exploited for prediction tasks where information is derived from large scale document collections? (2) Given alternative data structures to represent the underlying meaning captured in frame semantics, which data structure will be most effective? To address (1), sentences that have dependency parses and frame semantic parses, and specialized lexicons that incorporate aspects of sentiment in words, will be used to generate representations that include individual lexical items, sentiment of lexical items, semantic frames and roles, syntactic dependency information and other structural relations among words and phrases within the sentence. To address (2), we incorporate the information derived from semantic frame parsing, dependency parsing, and specialized lexicons into vector space, tree space and graph space representations, and kernel methods for the corresponding data structures are used for SVM (support vector machine) learning to compare their predictive power.

A vector space model beyond bag-of-words is first presented. It is based on a combination of semantic frame attributes,  $n$ -gram lexical items, and part-of-speech specific words weighted by a psycholinguistic dictionary. The second model encompasses a semantic tree representation that encodes the relations among semantic frame features and, in particular, the roles of the entity mentions in text. It depends on tree kernel functions for machine learning. The third is a semantic graph model that provides a concise and convenient representation of linguistic semantic information. It subsumes the vector space model and the semantic tree model by using a graph data structure for a unified representation for semantic frames, lexical items, and syntactic dependency relations derived from frame parses and dependency parses of sentences.

The general goal of this study is to ground information derived from NLP techniques to textual datasets in real world observations, where natural language semantics is used as a means to learn the semantic relations that are important in the domain, to understand what is relevant for objectives of interest of the practitioner. Experiments are conducted in a financial domain to investigate whether our computational linguistic methodologies applied to large-scale analysis of financial news can

improve the understanding of a company's fundamental market value, and whether linguistic information derived from news produces a consistent enough result to benefit more comprehensive financial models. Stock price data is aligned with news articles. Two kinds of labels are assigned: the existence of a price change and the direction of change. The *change* in price and *polarity* tasks are formulated as binary classification problems and bipartite ranking problems. Using the bag-of-words model and the proposed vector-space-model as benchmarks, the experiments show a significant improvement from the use of the semantic tree model. The semantic graph model with more expressive power outperforms both the vector space model and the tree space model. At best, there may be a weak predictive effect of news on price for a particular data instance, which is, for example, a company on a date, due to the fluctuation in uncertainty of financial market and the efficient market hypothesis. However, the proposed models and their outputs can provide useful information to guide financial market price prediction and to help business analysts discover potential investment opportunities. These advantages come from the rich expressive power of the semantic tree model and the semantic graph space model, since the models are able to learn the semantic relations that are important in the problem domain, and effectively discover the useful underlying structured semantic information from large-scale textual data.

# Table of Contents

<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>viii</b>
<b>I Background</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Language and the Real World . . . . .	2
1.2 A Motivating Example . . . . .	3
1.3 Main Contributions . . . . .	10
1.4 Organization of the Thesis . . . . .	11
<b>2 Literature Review</b>	<b>13</b>
2.1 Entity-Driven Text Analytics . . . . .	13
2.2 Financial News Analytics . . . . .	15
2.3 Structured Document Representation and Learning . . . . .	16
<b>3 Modeling Tools and Learning Methods</b>	<b>19</b>
3.1 Dependency Parsing . . . . .	19
3.2 Frame Semantics . . . . .	20
3.3 Dictionary of Affect in Language . . . . .	21
3.4 Named Entity Recognition . . . . .	21
3.5 Kernels and Support Vector Machines . . . . .	22

<b>II</b>	<b>Models to Discover the Structured Semantics in Text</b>	<b>25</b>
<b>4</b>	<b>Vector Space</b>	<b>27</b>
4.1	Motivation . . . . .	27
4.2	Lexical Features in Vector Space . . . . .	28
4.3	Semantic Frames in Vector Space . . . . .	29
4.4	Affects in Vector Space . . . . .	29
4.5	Putting It All Together for Vector Space Representation . . . . .	30
<b>5</b>	<b>Tree Space</b>	<b>32</b>
5.1	Motivation . . . . .	32
5.2	Constructing Semantic Tree Representation . . . . .	33
5.3	Tree Kernels to Measure Semantic Tree Similarity . . . . .	36
5.4	Tree Kernel on SemTree . . . . .	37
<b>6</b>	<b>Graph Space</b>	<b>40</b>
6.1	Motivation . . . . .	40
6.2	Constructing Semantic Graph Representation . . . . .	42
6.3	Variations of Semantic Graph Towards OmniGraph . . . . .	48
6.4	Graph Kernels to Measure Graph Similarity . . . . .	51
6.5	WL Graph Kernel Computation Example . . . . .	54
6.6	Node Edge Weighting Graph Kernel . . . . .	56
<b>III</b>	<b>Experiments</b>	<b>62</b>
<b>7</b>	<b>Financial News Analytics</b>	<b>64</b>
7.1	Background . . . . .	65
7.2	Corpus, Data Instances, and Labeling Methods . . . . .	66
7.3	Overall Experiments . . . . .	68
7.4	Vector Space Results and Discussion . . . . .	75
7.5	Tree Space Results and Discussion . . . . .	76
7.6	Graph Space Results and Discussion . . . . .	83

7.7	Company Mention Detection . . . . .	87
<b>8</b>	<b>GoodFor/BadFor Corpus Analytics</b>	<b>102</b>
8.1	Introduction . . . . .	102
8.2	Benefactive/Malefactive Identification Task . . . . .	103
8.3	Writer Attitude Detection Task . . . . .	106
8.4	Experiments and Results . . . . .	109
<b>IV</b>	<b>Conclusions</b>	<b>120</b>
<b>9</b>	<b>Conclusions</b>	<b>121</b>
<b>V</b>	<b>Bibliography</b>	<b>125</b>
	<b>Bibliography</b>	<b>126</b>



# List of Figures

1.1	Example Reuters news articles where <i>Google</i> is mentioned. The left news item reports <i>Nokia</i> found smartphone bugs on its latest Lumia phone, and <i>Google</i> is one of its competitors in the smart phone market. This news has a positive impact on the stock price of <i>Google</i> . The right news reports <i>Oracle</i> started a trial against <i>Google</i> due to <i>Google</i> 's Android operating system that tramples <i>Oracle</i> 's intellectual property rights to the Java programming language, which has a negative impact on <i>Google</i> 's stock price. . . . .	4
1.2	Summary of financial news items, e.g. as in Figure 1.1, pertaining to <i>Google</i> in April, 2012. Boxes mark up the three company mentions. The targeted company entity is in boldface. The underlined words evoke semantic frames. Our motivation is that entity-driven text analytics on news can predict the price movement of the targeted company, and exploiting such textual information with the help of semantic frames can identify information that predicts the price change event. . . . .	5
1.3	Desired features that capture the meaning from the two example sentences for the designated entity <i>Google</i> . . . . .	7
4.1	Example sentence and its frame semantic parse. . . . .	28
5.1	Constructing the semantic tree for the designated entity <i>Oracle</i> in sentence of Figure 4.1. . . . .	34
5.2	Constructing the semantic tree for the designated entity <i>Google</i> in sentence of Figure 4.1. . . . .	35

5.3	SemTree representation for the designated entity <i>Oracle</i> in sentence: <i>Oracle has accused Google of violating its intellectual property rights to the Java programming language.</i> . . . . .	36
5.4	Subset tree kernel for $k(T3, T1)$ and $k(T3, T2)$ . . . . .	38
5.5	Subset tree kernel for $k(T3, T1)$ and $k(T3, T2)$ . . . . .	39
5.6	When using SemTree representation and subset tree (SST) tree kernel, (a) (b) (c) are common tree fragments when comparing instance 3 to instance 1 ( $K(T1, T3) = 3$ ), while (c) is the only common tree fragments when comparing instance 3 to instance 2 ( $K(T2, T3) = 1$ ), as shown in Figure 5.5. . . . .	39
6.1	Example sentence, the frame semantic parse, and the dependency parse. . . . .	42
6.2	Semantic frames that are evoked for the sentence of Figure 4.1. Unlike SemTree where only the frames with designated entity are used, semantic graph representation make use of all frames. . . . .	43
6.3	Eight variants of graph representation for Oracle of sentence 1 . . . . .	44
6.4	Eight variants of graph representation for Oracle of sentence 1 . . . . .	45
6.5	Eight variants of graph representation for Oracle of sentence 1 . . . . .	46
6.6	Eight variants of graph representation for Oracle of sentence 1 . . . . .	47
6.7	Example sentence, the dependency parse, and the frame semantic parse. The red edges in the dependency parse helps recover the interactions among frames. . . . .	50
6.8	OmniGraph representation that includes lexical, dependency, and semantic information for <i>Humana</i> of the sample sentence in Figure 6.7. . . . .	51
6.9	Toy example of the WL graph kernel . . . . .	52
6.10	A subtree pattern of height 3 rooted at the node of <i>Designated Entity</i> , and the unfolding of this subtree pattern. The dashed area is equivalent to the semantic tree of Figure 5.2c. . . . .	55
6.11	Procedure of the computation for Weisfeiler-Lehman graph kernel with $h=1$ between instance 1 ( $G_1$ ) and instance 2 ( $G_2$ ) of Table 5.1. . . . .	57
6.12	Procedure of the computation for Weisfeiler-Lehman graph kernel with $h=1$ for instance 3 of Table 5.1 ( $G_3$ ). (a) Assign initial labels; (b) After iteration 0; (c) Sorted and prefixed multiset-label; (d) Label compression; and (e) After iteration 1. . . . .	58

6.13	Subgraph features up to 2 degree of neighbors that are explored by Node Edge Weighting graph kernel. . . . .	59
6.14	Toy example of the node edge weighting (NEW) graph kernel . . . . .	60
7.1	Pipeline of our experiments on Reuters news data. . . . .	67
7.2	Parametrizing OmniGraph <sup>NEW</sup> for companies in Consumer Staples sector. It shows a) the breakdown by stepsize for each of the 26 companies, and b) the total proportion across companies of node-edge weights for each feature type. . . . .	71
7.3	Sample OmniGraph features (OG) that have predictive power within or across sectors, compared with those from vector space (VS), from dependency trees (DT), and from SemTree (ST). . . . .	73
7.4	Best performing SemTree fragments for increase (+) and decrease (-) of price for consumer staples sector across training years. . . . .	79
7.5	ROC curves for the polarity task. . . . .	80
7.6	Ratio of feature types at top 100 and top 1000 ranked list by information gain for 2010 polarity prediction. . . . .	81
7.7	Example company and news sentences. . . . .	88
7.8	Framework of the text mining on financial news for stock market price prediction. . . . .	91
8.1	Example sentence, its benefactive/malefactive annotation, and the frame semantic parse. . . . .	105
8.2	SemTree representation for the object <i>a National Institute for Health and Clinical Effectiveness</i> of the sample sentence in Figure 8.1. . . . .	105
8.3	The dependencies among semantic frames, which is constructed based on syntactic dependency parsing. . . . .	106
8.4	OmniGraph representation that includes lexical, dependency, and semantic information for the object <i>a National Institute for Health and Clinical Effectiveness</i> of the sample sentence in Figure 8.1. . . . .	106
8.5	Example sentence and its writer attitude annotation. . . . .	107
8.6	SemTree representations for the agent and the object, respectively. . . . .	107
8.7	OmniGraph representations for the agent and the object, respectively. . . . .	108

8.8	Distribution of semantic frames that are identified in the GoodFor/BadFor dataset. The trendline is a log fit, with $R^2 = 0.856$ .	110
8.9	Number of the top 100 ranked features requiring each feature type for the Benefec- tive/Malefactive task.	116
8.10	Number of the top 100 ranked features requiring each feature type for the Writer Attitude task.	116
8.11	Graph features that predicts a positive polarity for the Object Entity in the Benefac- tive/Malefactive task.	117
8.12	Graph features that predicts a positive polarity for the Object Entity in the Benefac- tive/Malefactive task.	118
8.13	Graph features that predicts a negative polarity on the Designated Entity in the Writer Attitude task.	119

# List of Tables

4.1	FWD features ( <b>F</b> rame, bag-of- <b>W</b> ords, part-of-speech <b>D</b> AL score) and their value types. . . . .	31
5.1	Sample sentences with designated entities. . . . .	32
7.1	Description of news data. . . . .	66
7.2	Mean accuracy by sector for the majority class baseline, three benchmarks, and two graph kernel learnings on OmniGraph. The cases where the sector mean is significantly better than the baseline are marked by *. OmniGraph is significantly better than all three benchmarks in all cases. . . . .	69
7.3	FWD results for consumer staples sector for test year 2010. . . . .	77
7.4	Average MCC for the change and polarity tasks by feature representation, for 2008-2010; for 2011-2012; for all 5 years and associated p-values of ANOVAs for comparison to BOW. . . . .	80
7.5	Evaluation that concentrates on positive and negative predictions by Precision@TopK, DCG, MRR, and PNorm (lower is better). . . . .	81
7.6	A breakdown of performance by stepsizes ( $h$ ) using WL graph kernels for 4 variants of <i>SemGraph</i> . It shows the leave-one-out accuracies for some sample companies in the Energy sector. . . . .	85
7.7	The means and standard deviations of the leave-one-out accuracy over the companies in each of the eight GICS sectors. The performance of all variations of <i>OmniGraph</i> are shown. Boldface values are the best performance across different <i>OmniGraphs</i> . * indicates a p-value<.05 compared to baseline. . . . .	86

7.8	Description of news data for company mention detection. . . . .	90
7.9	A manual evaluation for company detection in a preliminary experiment. . . . .	95
7.10	Counts of company mentions by sentence. . . . .	95
7.11	Averaged test accuracy for each company by sector that uses 80% of the data for training 20% for testing. Boldface identifies a higher <i>CMD</i> mean and * identifies the <i>CMD</i> that is significantly better than the <i>Initial NER</i> with <i>p-value</i> < 0.05. . . . .	97
8.1	Top 50 most frequent frames in GoodFor/BadFor dataset. . . . .	111
8.2	Frame targets (lexical items that evoked the frames) for the top 10 most frequent frames (part 1). . . . .	112
8.3	Frame targets (lexical items that evoked the frames) for the top 10 most frequent frames (part 2). . . . .	113
8.4	Mean accuracy for Benefactive/Malefactive event and Writer Attitude tasks. . . . .	114

# Acknowledgments

First and foremost, I wish to express my deepest gratitude to my advisor, Rebecca Passonneau. I had the great good fortune to have an advisor who gave me the freedom to explore so many areas of the field at the beginning of my studies and under whose guidance I was able to direct my energies into research that shaped up to be a promising and thought provoking thesis topic. She has always been there to listen and advise. Under her tutelage I learned to question thoughts and express ideas and formulate hypotheses that were then written up. I am grateful for her careful reading and commentary regarding the countless revisions of this manuscript. Her patience and support helped me overcome many difficulties and finish this dissertation. This thesis would not exist if not for her insightful comments, constructive criticisms, and encouragement proffered at various stages of my research.

I wish to thank the members of my thesis committee: Kathleen McKeown introduced me to the exciting area of Natural Language Processing and helped me with many thoughtful comments and suggestions which contributed to the progress of my research work. Owen Rambow was a penetrating critic who asked probing questions while providing sound advice at the same time. Smaranda Muresan gave my thesis a close reading and provided invaluable feedback. Dragomir Radev provided perceptive criticism and inspiration when I needed it along with career guidance. Michael Collins taught me machine learning in NLP and served on my candidacy exam committee.

I had the good fortune to work with several talented and generous collaborators. Germán Creamer provided critical advice on preparing the framework for financial news analytics. Dingquan Wang helped with the research on kernel learning for ranking problems. Tifara Ramelson worked with me on research into Named Entity Recognition.

I was fortunate to participate in many research projects. I wish to express my profound gratitude to all those who helped me grow as a researcher and a person. I have fond memories of working on the power grid project with Axinia Radeva and Ashish Tomar. I will miss the many discus-

sions and meetings in which I learned so much. I am indebted to Cynthia Rudin who taught me ranking and optimization. I learned to mine Electronic Health Records while working with Ansaf Salieb-Aouissi. Haimonti Dutta afforded me the opportunity to learn the intricacies of machine learning while assisting her with research on her digital library project. Roger Anderson allowed me to work on his power grid project for a semester. Bob Carpenter tutored me in probabilistic models. Nizar Habash and Mona Diab inspired me with their passion for the subject matter, high intellectual standards and cutting edge research. David Waltz, Albert Boulanger, Ashish Gagneja, Hatim Diab, Manoj Pooleery, Arfath Pasha, Ramy Eskander, Yassine Benajiba, Daniel Alicea, Idrija Ibrahimagic, Derrick Lim, Kathy Hickey, and many others inspired and encouraged me and made the research lab feel like a second home.

In the intellectually stimulating environment of Columbia University I was fortunate to study and grow along with some brilliant fellow students. Special thanks go out to Apoorv Agarwal. Our discussions and collaboration allowed me to build a foundation in the field of structured document representation. Ilia Vovsha taught me much about machine learning, support vector machines, and we share the joy of teaching. To Ahmed El Kholy, Mohammed Altantawy, Daniel Bauer, Vinodkumar Prabhakaran, Heba Elfardy, Noura Farra, Faiza Khan Khattak, Mohammed Sadegh Rasooli, Wael Salloum, Hooshmand Shokri Razaghi, Sara Rosenthal, Sara Alkuhlani, Joshua Gordon, David Elson, Or Brian, Kapil Thadani, Wei-Yun Ma, Yin-Wen Chang, Karl Stratos, Jessica Ouyang, Bob Coyne, Christopher Kedzie, Shen Wang, Wei Liu, Hao Dang, and Changyin Zhou, among many. My fellow townsman Weiwei Guo was my buddy and lab mate. Working at his side provided me with many wonderful memories. Leon Wu helped me to adjust to life in New York and provided priceless advice concerning both life and career issues. Special thanks to William Hurt whose interest in the future of Artificial Intelligence inspired and motivated me.

To my friends with whom I crossed the Pacific to the United States: we shared both a great passion for learning and wonderful experiences in New York. Thanks to: Fengwei Zhang, Shanghao Li, Yi Wang, Cheng Cheng, Maoliang Huang, Junxiong Jia, Bai Xiao, Sinan Xiao, Hai Wang, Yuan Yuan, Fan Lin, Keng He, Bing Liu, Zhemin Zhang, Xintong Zhou, Xin Wang, Xiangrong Kong, Jia Li, Shih-hao Liao, Jun Hu, Peng Liu, Hao Li, Qi Li, Xiaorui Sun, Wei Xu, Jocelyn Lu, Lina Lu, Li An, Shimeng Sun, Juan Li, Liwei Wang, Xuwei Yang, and many that I cannot list them all.

None of my accomplishments would have been possible without the unconditional love and



support of my family. I thank my parents Yaoguang Xie and He Gu for setting me an example of integrity and hard work by encouraging my youthful dreams and for motivating me to explore the world around me and reminding me that whatever I do and wherever I go there will always be a place for me at home.

For the past 9 years I have shared my joys and sorrows with my wife Sharrie (Yunchun) Xu. Though separated by the wide Pacific, and once the Atlantic Ocean, her love, support and encouragement kept a smile on my face, my spirits up, and allowed me to face the challenges life set before me. One person walks fast; two people walk further. With her I start my next chapter.

To my parents Yaoguang Xie and He Gu, and my wife Sharrie.

## **Part I**

# **Background**

# Chapter 1

## Introduction

### 1.1 Language and the Real World

The relationship between language and the real world, and consequently our ability to use words to refer to entities in the world, provides a foundation for linguistic communication. Few current natural language processing systems are designed to make direct predictions on real world outcomes by semantic information discovery from natural language text. However, by linking text to real world outcomes of entity mentions there are many opportunities for NLP research to mine. Example problems include a financial application that predicts the situation of a company involved in a lawsuit from the ways of narrating evidence stories in news reports, a survey tool that understands the public's opinions of an ongoing government act from the orientation of words, and a healthcare system that predicts a patient's illness by the word choice of symptom descriptions in doctor's notes. Facilitating these predictions through natural language understanding motivates this research.

Although document level analysis of various kinds, such as topical categorization of news and sentiment analysis of social media, is well-studied, there is still a need for a fine-grained approach that uses semantic and relational information to analyze the entities mentioned in documents. Depending on the domain, certain entities can be designated for different purposes of study. For example, in the political domain, a designated entity can be a candidate in an election or a pending congressional bill [Gerber *et al.*, 2009]. In product reviews, a designated entity can be a product (e.g. *I like the design of iPod video*) [Scheible and Schütze, 2013]. Designated entities can also be social issues, government acts, new events, or opinions (e.g., *Shiite leaders accused Sunnis of*

*a mass killing of Shiites in Madaen, south of Baghdad*) [O'Connor *et al.*, 2013]. In the financial domain, a designated entity can be a company in the stock market, which is of interest to investors and traders (e.g., *Facebook bought WhatsApp for \$19 billion; Oracle sued Google over Android*) [Feldman *et al.*, 2011a]. These are all examples of entity-driven text analysis.

In entity-driven text analysis, where we specify the designated entity of interest, we can use the information and signals from the real world to label entity mentions and make predictions. Although real world events are objective in one sense, they are often placed on a scale that is at least partly subjective or relative. We can quantify the impact of the information in a document using the scale of the impact of the real world event. By analyzing the relationship between the real world phenomenon and the natural language text, a model can be built to connect the two. This model can in turn automate the process of information discovery and real world event outcome prediction.

In this study, documents and the entities to be modeled are encoded in linguistically enriched vector space, and novel tree and graph representations. These representations preserve a large amount of rich linguistic information, such as the entities and their semantic roles, semantic frames and the dependencies among frames, and lexical items. They use natural language semantics as a means to an end, where the end goal is to learn the semantic relations that are important in the domain, to understand what is relevant for entities of interest. Our focus is on how to construct the essential picture of an entity from textual sources in order to predict the outcomes of real world entities.


## 1.2 A Motivating Example

Words are the presentation of knowledge and information. However, the ability of humans to interpret words, to generalize similarity of meanings, and to interpret text goes far beyond understanding the individual words. Such understanding requires the ability to analyze the structures and meanings behind words, and how all linguistic features, i.e. lexical items, syntactic dependencies, and semantic information, cooperate to reflect the writer's purpose. What is more, given successful large-scale pattern recognition of deep semantic information, retrospective analysis of predictive features can provide insights into the entities of interest and the domain.

As mentioned in the previous section, we have seen the potential of entity-driven text analytics

### Nokia's U.S. ambitions hit by smartphone bug

Recommend 26 people recommend this. Be the first of your friends.



By Tarmo Virki  
HELSINKI | Wed Apr 11, 2012 5:58am EDT

**(Reuters) - Nokia has found a software bug in its Lumia 900 smartphone, its answer to Apple's iPhone, and is effectively giving the model away until it is fixed, blunting its bid to turn around its fortunes in the United States.**

Nokia's first 4G phone, which it markets with the strapline "an amazingly fast way to connect", can occasionally lose its data connection as a result of the bug, **Nokia** said.

Though still the world's biggest volume maker of cellphones, Nokia lost the top spot in the lucrative smartphone market last year to **Apple** and **Google**, in part due to its weak performance in the United States, where its smartphones have slipped to less than a 1 percent market share.

Twitter 65

Share

Share this

Email

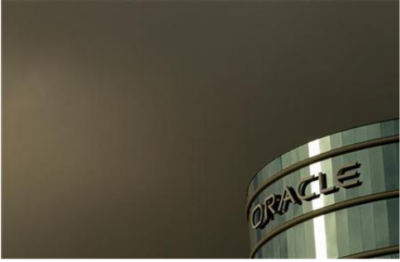
Print

**Related News**

- Nokia flagship smartphone has bug, setback to US ambitions Wed, Apr 11 2012
- Exclusive: China's ZTE planned U.S. computer sale to Iran Tue, Apr 10 2012
- Microsoft to struggle vs. Apple, Google in tablets Tue, Apr 10 2012
- Samsung hits a high Note, posts record quarterly profit Fri, Apr 6 2012
- Microsoft shuts German distribution center in patent dispute Mon, Apr 2 2012

### Oracle kicks off busy trial season against Google

Recommend 30 people recommend this. Be the first of your friends.



By Dan Levine  
SAN FRANCISCO | Fri Apr 13, 2012 7:36am EDT

**(Reuters) - Oracle Corp is set to go to trial next week against Google Inc in a high-stakes dispute over smartphone technology, the biggest case in what is shaping up to be an intense year in court for the enterprise software giant.**

Jury selection is set for Monday in San Francisco federal court. Oracle claims Google's Android operating system tramples on its intellectual property rights to the Java programming language. Google says it doesn't violate Oracle's patents, and that Oracle cannot copyright certain parts of Java.

Twitter 91

Share

Share this

Email

Print

**Related News**

- Google stock split helps Page, Brin maintain grip Thu, Apr 12 2012
- Facebook to buy Instagram for \$1 billion Mon, Apr 9 2012
- Microsoft trumps Amazon, others for AOL patents Mon, Apr 9 2012
- Viacom wins reversal in landmark YouTube case Thu, Apr 5 2012
- Accused September 11 mastermind to face trial at Guantanamo Wed, Apr 4 2012

Figure 1.1: Example Reuters news articles where *Google* is mentioned. The left news item reports *Nokia* found smartphone bugs on its latest Lumia phone, and *Google* is one of its competitors in the smart phone market. This news has a positive impact on the stock price of *Google*. The right news reports *Oracle* started a trial against *Google* due to *Google*'s Android operating system that tramples *Oracle*'s intellectual property rights to the Java programming language, which has a negative impact on *Google*'s stock price.

On Wednesday, April 11th, 2012, **Google Inc** announced its first quarterly earnings report, a week before the April 20 options contracts expiration in contrast to its history of reporting a day before monthly options expirations. In the news of the same day, **Google** is reviewed for its *aggressive* reposition to mobile gadgets and online social network service, *aggressive* hiring, and *swelling* cash coffers. Additionally, **Nokia** found smartphone bugs on its latest Lumia 900 that *benefits* **Google**'s Android phone market. The stock price of **Google** surged 3.85% from April 10th's \$626.86 to 12th's \$651.01. On Friday, April 13th, news reported **Oracle Corp** would sue **Google Inc**, claiming **Google**'s Android operating system tramples its intellectual property rights. Jury selection was set for the next Monday. **Google**'s stock price tumbled 4.06% on Friday, and continued to drop in the following week.

Figure 1.2: Summary of financial news items, e.g. as in Figure 1.1, pertaining to *Google* in April, 2012. Boxes mark up the three company mentions. The targeted company entity is in boldface. The underlined words evoke semantic frames. Our motivation is that entity-driven text analytics on news can predict the price movement of the targeted company, and exploiting such textual information with the help of semantic frames can identify information that predicts the price change event.

in a variety of domains, e.g. health care [Salleb-Aouissi *et al.*, 2011] and energy [Xie *et al.*, 2012], where text analytics that focus on very different domain entities (e.g. a medical symptom or a utility structure that come with text descriptions) are used to help knowledge discovery or real world forecast. A question is: what makes a good model for such knowledge discovery or real world forecast? Is it purely because of a high prediction accuracy, or is there something more?

A good model should be highly predictive, explainable, and also able to provide insights into the problem domain for the end user. Let's consider a scenario. A financial advisor's responsibility is to acquire and analyze information in the market, and provide investment recommendations to her client investors. One day, she, the financial advisor, puts together recent market events from newspapers for a portfolio. Based on her decades of experience in the financial industry, she told her client that she would change her recommendation grade for a company from *BUY* to *SELL*, because she expects the stock price for that company will go down in the following trading day. One of her clients has known her for a long time and has been following her advice to invest for many years, earning good money. He would just follow her advice without any questioning. There is another client who just switched to her because of her reputation in the industry. However, he is interested in more than just knowing the recommendation, so he started to question: How did you come up with this recommendation? What is the reason behind the prediction procedure? Is there a story or any market events that happened related to the company that helped make the decision? A good financial advisor should be able to answer these questions, explain the reasons behind the prediction, and inform the client by bringing insights to the problem domain. So should an NLP model.

Figure 1.2 shows a constructed example based on extracts from financial news about *Google* in April, 2012. It illustrates how a series of events reported in the news precedes and potentially predicts a large change in *Google*'s stock price. *Google*'s early announcement of quarterly earnings possibly presages trouble, and its stock price falls soon after reports of a legal action against *Google* by *Oracle*. To produce a coherent story, the original sentences were edited for Figure 1.2, but they are in the style of actual sentences from our dataset, e.g. Reuters news as shown in Figure 1.1. Accurate detection of events and relations that might have an impact on stock price should benefit from document representation that captures sentiment in lexical items (e.g., *drop*) combined with the conceptual relations captured by FrameNet [Ruppenhofer and Rehbein, 2012]. FrameNet is a



lexical conceptual representation that can be used to capture key relationships of who does what to whom, and can generalize the same role relations for different words, e.g. *stock surged* versus *stocks rose*, or *tumbled* and *drop* in the example in Figure 1.2. A frame in frame semantics [Fillmore, 1976] is a lexical semantic representation of the conceptual roles played by parts of a clause, and relates different lexical items (e.g., *report*, *announce*) to the same situation type. In Figure 1.2, some of the words that evoke frames have been underlined, and company entities of interest are in boldface.

Two key sentences in Figure 1.2 describe market events that drove *Google's* price up and down.

(1) Nokia found smartphone bugs on its latest Lumia 900 that benefits Google's Android phone market.

(2) Oracle Corp would sue Google Inc, claiming Google's Android operating system tramples its intellectual property rights.

There are many challenges to process the above text to model the outcome of the entity. These include (1) named entity recognition, e.g. company mention detection for Nokia, Google, and Oracle; (2) sentiment analysis, e.g. *benefits*; (3) influential events that are not expressed in sentiment words, e.g. the lawsuit scenario; (3) distinctions between the roles of entities, e.g. who is the beneficiary, who is the plaintiff and who is the defendant; (4) generalization of word meanings for different lexical items, e.g. *sue* and *accuse*. To address these challenges motivates this study, and we propose solutions based on models that take advantage of linguistic resources and NLP techniques to integrate lexical, syntactic, and semantic analysis as needed. A desired goal is to obtain a rich feature representation that preserves the meaning of the sentence. Machine learning applied to this representation can not only make predictions but also allow the users to interpret the model.



Figure 1.3: Desired features that capture the meaning from the two example sentences for the designated entity *Google*.

Figure 1.3 shows two graphs corresponding to the desired features that capture the meaning from the two example sentences. The feature on the left captures that Google is a beneficiary in sentence 1. The feature on the right captures that Google is mentioned in a statement and is the plaintiff in sentence 2. The questions of whether structured features such as these are useful for text forecasting, and how to come up with useful features, motivate our study of linguistically-rich, structured data representation for feature engineering.

Another aspect that motivates this study is the problem domain of financial analytics, which provides many opportunities for NLP research. Many news organizations that feature financial news, such as Reuters, the Wall Street Journal, and Bloomberg, devote significant resources to the analysis of corporate news. Much of the data that would support studies of a link between the news media and the market are publicly available. Among the debates of the predictability of financial news on stock prices, [Tetlock *et al.*, 2008] pointed out that linguistic communication is a potentially important source of information about firms' fundamental values. Because very few stock market investors directly observe firms' production activities, they get most of their information second-hand. Their three main sources are analysts' forecasts, quantifiable publicly disclosed accounting variables, and textual descriptions of firms' current and future profit-generating activities. If analyst and accounting variables are incomplete or biased measures of firms' fundamental values, linguistic variables may have supplementary explanatory power for firms' future earnings and returns [Tetlock *et al.*, 2008].

Our examples come from finance, a field driven by the acquisition of information to evaluate financial instruments. This study investigates the role of NLP to analyze financial news. We discuss how this NLP research can potentially benefit the existing quantitative-based financial models, however, we do not directly test whether our results can improve those financial models. Quantitative models in finance evaluate the underlying value of financial instruments, and are mainly based on numerical data. Such models include the popular Capital Asset Pricing Model (CAPM) [Sharpe and Sharpe, 1970] in portfolio management, and a more recent Fama-French model [Fama and French, 1993] that incorporates the explanation of market behavior, and the Copula Model [Li, 2000] for PD (probability of default) estimation in modeling credit risk. These quantitative models often fail to meet the desire from fundamental analysts to know what happened and why it happened, alongside quantitative analyses. In fundamental analysis, financial advisors, credit analysts, and traders need

to read hundreds of articles everyday to look for risks or investment opportunities. Therefore, a more explainable quantitative model that can tell the stories of market events would meet analysts' needs for information acquisition. Natural language processing that mines financial news provides a bridge, and can also benefit existing models. For example, the ADS model [Rydberg and Shephard, 2003] decomposes stock price analysis of financial data into three parts - activity (a binary process modeling the price move or not), direction (another binary process modeling the direction of the moves) and size (a number quantifying the size of the moves). Our work looks into two binary classification tasks for news - price change and polarity, which are analogous to the ADS model's activity and direction components. The Binomial Model [Cox *et al.*, 1979], a discrete numeric method for the famous Black-Scholes Model [Black and Scholes, 1973], estimates option pricing with the assumption that price changes over time are brownian motions. Implementations often rely on Monte Carlo simulation of price movement, based on the volatility estimated over a historical time frame. The outcome of our study that estimates the impact of news for price change at each timestamp can potentially be used to improve the Binomial Model. For example, the Binomial Model uses a single probability of price change (e.g. up or down) to simulate the price movement over time for option pricing, whereas use of text forecasting based on news could potentially provide more accurate probabilities of whether the price is going up or down for each moment when a news item is released.

There has been a debate on whether conventional news can be used to predict changes in the stock market. This work contributes to this question. In the efficient market hypothesis, all available information is incorporated in price, and investors cannot make excess profits from the market. However, earlier works have shown that stock prices appear to under-react to the market [Chan, 2003], and that there is a one-day delay of the price to react to the news [Tetlock, 2007]. We carry out a formal study that applies natural language processing techniques that rely on semantic features to model news articles, and predict changes in stock price of specific companies. Price information of the companies mentioned is used to label the financial news data. Our experiments test several document representations for document classification and ranking tasks. Our results show that the market may not be that 'efficient', and price has not fully incorporated all information, but can be partially predicted by news.

Our study not only shows that price movements can be predicted more accurately by leveraging

rich linguistic features, but also structured document representations based on lexical items, syntactic dependencies, and frame semantics can provide interpretable features, which is beneficial to investors for information discovery in the financial market.

### 1.3 Main Contributions

This study on text analytics exploits rich linguistic features. We hypothesize that rich structured representation of text is crucial for entity-driven text analytics, where it is important to detect what situations the entity participates in, and in what role. *BOW* is still in most general use and has been highly effective for tasks such as classification of financial documents [Purda and Skillicorn, 2014; Wintrode and Khudanpur, 2014]. This work aligns with much recent work on representation for text classification that builds on linguistically informed features, such as syntactic and semantic information. For example, [Sayeed *et al.*, 2012] use syntactic structures for word-level sentiment detection. [Kim and Hovy, 2006] use semantic role labeling to label opinion holders and topics to mine online news. These works have shown promising results of the incorporation of richer linguistic information into feature space, e.g. syntactic dependencies or semantic frames. However, no fundamental paradigm is proposed for a coherent and extensible way to combine all this linguistic information for text, and to learn the structured and relational information from the uniform representation, as we do here. We study a variety of representation schemas, including vector, tree, and graph structures, and propose a principled document representation that can include different levels of linguistic information, such as lexical, syntactic, and semantic features, in a uniform way. Our experiments address a challenging real world text forecasting problem that predicts price movement of individual companies in the stock market. We also test the representation and learning method on a fine-grained sentiment analysis corpus that predicts object benefits and writer attitudes. We find that our proposed structured representation and learning method achieve significant improvements over the baselines and traditional text modeling methods.

As an overview, the main contributions of this thesis include:

1. A demonstration of the benefits of frame semantic features and how to use them in document representation. Results show that frame semantic features are very useful and predictive for polarity detection and sentiment analysis in the financial domain.

2. A novel tree structured representation to model entity driven text analytics problems, where the root of the tree is the entity to be modeled and the other nodes along the trees are its semantic roles and semantic frame features. The semantic tree representation is used with a vector space of lexical items and part-of-speech-specific psycholinguistic dictionary-based features for machine learning.

3. A rich and principled graph-based document representation of mentioned entities that incorporates different levels of linguistic information, such as lexical items, syntactic dependencies, and semantic frames and roles. We proposed a new graph kernel learning for use with our graph document representation. Like some other graph kernels, it allows a recursive computation on the substructures of graphs. Our kernel exploits different weightings on different node and edge feature types for fine-grained feature exploration.

4. A contribution to the novel area of text forecasting, where text is linked to designated entities, and natural language learning on text is used to predict the outcome of real world entities. We demonstrate the effectiveness of NLP to predict price movement of companies in different market sectors solely based on financial news, which is a very challenging task.

5. Demonstration that our linguistically-rich graph structured representation with graph kernel learning outperforms several baselines on a new fine-grained sentiment analysis dataset - the Good-For/BadFor corpus in MPQA (multi-perspective question answering), where two polarity classification tasks are involved.

6. A comprehensive comparison between different structures of documentation representation and their corresponding learning methods, e.g. vector, tree, and graph space models. We found that the application of a rich semantic feature learning can lead to interpretable features from trees and graphs.

7. Experiments on coreference resolution on company mentions in financial news that shows coreference does not benefit the structured representation for the price prediction task.

## 1.4 Organization of the Thesis

The thesis is organized as follows. Chapter 1 introduces the study background of entity driven text analytics, and provide a motivating example. Chapter 2 discusses the related work on entity driven

text analytics, financial news analytics, and structured document representation and learning. Chapter 3 introduces the linguistic resources and related NLP tools that our models build on, including syntactic dependency parsing, frame semantics and semantic parsing, DAL - a linguistic resource that provides valence score, named entity recognition, and kernels based machine learning. Chapters 4, 5, 6 introduce our main methods. They separately describe our data representation in vector, tree, and graph space, and their corresponding learning methods for different representations. Chapter 7 describes the results and discussions in financial news analytics where Reuters news are used to predict the price change of companies in different market sectors. Chapter 8 presents the experiments on a new fine-grained sentiment analysis dataset - the GoodFor/BadFor corpus. Chapter 9 concludes the thesis.

## Chapter 2

# Literature Review

### 2.1 Entity-Driven Text Analytics

Entity-driven text analytics, where large collections of documents are analyzed to study the entity mentions in text, has recently become an active research topic in NLP. These studies usually associate the texts that describe entities of interest to the relations or outcomes of those entities in real world phenomenon. This analysis can facilitate better understanding of the entity mentions, or make predictions, as in the more specific area that has recently been called text forecasting. [O'Connor *et al.*, 2013] associate the textual data with the real world political information to learn international relations. The entities in their study are countries, and the textual data are newswire articles, e.g., *Pakistan prompt accused India*. Their data instance includes country entities at a timestamp. Their model consists of the temporal information and a tuple of two entities that form a relation or participate in an event. They use dependency structure information to distinguish the roles of entities mentioned in sentences. This work is similar to theirs in that we also focus on the analysis of news articles on a predefined set of entities. Rather than learning the relations between countries in the political domain, we learn the stock market performance of company entities in a financial domain. [Bamman *et al.*, 2013] analyze movie plot summaries to predict the personas for characters. They use rich linguistic features such as syntactic and semantic parses to capture the stereotypical actions of which character entities are the agent and patient, as well as attributes by which they are described. Our study also explores rich linguistic features, such as syntactic dependency relations and frame and semantic role information, but to make predictions about company entities. [Scheible and

Schütze, 2013] work on entity-oriented sentiment analysis to identify the relevance of sentiment to entities and the polarity of the sentiment. Their approach focuses on subjective text where adjectives or verbs reveal attitude. This study not only works on entity-driven sentiment analysis but also aims to model the impact of objective events and how they affect the perceived status of an entity in addition to sentiment. For example news of *mergers & acquisitions* can affect perception of the companies involved.

A key question for entity analysis is how to form data instances that are based on real world entities, what data structure to use to represent the data instances, how to label the data instances, and how to build models to perform learning. Much recent research relies on real world phenomena to automate the process of data instance labeling, in contrast to most of the traditional studies in NLP where human annotations are used. This research exploits the use of supervision from real world phenomenon on data instances about real world entities. It is related to work on distant learning or weakly supervised learning where existing knowledge bases are used as source of supervision and little human annotation is required for relation extraction. [Mintz *et al.*, 2009] in their recent work use distant learning to heuristically align the given knowledge base, Freebase, to text and rely on this alignment to learn a relation extractor. Their approach is based on the assumption that if two entities participate in a relation, all sentences that mention these two entities express that relation. In later studies, distant learning is improved to solve practical problems involved in real world datasets in order to tolerate noisy labels [Riedel *et al.*, 2010], to support multiple relations [Hoffmann *et al.*, 2011; Surdeanu *et al.*, 2012], and to estimate the probabilities of a pattern showing relations [Takamatsu *et al.*, 2012; Min *et al.*, 2013]. Although this study is not for relation extraction, it uses domain knowledge as a source of supervision to learn relations between the entities mentioned in text and real world outcomes on the entities, and models entities by structured features based on meaning.

The benefit of analyzing the entities in texts based on real world problems is that we can acquire more knowledge about the entities of interest through mining natural language text, and can further make predictions about the outcomes to the entity. An advantage is that we can use real world phenomena to label documents while traditional NLP problems usually require a significant amount of annotation. One of our prior works can be found in the study of [Salleb-Aouissi *et al.*, 2011] in a healthcare domain. Electronic Health Records (EHRs) are mined to help the study of *infant colic* -



a poorly-understood condition defined as persistent inconsolable crying in healthy babies between 2 weeks and 4 months of age, where the baby seems to be in great discomfort and is difficult to soothe. Another example of our prior work that connects language and real world phenomena lies in the energy domain. The study of [Xie *et al.*, 2012] works with Con Edison to assess the vulnerability of power grids to maintain the reliability of the secondary electrical grid in New York City. The goal of the project is to develop interpretable models to rank power grid structures (manholes and service boxes) with respect to their vulnerability to a serious event, such as fire or explosion.

Our work is closely related to text forecasting, which is an emerging collection of problems in which text documents are used to make predictions about measurable phenomena in the real world [Kogan *et al.*, 2009]. In [Smith, 2010], text forecasting is defined as a challenge for natural language processing and machine learning: *Given a body of text  $T$  pertinent to a social phenomenon, make a concrete prediction about a measurement  $M$  of that phenomenon, obtainable only in the future, that rivals the best known methods for forecasting  $M$ .* Our study aligns with text forecasting: we build NLP models that rely on financial news to predict the future price movement of the company mentions. However, our goal is not only to make predictions from text, but also to build models that allow flexible feature engineering to generate interpretable features, and to provide insights to the problem domain.

## 2.2 Financial News Analytics

The financial domain is an area with a lot of recent active research. A growing literature evaluates the financial effects of media on the market [Tetlock, 2007; Stromberg, 2004; Gentzkow, 2006; Gerber *et al.*, 2009; Gentzkow and Shapiro, 2010; Engelberg and Parsons, 2011]. Recent work has applied NLP techniques to various financial media (conventional news, tweets) to detect sentiment in conventional news [Devitt and Ahmad, 2007; Haider and Mehrotra, 2011] or message boards [Chua *et al.*, 2009], or to discriminate expert from non-expert investors in financial tweets [Bar-Haim *et al.*, 2011]. [Kogan *et al.*, 2009] analyze quarterly earning reports to predict stock return volatility and to predict whether a company will be delisted. [Luss and d’Aspremont, 2008] use text classification to model price movements of financial assets on a per-day basis. They tried to predict the direction of return and abnormal returns, defined as an absolute return greater than a predefined

threshold. [Schumaker *et al.*, 2012] propose a stock price prediction system based on financial news. They represent documents by boolean valued BOW and named entities. [Wang *et al.*, 2011] present a framework for mining the relationships between news articles and financial instruments using a rule-based expert system. [Schumaker *et al.*, 2012] treat stock price prediction as a sentiment analysis problem to distinguish positive and negative financial news. [Tetlock, 2007] and [Tetlock *et al.*, 2008] quantify pessimism of news using General Inquirer (GI), a content analysis program. [Feldman *et al.*, 2011b] apply sentiment analysis on financial news using rule-based information extraction and dictionary-based prior polarity scores. In this study, we work on the financial domain and our goal is to ground information derived from NLP techniques to financial news in real world stock market observations. Most of the active research, as described above, explores the financial instruments where mining news can be beneficial. However, none of these focuses on document representation with rich linguistic features, and they rarely go beyond a BOW model. A main focus of the study is on the development of data representation for entities in documents that can not only provide effective and efficient learning but also facilitate model interpretation.

### 2.3 Structured Document Representation and Learning

Two questions guide our study: 1) what linguistic features are necessary for text representation, e.g. words, topics, syntactic or semantic features, and 2) what data structures should be used to encode such linguistic information. The representation for document classification in most general use is vector-based bag-of-words (BOW) model, which has been highly effective [Forman, 2003]. It is difficult to surpass for many document classification tasks, but cannot capture relational information or semantic similarity. Methods to investigate semantic similarity that have proven successful for paraphrase detection [Deerwester *et al.*, 1990; Dolan *et al.*, 2004] include latent variable models that simultaneously capture the semantics of words and sentences, such as latent semantic analysis (LSA) [Deerwester *et al.*, 1990] or latent Dirichlet allocation (LDA) [Blei *et al.*, 2003]. However, our task differs from paraphrase detection, and our focus is beyond the semantic similarity measures based on word co-occurrence patterns.

Although a vector space model can contain rich features such as bag-of-words, bag-of-frames, and word affect based on a psycholinguistic dictionary, representing text as a set (bag), disregard-

ing ordering and syntax, cannot express relational or structural information such as subject-object relations or predicate argument structures. Tree representation has been used to encode relational and structural linguistic information. [Wilson *et al.*, 2009] construct phrase-level modification features using syntactic parses, such as whether a word modifies or is modified by a subjective lexicon. [Sayeed *et al.*, 2012] use grammatical structure by constructing a suffix-tree data structure to represent syntactic relationships between opinion targets and opinion-bearing words for word-level sentiment detection. Their use of grammatical structure can discriminate between parts of the structure that are relevant to target-opinion word relations and those that are not. [Kim and Hovy, 2006] exploit the semantic structure of a sentence, anchored to an opinion bearing verb or adjective. They use semantic role labeling as an intermediate step to label opinion holders and topics for opinion mining on online news. [Agarwal *et al.*, 2014] use features derived from semantic annotations based on FrameNet to construct semantic trees for social network extraction. [Agarwal *et al.*, 2011] use trees to represent short texts for sentiment analysis on Twitter data. They rely on tree kernels for machine learning. Tree kernels have been proven to be an effective way to automate the exploration of syntactic and semantic information in text. The tree kernel [Moschitti, 2006; Collins and Duffy, 2002] is a function of tree similarity based on common substructures (tree fragments). Fast algorithms for kernel computation run in linear time on average, either by dynamic programming [Collins and Duffy, 2002], or pre-sorting production rules before training [Moschitti, 2006]. This benefits the application of tree kernels on NLP problems to efficiently learn linguistic phenomena and patterns in sub-tree structures. However, there are topological constraints on trees, such as no allowance of cycles and the distinction between root and leaf nodes that hinder the development of additional feature engineering. This motivates a more general data representation.

Graphs provide a more general, flexible and efficient data structure for problems as diverse as prediction of toxicity based on molecular structure [Wale *et al.*, 2008], analysis of 3-D scenes in virtual environments [Fisher *et al.*, 2011], and social network analysis [Ediger *et al.*, 2010]. They have been used in many NLP tasks, such as polarity of words [Hassan and Radev, 2010], opinion bearing words and opinion targets [Sayeed *et al.*, 2012], coreference [Nicolae and Nicolae, 2006], and dependency parsing [McDonald *et al.*, 2005a]. The problem of how to construct a meaningful graph representation and how to measure the similarity of graphs is at the core of learning on graphs. Most of the studies on graphs are based on the assumption that data instances with similar structure

have similar outcomes. This motivates our study on graph-structured semantic information derived from texts. We have found little if any work that applies graphs to large-scale semantic analysis of documents. This work is to study the representation of rich linguistic information, including semantic frames and syntactic dependency for large-scale text mining.

Several studies have focused on the learning on graph structures, and different graph kernels have been defined in machine learning. Graph kernels can be categorized into three classes: graph kernels based on walks [Kashima *et al.*, 2003; Gärtner *et al.*, 2003] and paths [Borgwardt and Kriegel, 2005], graph kernels based on limited-size subgraphs [Horváth *et al.*, 2004; Shervashidze *et al.*, 2009], and graph kernels based on subtree-like patterns [Mahé and Vert, 2009]. Among them, Weisfeiler-Lehman graph kernel [Shervashidze *et al.*, 2011] based on subtree-like patterns, short for *WL graph kernel*, is the one of particular interest. It can effectively measure the similarity between graphs, and subsumes tree kernel learning on tree structures. WL graph kernel also has a lower computational complexity compared to other graph kernels. Its computation is based on the Weisfeiler-Lehman test of isomorphism [Weisfeiler and Lehman, 1968], which iteratively compares the similarity of graphs based on an incremental size of node neighborhoods. This study uses WL graph kernel for machine learning to extract and benefit from the relational structures in a semantic graph representation with rich linguistic information. WL graph kernel is efficient at neighborhood augmentation but often results in coarse-grained features, because all neighbors of a node are expanded all together during the augmentation, and without distinguishing the node or edge types. We develop a novel node edge weighting (NEW) graph kernel that iteratively augments each of the neighboring nodes, and weights nodes and edges based on their feature types. NEW graph kernel generates finer-grained features that allow partial match of graph substructures, and provides a complimentary approach to WL kernel.

## Chapter 3

# Modeling Tools and Learning Methods

This chapter describes a set of tools and learning methods used in this thesis. Some of them have been widely applied in natural language processing: frame semantics and FrameNet for meaning analysis, dependency parsing for syntactic analysis on sentences, support vector machines for machine learning, and convolution kernels that measure the similarity of structured data.

### 3.1 Dependency Parsing

Dependency parsing, or dependency-based syntactic parsing, is an approach to automatic syntactic analysis of natural language inspired by the theoretical linguistic tradition of dependency grammar. Dependency parsing's advantages can be attributed to 1) its transparent encoding of predicate-argument structure, 2) its ability to parse flexible or free word orders, and 3) its utility as an intermediate representation for semantic parsing.

The basic assumption underlying dependency grammar is the idea that syntactic structure essentially consists of words linked by binary, asymmetrical relations called dependency relations (or dependencies for short). A dependency relation holds between a syntactically subordinate word, called the dependent, and another word on which it depends, called the head.

The information encoded in a dependency structure representation is different from the information captured in a phrase structure representation of a syntactic parse. Phrase structure represents the grouping of words into phrases, classified by structural categories such as noun phrase (NP) and verb phrase (VP), while dependency structure represents head-dependent relations between words,

classified by functional categories such as subject (SBJ) and object (OBJ).

In our experiments, we use the MST dependency parser [McDonald *et al.*, 2005b] that implements the Eisner algorithm [Eisner, 1996]. It provides an efficient and robust performance on our dataset.

## 3.2 Frame Semantics

Frame semantics [Fillmore, 1976] aims for a conceptual representation that generalizes from words and phrases to abstract scenarios, or frames, that capture explicit and implicit meanings of sentences. The central idea of frame semantics is that word meanings must be described in relation to semantic frames - schematic representations of the conceptual structures and patterns of belief, practices, institutions, or images, that provide a foundation for meaningful interaction in a given speech community [Fillmore *et al.*, 2003].

FrameNet is a computational lexicography project that extracts information about the linked semantic and syntactic properties of English words from large electronic text corpora, using both manual and automatic procedures, and presents this information in a variety of web-based reports. FrameNet identifies and describes semantic frames, and analyzes the meanings of words by directly appealing to the frames that underlie their meanings.

The primary units of lexical analysis in FrameNet are the frame and the lexical unit [Cruse, 1986], defined as a pairing of a word with a sense, for example, the *hot* of temperature and the *hot* of taste experiences are two among the many lexical units associated with the adjective *hot*. Generally speaking, the separate senses of a word correspond to the different semantic frames that the word can participate in. When a word's sense is based on a particular frame, we say that the word evokes the frame: thus, the word *hot* is capable of evoking a temperature scale frame in some contexts and a particular taste experience frame in others. Interpreting a sentence containing this word requires assumptions about which frame is relevant in the given context.

Based on the theory of frame semantics, there has been much research on semantic frame-based semantic parsing. The parser used in this study is SEMAFOR<sup>1</sup> [Das and Smith, 2011; Das and Smith, 2012]. It is a statistical parser that uses a rule-based frame target identification,

---

<sup>1</sup> <http://www.ark.cs.cmu.edu/SEMAFOR>.

a semi-supervised model that expands the predicate lexicon of FrameNet for semantic frame classification, and a supervised model for argument identification. The algorithm takes into account various linguistic constraints, such as relationships between pairs of semantic roles, and has seen significantly better frame-semantic parsing performance on unobserved, out-of-domain lexical units. This parser achieved the state-of-the-art results on the SemEval 2007 benchmark dataset.

### 3.3 Dictionary of Affect in Language

The Dictionary of Affect in Language (DAL) [Whissel, 1989] is a psycholinguistic resource designed to quantify the nuances of emotional words. The dictionary grew out of a tradition of lexical-emotional research. After a revision in 1998, it was extended to 8,742 words that were annotated for three dimensions: Pleasantness (Pls), Activation (Act), and Imagery (Img). The documentation of DAL lists the mean and standard deviation of all words belonging to each of the three categories, and provides a standardized scoring method to illustrate the possible ways to use the resources. Earlier works introduced DAL to natural language processing, and it has been proven to be an effective resource for sentiment analysis. [Agarwal *et al.*, 2009] introduced part-of-speech specific DAL features for sentiment analysis. We follow their approach to create vector space features in DAL by averaging the pleasantness, activation, and imagery scores for all words, verb only, adjective only, and adverb only words.

### 3.4 Named Entity Recognition

Coreference resolution is the task of finding all expressions that refer to the same entity in a discourse. It is important for natural language understanding tasks like summarization, question answering, and information extraction [Lee *et al.*, 2013].

The long history of coreference resolution has shown that the use of highly precise lexical and syntactic features is crucial to high quality resolution [Ng and Cardie, 2002; Lappin and Leass, 1994; Poesio *et al.*, 2004; GuoDong and Jian, 2004; Bengtson and Roth, 2008; Haghighi and Klein, 2009]. Recent work has also shown the importance of global inference - performing coreference resolution jointly for several or all mentions in a document - rather than greedily disambiguating individual pairs of mentions [Morton, 2000; Luo *et al.*, 2004; Yang *et al.*, 2004; Culotta *et al.*, 2007;

Yang *et al.*, 2008].

We experiment with the coreference resolution system of Stanford CoreNLP [Manning *et al.*, 2014] for entity mention detection. This system implements the multi-pass sieve coreference resolution (or anaphora resolution) system described in [Lee *et al.*, 2011; Raghunathan *et al.*, 2010; Lee *et al.*, 2013; Recasens *et al.*, 2013]. Their system is a collection of deterministic coreference resolution models that incorporate lexical, syntactic, semantic, and discourse information. All these models use global document-level information by sharing mention attributes, such as gender and number, across mentions in the same cluster [Lee *et al.*, 2011]. In overall, the Stanford CoreNLP Coreference Resolution System consists of three main stages: mention detection, followed by coreference resolution, and post-processing. In the mention detection step, mentions are extracted and relevant information about mentions, e.g., gender and number, is prepared for the next step. The coreference resolution stage implements a system that performs entity-centric coreference, where all mentions that point to the same real-world entity are jointly modeled, in a rich feature space solely using simple, deterministic rules [Lee *et al.*, 2013]. Sieves in this stage are sorted from highest to lowest precision. For example, the first sieve (i.e., highest precision) requires an exact string match between a mention and its antecedent, whereas the last one (i.e., lowest precision) implements pronominal coreference resolution. Post-processing is performed to adjust our output to the task specific constraints, e.g., removing singletons [Lee *et al.*, 2011].

### 3.5 Kernels and Support Vector Machines

Support vector machines (SVMs) are hyperplane learning algorithms that 1) map the training data into a higher dimensional feature space, and 2) construct a separating hyperplane with maximum margin in that feature space. This yields a nonlinear decision boundary in input space. The key step is to use kernel functions as similarity measures. It is possible to compute the separating hyperplane without explicitly carrying out the map into the feature space. For a function  $\Phi$  that maps a data point  $x$  in feature space  $\mathcal{H}$ , and kernel  $k$ , we have:

$$\Phi : \mathcal{X} \rightarrow \mathcal{H} \tag{3.1}$$

$$x \mapsto \mathbf{x} := \Phi(x) \tag{3.2}$$



$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle \quad (3.3)$$

In practice, a separating hyperplane may not exist, e.g. when there is an overlap of data points of different classes. To allow examples that are not strictly separable, one introduces slack variables  $\xi$ . This results in a soft margin classifier obtained by minimizing the objective function

$$\text{minimize } \tau(\mathbf{w}, \xi) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i \quad (3.4)$$

subject to

$$\xi_i \geq 0 \text{ for all } i = 1, \dots, m \quad (3.5)$$

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \text{ for all } i = 1, \dots, m \quad (3.6)$$

where the constant  $C > 0$  determines the trade-off between margin maximization and training error minimization. Incorporating a kernel, and rewriting it in terms of Lagrange multipliers, leads to the problem of maximizing

$$\text{maximize } W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad (3.7)$$

subject to

$$0 \leq \alpha_i \leq C \text{ for all } i = 1, \dots, m, \text{ and } \sum_{i=1}^m \alpha_i y_i = 0 \quad (3.8)$$

where  $\alpha_i$  are the Lagrange multipliers.

The main contribution of this dissertation is to develop data representations and their corresponding kernel functions for natural language text using semantic information, which is related to Equations (3.1)-(3.3). In particular, embedding data into  $\mathcal{H}$  via the mapping  $\Phi$  provides us the freedom to choose a non-linear map  $\Phi$  that transforms the original data representation into one that is more suitable for a given set of problems. In many NLP tasks the input domain cannot be neatly formulated as a subset of  $\mathbb{R}^d$ . Instead, the objects being modeled are strings, trees or other discrete structures which require some mapping  $\Phi$  to convert them into feature space for convenient similarity measures [Collins and Duffy, 2001]. This study explores the possible mapping  $\Phi$  that extend vector space bag-of-words model to semantic tree space model, and an even more generalized semantic graph space model, together with psycholinguistic dictionary based models for semantic

orientation and semi-supervised topic models. These kernels on trees or graphs that involve a recursive calculation over the *parts* of the data structure are instances of *convolution kernels*, which were introduced by [Haussler, 1999] and [Watkins, 2000].

## **Part II**

# **Models to Discover the Structured Semantics in Text**

Although the focus of the study is not on semantic parsing, we explore different approaches to utilize frame semantic parses to construct rich-featured data representations and to perform feature engineering. We compare three approaches for the use of semantic frames. The first is a rich vector space model based on semantic frame attributes. The second uses a tree representation that encodes semantic frame features, and depends on tree kernels for learning. The third is a more general form based on graphs derived from semantic frames, and uses graph kernels for support vector learning.

## Chapter 4

# Vector Space

### 4.1 Motivation

Bag-of-words (BOW) has been proven to be efficient and effective and provides strong baselines for many NLP tasks such as text categorization, information retrieval, and sentiment analysis. The limited expressiveness of BOW, however, makes it difficult to identify the underlying scenarios in text by generalizing the meanings of words when the word forms are different (e.g. *sue* and *accuse* both indicate a judgment communication scenario), or distinguish the word senses for the same word form (e.g. *right* for correctness versus a legal entitlement). Our vector space model that incorporates features from FrameNet aims to generalize word meanings and provide rich semantics for document representation.

Consider the following sentences:

[ **Oracle** *Communicator* ] [ *sued* *Judgment\_communication* ] [ **Google** *Evaluee* ] [ *in August 2010* *Time* ], [ *saying* *Statement* ] [ **Google's Android mobile operating system infringes its copyrights and patents for the Java programming language** *Message* ]. (a)

[ **Oracle** *Communicator* ] *has* [ *accused* *Judgment\_communication* ] [ **Google** *Evaluee* ] *of* [ *violating its intellectual property rights to the Java programming language* *Reason* ]. (b)

[ **Oracle** *Communicator* ] *has* [ *blamed* *Judgment\_communication* ] [ **Google** *Evaluee* ] *and* [ *alleged* *Statement* ] *that* [ *the latter has committed copyright infringement related to Java programming language held by Oracle* *Message* ]. (c)

[ **Oracle's Ellison** *Speaker* ] [ *says* *Statement* ] [ *couldn't sway Google on Java* *Message* ]. (d)

Sentences *a*, *b* and *c* are semantically similar, but lexically rather distinct: the shared words are the company names and *Java* (*programming language*). Bag-of-Words (BOW) document representation is difficult to surpass for many document classification tasks, but cannot capture the degree of semantic similarity among these sentences. Methods that have proven successful for paraphrase detection [Deerwester *et al.*, 1990; Dolan *et al.*, 2004], as in the main clauses of *b* and *c*, include latent variable models that simultaneously capture the semantics of words and sentences, such as latent semantic analysis (LSA) or latent Dirichlet allocation (LDA). However, our task goes beyond paraphrase detection. The first three sentences all indicate an adversarial relation of *Oracle* to *Google* involving a negative judgement. It would be useful to capture the similarities among all three of these sentences, and to distinguish the role of each company (who is suing and who is being sued). Further, these three sentences potentially have a greater impact on market perception of *Google* in contrast to a sentence like *d*, that refers to the same conflict more indirectly, and whose main clause verb is *say*. We hypothesize that semantic frames can address these issues.

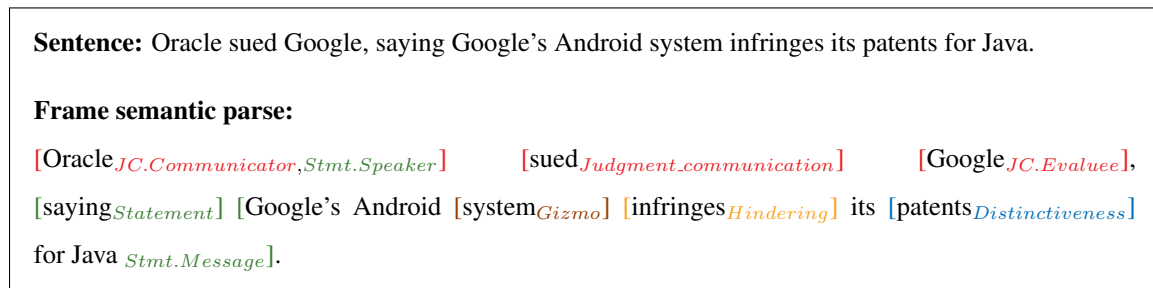


Figure 4.1: Example sentence and its frame semantic parse.

## 4.2 Lexical Features in Vector Space

Vector space representation is a typical representation for natural language processing and machine learning. Bag-of-words (BOW) representation is a typical vector space representation where a data instance is represented by a collection of words, and it has achieved high performance in many NLP tasks such as text categorization and sentiment analysis.

### 4.3 Semantic Frames in Vector Space

Here we use frame attributes instead of word tokens to construct the feature space. The basic components of frame semantics include frame names, frame targets (lexical units that evoke the frames), and frame elements (semantic roles). These three components can be used to represent documents. A data instance can thus be represented as a vector of a pre-defined length of  $M$ , and be mapped to a point in this  $M$ -dimensional space where each feature forms a dimension. As an analogy to *BOW*, we call it bag-of-frames (*BOF*) model.

Figure 4.1 shows a sentence with its semantic parse. Two frames, *Judgment\_communication* and *Statement*, are evoked by lexical units *sued* and *saying* respectively. For the *Judgment\_communication* frame, two elements are filled: *Communicator* and *Evaluee*. For the *Statement* frame, two elements are also filled: *Speaker* and *Message*. To construct a *BOF* vector space for this sentence, we have the following features: *FRAME\_NAME-Judgement\_communication*, *FRAME\_NAME-Statement*, *FRAME\_TARGET-Judgement\_communication-sue*, *FRAME\_TARGET-Statement-say*, *FRAME\_ELEMENT-Judgement\_communication-Communicator*, *FRAME\_ELEMENT-Judgement\_communication-Evaluee*, *FRAME\_ELEMENT-Statement-Speaker*, and *FRAME\_ELEMENT-Statement-Message*.

### 4.4 Affects in Vector Space

Semantic orientation of words has been proven effective in tasks such as sentiment analysis and opinion mining. These dictionary-based scoring features have been widely used in sentiment-based classification tasks. The dictionaries are usually developed by psycholinguists. The tasks often focus on classifying the semantic orientation of individual words or phrases or documents using linguistic heuristics, pre-selected sets of seed words, or human labeling. For example, General Inquirer (GI) is a dictionary-based content analysis program that quantifies texts by counting the words in a predetermined set of 77 categories, including 2 large valence categories - positive and negative, and some other categories such as pleasure and pain, strong and weak, active and passive. [Goyal and Daumé, 2011] used GI as a benchmark to evaluate the semantic orientation of words. [Mohammad and Turney, 2010] used GI to help create a high-quality, moderate-sized emotion lexicon using Mechanical Turk. [Tetlock, 2007] utilizes GI to quantitatively measure the interactions between the

media and stock market.

As described in Section 3.3, the Dictionary of Affect in Language (DAL) [Whissel, 1989] is a psycholinguistic resource designed to quantify the undertones of emotional words. It contains 8,742 words that were annotated for three dimensions: Pleasantness (Pls), Activation (Act), and Imagery (Img). Earlier works introduced DAL to natural language processing, and it has been proven to be an effective feature space for sentiment analysis. For example, [Agarwal *et al.*, 2009] introduced part-of-speech specific DAL features for sentiment analysis. We follow their approach to create vector space features in DAL by averaging the pleasantness, activation, and imagery scores for all words, verb only, adjective only, and adverb only words. This vector space representation in affects can be conveniently incorporated to *BOW* and *BOF* to form a linguistically rich representation.

## 4.5 Putting It All Together for Vector Space Representation

Table 4.1 lists 24 types of features, including semantic **F**rame attributes, bag-of-**W**ords, and scores for words in **D**AL by part of speech (part-of-speech DAL, or p**D**AL). We refer to these features as **FWD** features. Note that semantic frame attributes include frame names (**F**), frame targets (**FT**), and frame elements (**FE**).

Although boolean values are often used to indicate the presence of the features in vector space, weighted versions can also be used to scale the value of the frame attributes. For example, frequency and inverse-document-frequency are two of many possible weighting schemas. We define *idf*-adjusted weighted frame features, such as  $w^F$  for attribute  $F$  in document  $d$  as  $w_{F,d} = f(F, d) \times \log \frac{|D|}{|d \in D: F \in d|}$ , where  $f(F, d)$  is the frequency of frame  $F$  in  $d$ ,  $D$  is the whole document set and  $|\cdot|$  is the cardinality operator.



Category	Features	Value type
<b>Frame</b> attributes	F, FT, FE	$\mathbb{N}$
	wF, wFT, wFE	$\mathbb{R}_{\geq 0}$
<b>BoW</b>	UniG, BiG, TriG	$\mathbb{N}$
	wUniG, wBiG, wTriG	$\mathbb{R}_{\geq 0}$
<b>pDAL</b>	all-Pls, all-Act, all-Img	$\mathbb{R}_{\sim \mu=0, std=1}$
	VB-Pls, VB-Act, VB-Img	$\mathbb{R}_{\sim \mu=0, std=1}$
	JJ-Pls, JJ-Act, JJ-Img	$\mathbb{R}_{\sim \mu=0, std=1}$
	RB-Pls, RB-Act, RB-Img	$\mathbb{R}_{\sim \mu=0, std=1}$

Table 4.1: FWD features (**F**rame, bag-of-**W**ords, part-of-speech **DAL** score) and their value types.

## Chapter 5

# Tree Space

### 5.1 Motivation

	id	DE	Text	Label
train	1	Oracle	<b>Oracle</b> sued Google in August 2010, saying Google's Android mobile operating system infringes its copyrights and patents for the Java programming language.	+
	2	Google	Oracle <b>sued Google</b> in August 2010, saying Google's Android mobile operating system infringes its copyrights and patents for the Java programming language.	-
test	3	Oracle	<b>Oracle</b> has accused Google of violating its intellectual property rights to the Java programming language.	+
	4	Google	Oracle <b>has accused Google</b> of violating its intellectual property rights to the Java programming language.	-

Table 5.1: Sample sentences with designated entities.

In the previous section semantic frames are used as features in vector space, however, representing text as a set (bag), disregarding ordering and syntax, provides no relational or structural information, such as subject-object relations or predicate argument structure. For instance, consider the example data instances in Table 5.1. Assume instances 1 & 2 are training data and instances 3 & 4 are test data. Notice that instances 1 & 2 have the same text but the designated entities (in blue) are different, and the same for instances 3 & 4. It motivates us to design a data representation that can distinguish the different designated entities for the same text and capture the semantic

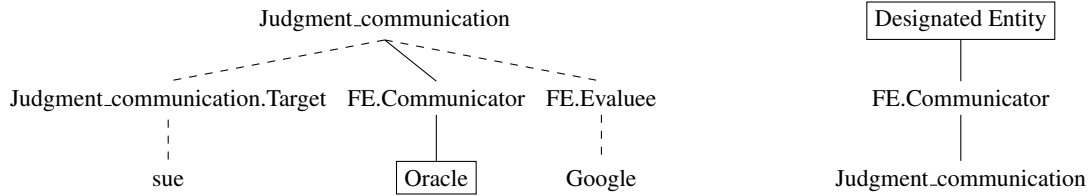
information about the designated entity. We also want a similarity measure (i.e. a type of kernel function) that can be applied to the data representation and achieve the desired similarity score. For example, to correctly classify the test data, we want a kernel function that measures instance 3 to instance 1 with a higher similarity score than to instance 2, and measures instance 4 to instance 2 with a higher similarity score than to instance 1. We hypothesize that the structural information in frame semantics can be better exploited using a tree data representation and a tree kernel that fulfills the similarity measure requirement.

## 5.2 Constructing Semantic Tree Representation

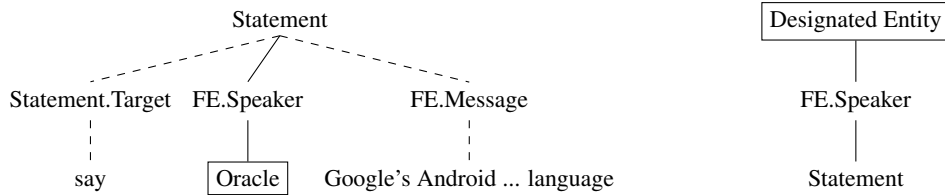
We developed a **semantic tree** (*SemTree*) data representation, where the frame semantic information is encoded in trees. *SemTree* aims at a general method to represent a named entity (e.g. a company) that the text (e.g. financial news) is about.

Consider the example in Figure 4.1, which is a sentence from a Reuters news article on April 17th, 2012 that describes a lawsuit in the Information Technology sector between the company *Oracle* and *Google*. The semantic frame parse of a sentence is a collection of frame structures. For example, for the two frames (*Judgment\_communication* and *Statement*) detected from the example sentence of Figure 4.1, each has a target and a number of frame elements. Each frame can be represented in the form of a tree, where the root is the frame name, one child is the target word, and the other children are frame elements, as shown in the left subfigures of Figure 5.1(a) and (b).

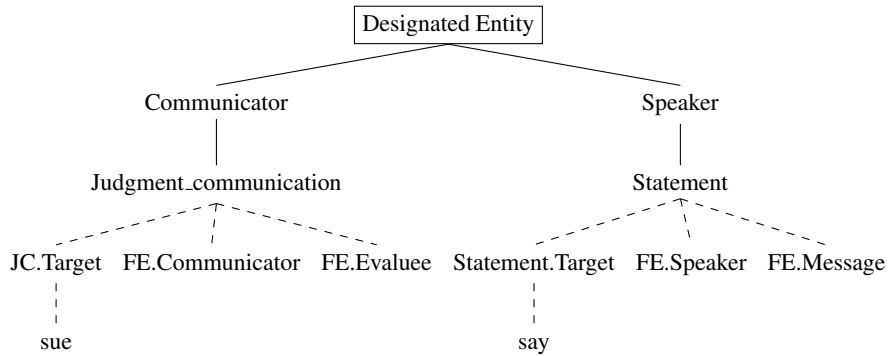
*SemTree* can be constructed to encode the original frame structure and its leaf words and phrases, and highlights a designated entity at a particular node as follows. For each frame evoked by a lexical item (target word), a *backbone* is found by extracting the path from the root to the role filler mentioning a designated entity via the semantic role (frame element) node; the backbone is then reversed to promote the designated entity to the root node. If multiple frames have been assigned to the same designated entity, their backbones are merged. Lastly, the frame elements and frame targets are inserted at the frame node. As will be described in section 7.2, a data instance corresponds to all the news associated to a company on a day. For this *SemTree* representation, the backbones of all frames with the same designated entity mentioned in news for a given day are merged at the root to create a single data instance.



(a) Extracting the backbone for *Judgment\_communication* frame for the designated entity *Oracle*.

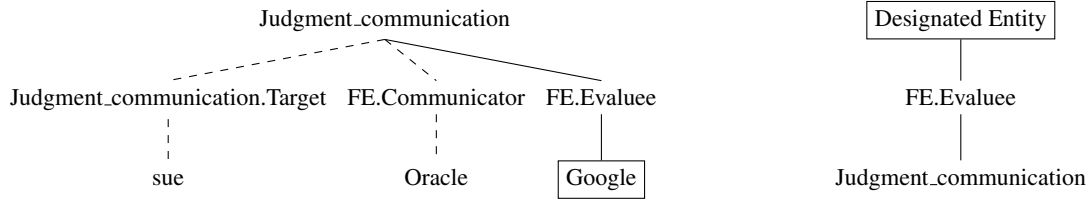


(b) Extracting the backbone for *Statement* frame for the designated entity *Oracle*.

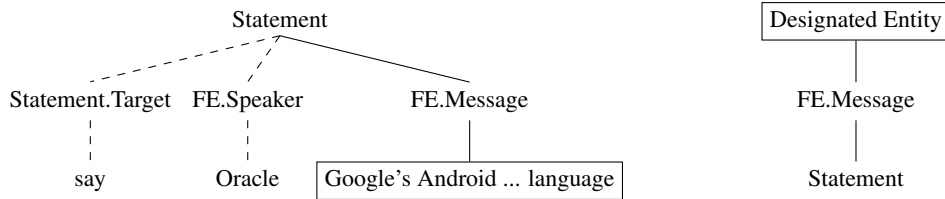


(c) Merging the backbones at the root for a tree representation for the designated entity *Oracle* in sentence of Figure 4.1.

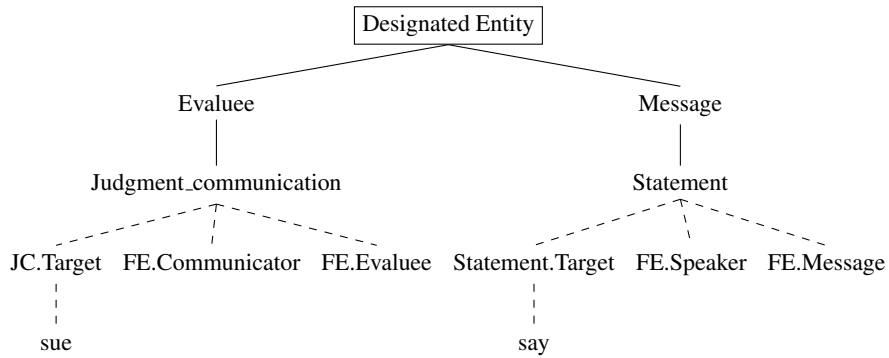
Figure 5.1: Constructing the semantic tree for the designated entity *Oracle* in sentence of Figure 4.1.



(a) Extracting the backbone for *Judgment\_communication* frame for the designated entity *Google*.



(b) Extracting the backbone for *Statement* frame for the designated entity *Google*.



(c) Merging the backbones at the root for a tree representation for the designated entity *Google* in sentence of Figure 4.1.

Figure 5.2: Constructing the semantic tree for the designated entity *Google* in sentence of Figure 4.1.

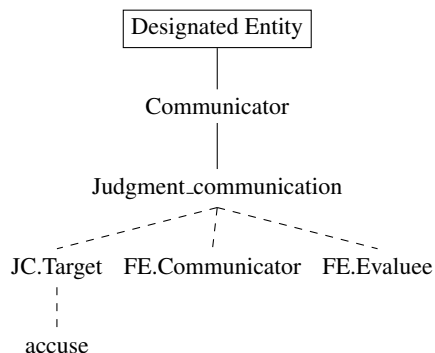


Figure 5.3: SemTree representation for the designated entity *Oracle* in sentence: *Oracle has accused Google of violating its intellectual property rights to the Java programming language.*

For the example sentence in Figure 4.1 for the designated entity *Oracle*, the semantic parse has two frames, one corresponding to the main clause (verb *sue*), and the other for the tenseless adjunct (verb *say*). The reversed paths extracted from each frame root to the designated entity *Oracle* become the backbones (Figures 5.2a & 5.2b). After merging the two backbones we get the resulting *SemTree*, as shown in Figure 5.2c. By the same steps, this sentence would also yield a *SemTree* with *Google* at the root, in the role of EVALUEE, as shown in Figure 5.2.

### 5.3 Tree Kernels to Measure Semantic Tree Similarity

The tree kernel [Moschitti, 2006; Collins and Duffy, 2002] is a function of tree similarity, based on common substructures (tree fragments). There are two types of substructures. A subtree (ST) is defined as any node of a tree along with all its descendants. A subset tree (SST) is defined as any node along with its immediate children and, optionally, part or all of the children’s descendants. Each tree is represented by a  $d$  dimensional vector where the  $i$ ’th component counts the number of occurrences of the  $i$ ’th tree fragment. SST has a finer granularity than ST when measuring similarity, while ST has a lower computational complexity.

Define the function  $h_i(T)$  as the number of occurrences of the  $i$ ’th tree fragment in tree  $T$ , so that  $T$  is now represented as  $\mathbf{h}(T) = (h_1(T), h_2(T), \dots, h_d(T))$ . We define the set of nodes in trees  $T_1$  and  $T_2$  as  $N_{T_1}$  and  $N_{T_2}$  respectively. We define the indicator function  $I_i(n)$  to be 1 if the subtree  $i$  is seen rooted at node  $n$ , and 0 otherwise. It follows that  $h_i(T_1) = \sum_{n_1 \in N_{T_1}} I_i(n_1)$  and

$h_i(T_2) = \sum_{n_2 \in N_{T_2}} I_i(n_2)$ . Their similarity can be efficiently computed by the inner product,

$$\begin{aligned}
K(T_1, T_2) &= \mathbf{h}(T_1) \cdot \mathbf{h}(T_2) \\
&= \sum_i h_i(T_1) h_i(T_2) \\
&= \sum_i (\sum_{n_1 \in N_{T_1}} I_i(n_1)) (\sum_{n_2 \in N_{T_2}} I_i(n_2)) \\
&= \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \sum_i I_i(n_1) I_i(n_2) \\
&= \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)
\end{aligned}$$

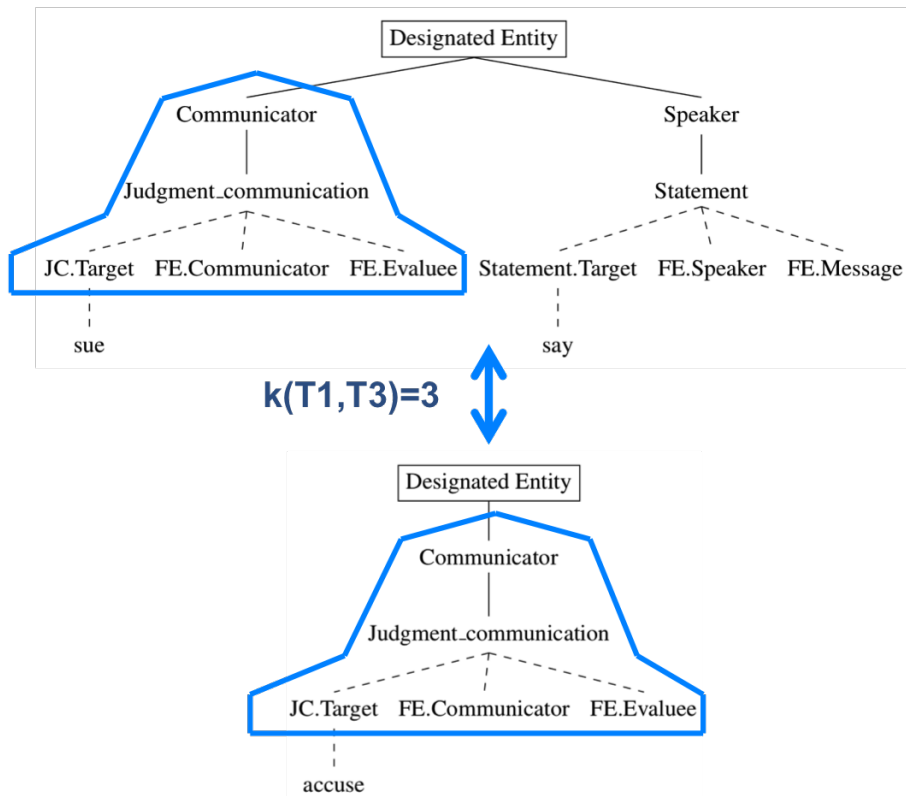
where  $\Delta(n_1, n_2)$  is the number of common fragments rooted in the nodes  $n_1$  and  $n_2$ . If the productions of these two nodes (themselves and their immediate children) differ,  $\Delta(n_1, n_2) = 0$ ; otherwise iterate their children recursively to evaluate  $\Delta(n_1, n_2) = \prod_j^{|\text{children}|} (\sigma + \Delta(c_{n_1}^j, c_{n_2}^j))$ , where  $\sigma = 0$  for ST kernel and  $\sigma = 1$  for SST kernel.

The kernel computational complexity is  $O(|N_{T_1}| \times |N_{T_2}|)$ , where all pairwise comparisons are carried out between  $T_1$  and  $T_2$ . However, there are fast algorithms for kernel computation that run in linear time on average, either by dynamic programming [Collins and Duffy, 2002], or pre-sorting production rules before training [Moschitti, 2006].

## 5.4 Tree Kernel on SemTree

For the example data instances in Table 5.1, assume instances 1 & 2 are training data and instances 3 & 4 are test data. To correctly classify the test data, we want a kernel function that measures instance 3 to instance 1 with a higher similarity score than to instance 2 ( $k(T3, T1) > k(T3, T2)$ ), and measures instance 4 to instance 2 with a higher similarity score than to instance 1 ( $k(T4, T2) > k(T4, T1)$ ). *SemTree* representation with SST tree kernel achieves our requirement. As shown in Figure 5.5, using SST tree kernel, when comparing instance 3 with instance 1:  $k(T3, T1) = 3$ , while comparing instance 3 with instance 2:  $k(T3, T2) = 1$ .

**Oracle** sued Google in August 2010, saying Google’s Android mobile operating system infringes its copyrights and patents for the Java programming language.

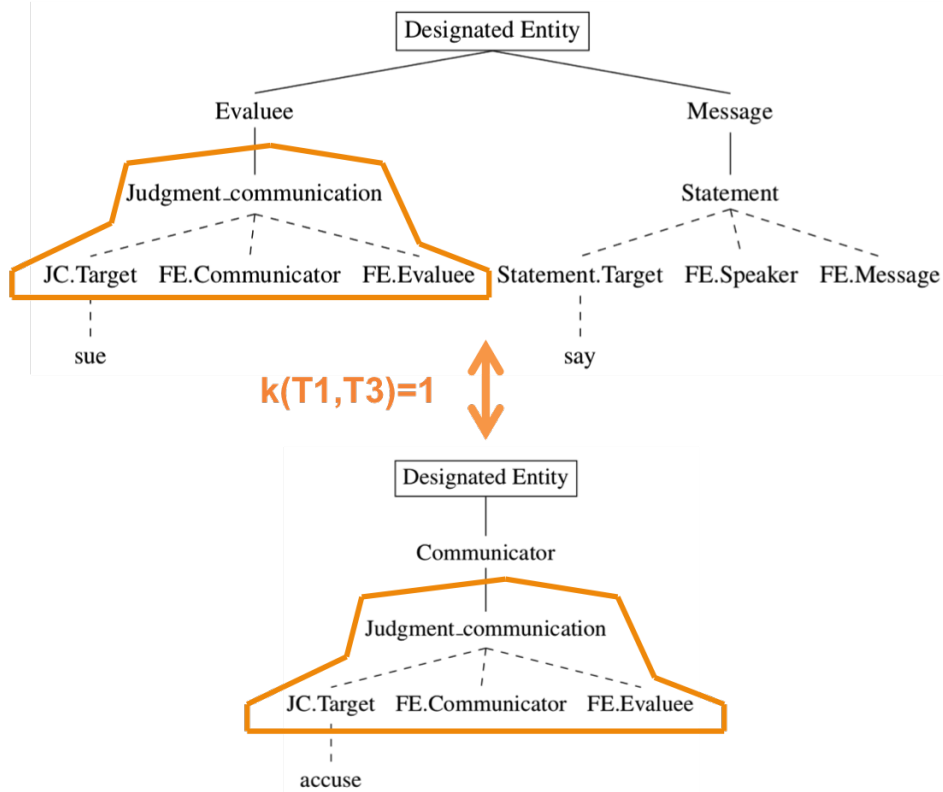


**Oracle** has accused Google of violating its intellectual property rights to the Java programming language.

Figure 5.4: Subset tree kernel for  $k(T3, T1)$  and  $k(T3, T2)$ .



Oracle sued **Google** in August 2010, saying Google’s Android mobile operating system infringes its copyrights and patents for the Java programming language.



**Oracle** has accused Google of violating its intellectual property rights to the Java programming language.

Figure 5.5: Subset tree kernel for  $k(T3, T1)$  and  $k(T3, T2)$ .

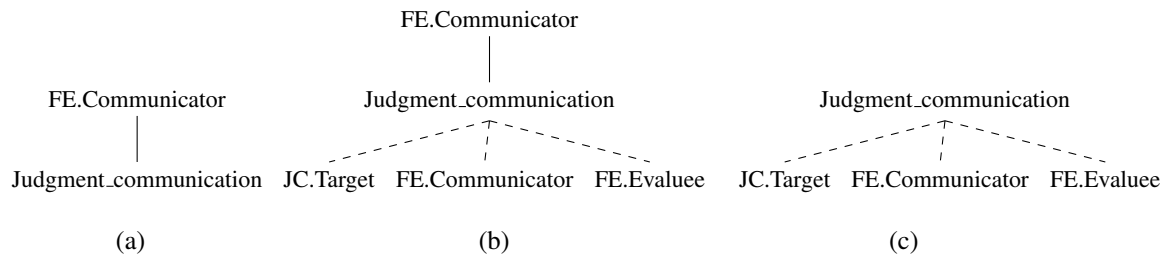


Figure 5.6: When using SemTree representation and subset tree (SST) tree kernel, (a) (b) (c) are common tree fragments when comparing instance 3 to instance 1 ( $K(T1, T3) = 3$ ), while (c) is the only common tree fragments when comparing instance 3 to instance 2 ( $K(T2, T3) = 1$ ), as shown in Figure 5.5.

## Chapter 6

# Graph Space

### 6.1 Motivation

In this section, we extend the structured semantic representation from a tree space to a semantic graph space, and apply a graph kernel as a similarity measure for machine learning. *SemTree* in the previous section captures information about designated entities through semantic role and frame information derived from semantic parses. It encodes a subset of semantic frames where a designated entity fills a role. The root of the tree is the designated entity and the tree has a pre-defined depth. The frames without a designated entity as a role filler are discarded. In a preliminary experiment, we found that incorporating *SemTree* in the vector space model improved the performance but *SemTree* alone did not. The question is, can we do better than *SemTree* using a topologically more general representation (e.g. allow cycles and no distinction between root and leaf nodes)?

Graphs are a flexible and efficient data structure for problems as diverse as prediction of toxicity based on molecular structure [Wale *et al.*, 2008], analysis of 3-D scenes in virtual environments [Fisher *et al.*, 2011], and social network analysis [Ediger *et al.*, 2010]. They have been used in many NLP tasks, such as polarity of words [Hassan and Radev, 2010], opinion bearing words and opinion targets [Sayeed *et al.*, 2012], coreference [Nicolae and Nicolae, 2006], and dependency parsing [McDonald *et al.*, 2005b]. We have found, however, little if any work that applies graphs to large-scale semantic analysis of documents. Most of the studies on graphs are based on the assumption that data instances with similar structure have similar outcomes. The problem of how to construct a meaningful graph representation and how to measure the similarity of graphs is at

the core of learning on graphs. This motivates our study on graph-structured semantic information derived from texts.

Consider the following 3 sentences selected from news articles:

“The accreditation renewal also *underscores* the *quality* of our work with **Humana** members, customers, clients, payors and health care providers by confirming our compliance with national standards for PBM services,” *said* William Fleming, *vice president* of **Humana Pharmacy Solutions**. (a)

“The testing program *highlighted* the *abilities* of the Navy, **Raytheon Missile Systems** and NASA to effectively partner on this very complicated testing program and deliver what would have been previously unobtainable data,” *said* Don Nickison, *chief* of the NASA Ames Wind Tunnel operations division. (b)

“The initiation of a dividend and the renewed share repurchase authorization *underscore* the board and management’s confidence in **Symantec**’s long-term business outlook and *ability* to generate significant free cash flow on a consistent basis,” *said* **Symantec**’s *executive vice president and chief financial officer*, James Beer. (c)

Each of the above three sentences mentions a company in a different market sector, *Humana* in Healthcare (a), *Raytheon* in Industrials (b), and *Symantec* in Information Technology (c). They describe different events and use diverse words in their own domains. Interestingly, the stock price of all three companies went up the next day after the news. All three sentences describe a scenario that a leader in an organization is making a statement that conveys the importance of some aspect of the company’s capability. The relevant frames include *Leader*, *Statement*, *Convey\_importance*, and *Capability*. These semantic frames are related to and depend on one another. We hypothesize that such intra-sentence dependencies among frames can be modeled through syntactic dependency parses, and they provide important features for semantic frame-based document representation.

In this section we present a flexible and extensible graph representation that not only subsumes *SemTree*, but is also able to incorporate syntactic information to model the intra-sentence dependencies among semantic frames. It provides a unified data representation for bag-of-words, frame semantic features, and syntactic dependencies. For machine learning on graphs, we apply a graph kernel that can capture similarities within various degrees of node neighborhood.

We start from a graph representation that builds on frame semantic features which extends *SemTrees*, and we name it *Vanilla SemGraph*. We then introduce other feature types that can be incorporated into this flexible graph representation, such as syntactic dependencies (*SemDepGraph*), and lexical items (*SemLexGraph*). At the end, we combine all these feature types and we present

our resulting graph representation as *SemDepLexGraph*, or simply *OmniGraph*.

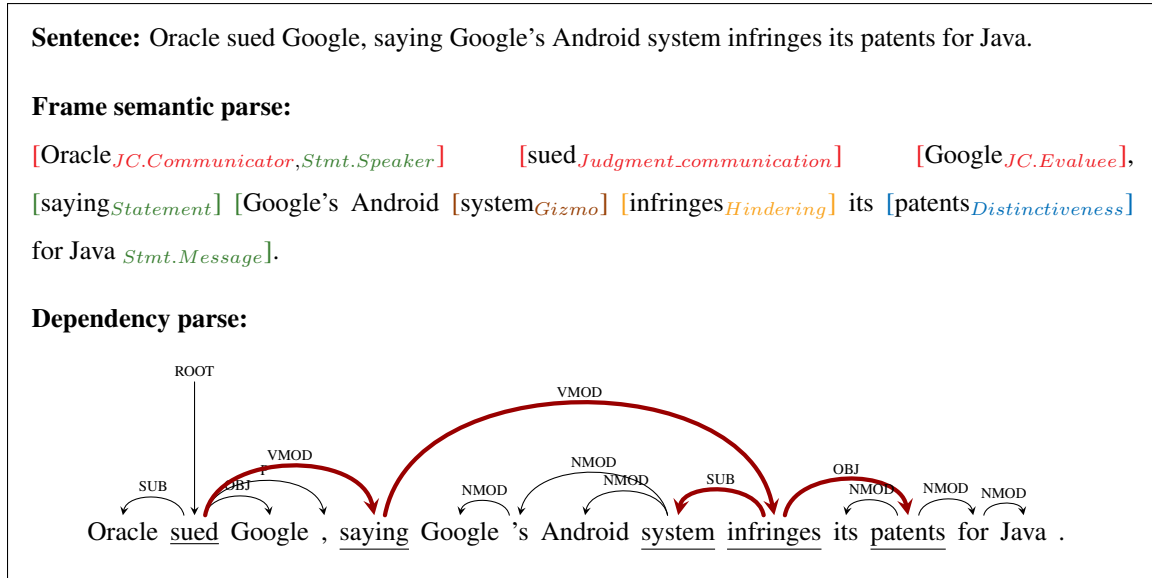
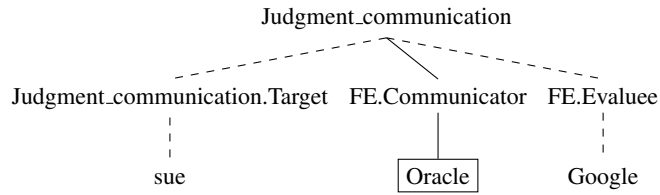


Figure 6.1: Example sentence, the frame semantic parse, and the dependency parse.

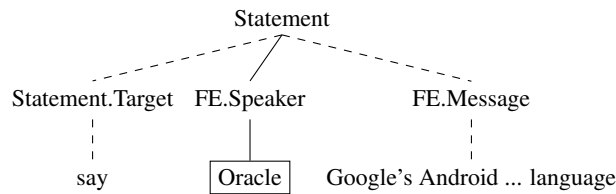
## 6.2 Constructing Semantic Graph Representation

Our semantic graph aims at a concise and convenient representation of linguistic semantic information. For each sentence that mentions an entity of interest, the representation should identify the frame, and should include the semantic relations that the entity participates in. Criteria that guided our design decisions were to 1) focus on a designated entity; 2) capture semantic roles and other semantic features of the entities; 3) have a more general topology than trees, e.g., with cycles, and no distinctive root or leaf nodes; and support feature engineering through 4) extensibility and 5) use of an efficient and flexible kernel.

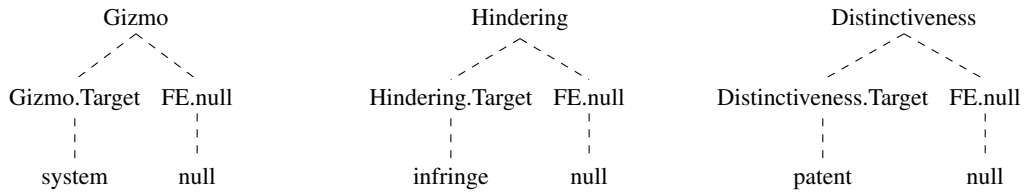
Frames, frame targets (lexical units that evoke frames), frame elements (roles), and the entity of interest appear as nodes in a *Vanilla SemGraph*. The entity to be modeled is the *Designated Entity* (DE; e.g. *Oracle* of the sentence in Figure 6.1). Figure 6.3a) shows an example *Vanilla SemGraph* for the sentence in Figure 6.1. For readability, nodes in the figure are distinguished by shape: ellipses for entities, boxes for frames, rounded boxes for frame targets, and diamonds for



(a) *Judgment\_communication* frame where two frame elements have been filled. The designated entity *Oracle* fills the Communicator role.

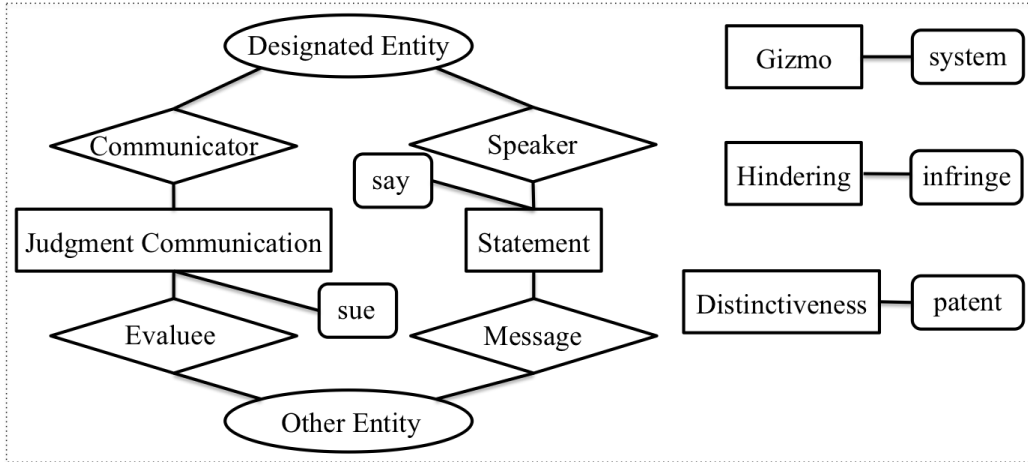


(b) *Statement* frame where two frame elements have been filled. The designated entity *Oracle* fills the Speaker role.

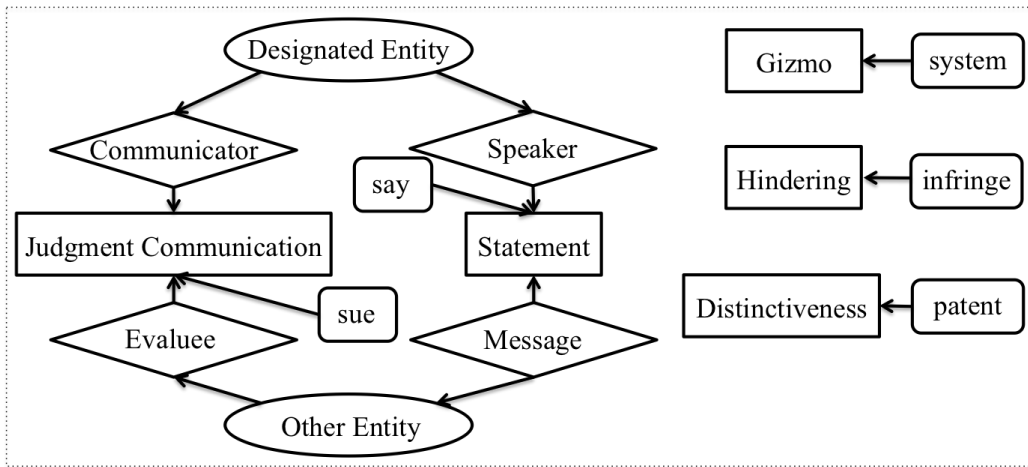


(c) Three other frames that are identified in the sentence. No frame elements are filled for these three frames.

Figure 6.2: Semantic frames that are evoked for the sentence of Figure 4.1. Unlike SemTree where only the frames with designated entity are used, semantic graph representation make use of all frames.

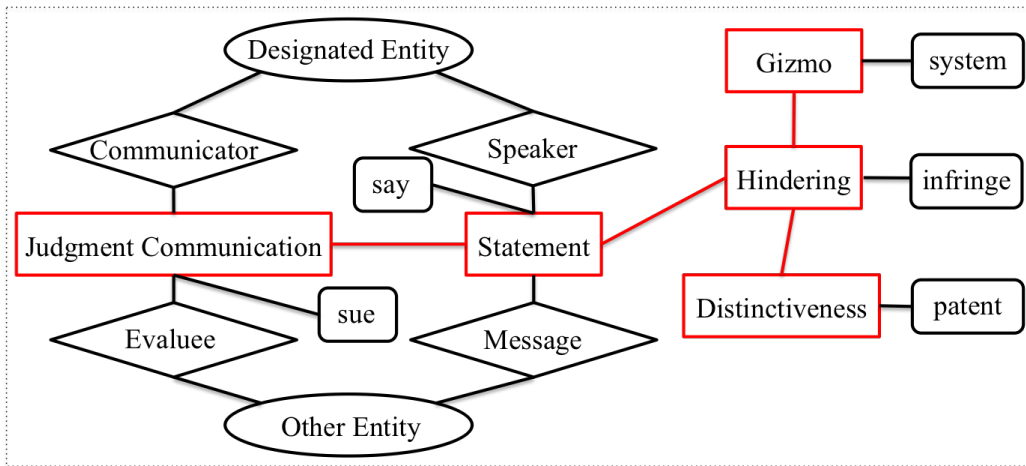


(a) Vanilla SemGraph representation.

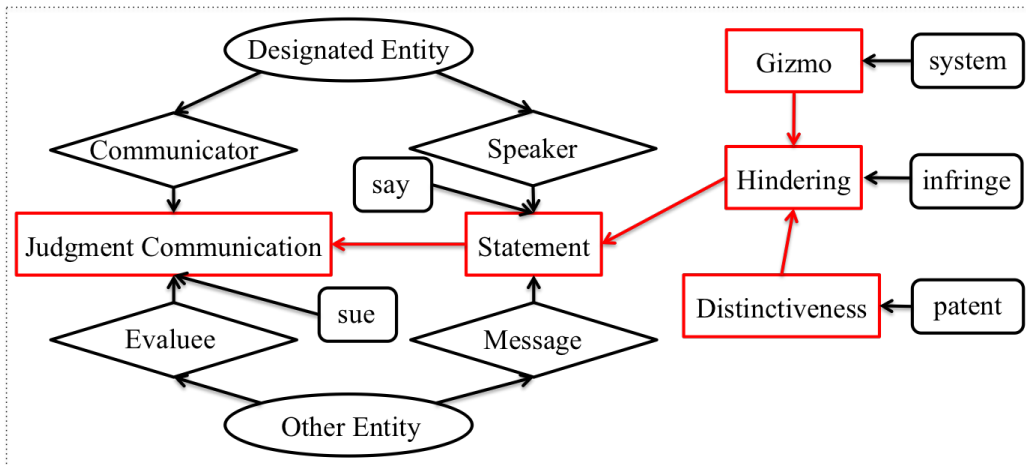


(b) Directed Vanilla SemGraph representation.

Figure 6.3: Eight variants of graph representation for Oracle of sentence 1

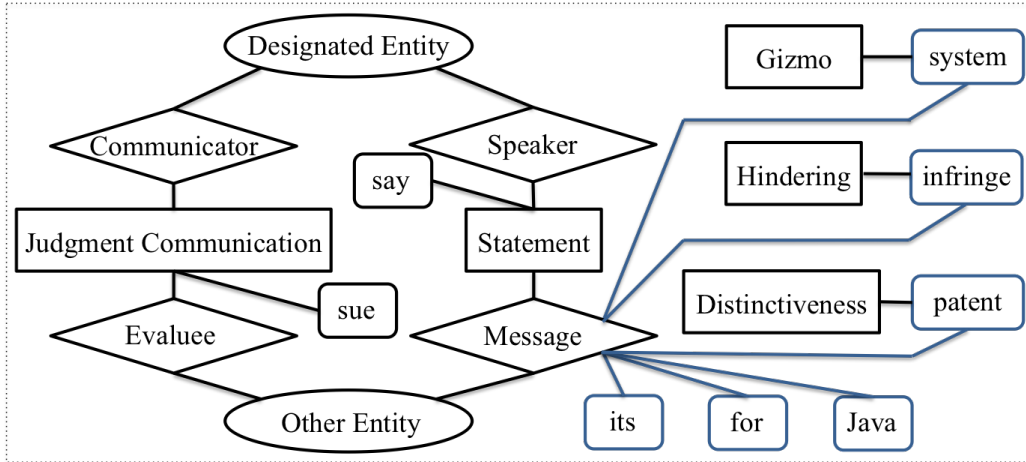


(a) SemDepGraph representation.

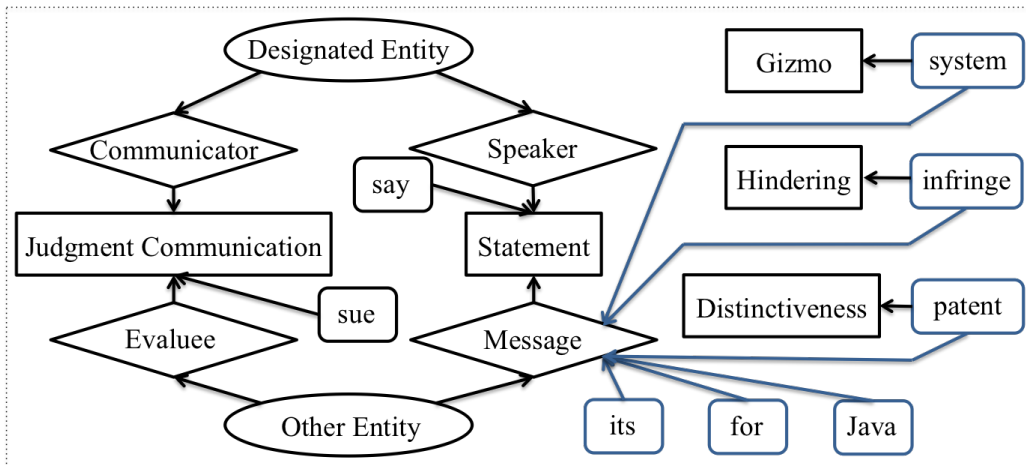


(b) Directed SemDepGraph representation.

Figure 6.4: Eight variants of graph representation for Oracle of sentence 1



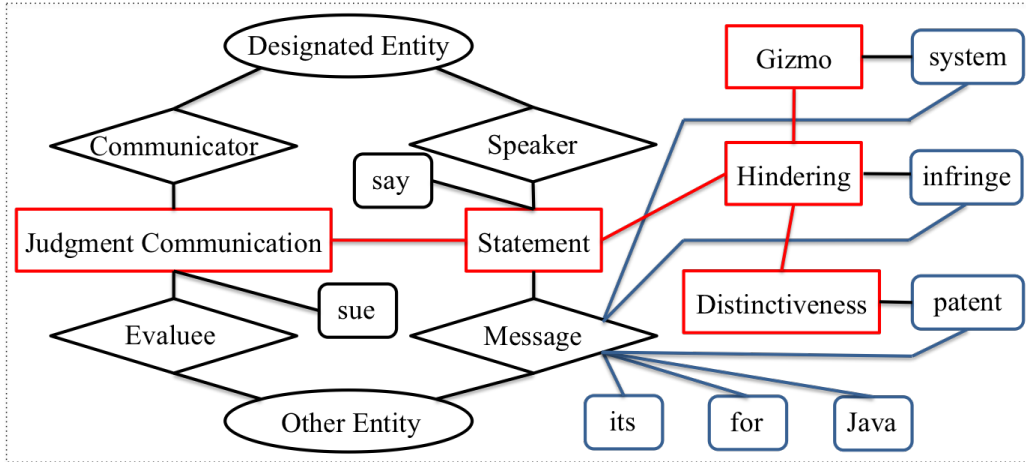
(a) SemLexGraph representation.



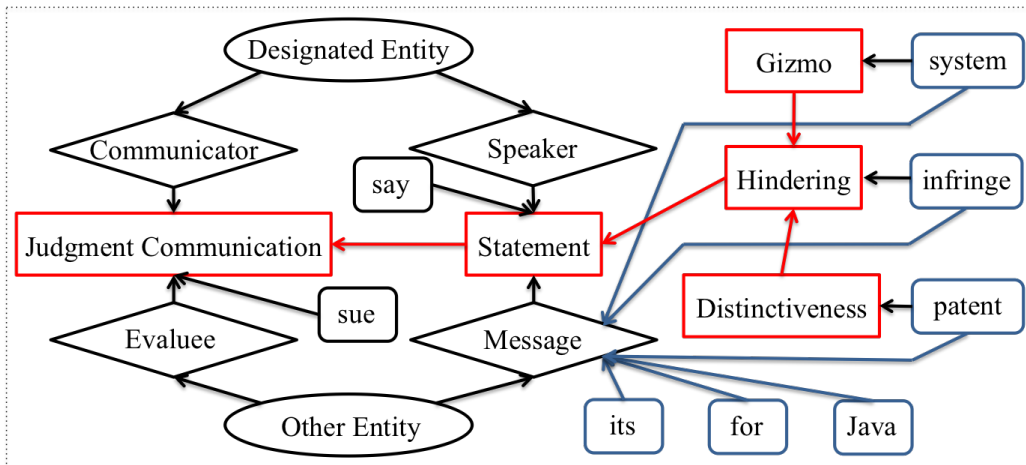
(b) Directed SemLexGraph representation.

Figure 6.5: Eight variants of graph representation for Oracle of sentence 1





(a) SemDepLexGraph, or OmniGraph, representation.



(b) Directed SemDepLexGraph, or OmniGraph, representation.

Figure 6.6: Eight variants of graph representation for Oracle of sentence 1

frame elements. For the *Vanilla SemGraph*, an edge connects two nodes in the following cases: (1) a frame target and the frame it evokes (e.g.  $\langle \textit{sue}$  and  $\textit{Judgment\_Communication}$ ); (2) a frame element and the frame it belongs to (e.g.  $\langle \textit{Speaker}$  and  $\textit{Statement}$ ); and (3) an entity and the frame element it fills (e.g.  $\langle \textit{Designated\_Entity}$  and  $\textit{Speaker}$ ). Other frames identified in this sentence, such as the *Hindering* frame evoked by *infringe*, are listed as independent subgraphs. Other variants of our semantic graph, as will be described in the next section, have different nodes and edges.

### 6.3 Variations of Semantic Graph Towards OmniGraph

The inherent extensibility of *Vanilla SemGraph* enables us to engineer variant representations with ease.

**SemDepGraph** captures intra-sentence syntactic dependencies among frames. In the example sentence, the *Statement* frame is evoked by the main clause verb, *Convey\_importance* is evoked by the verb *underscore* of the embedded statement, and *Capability* is evoked by the head noun of the direct object of *underscore*. These dependencies are partly represented as nested frames, but are more generally recoverable from the dependency parse. In Figure 6.4a), the edges in red derive from the dependency parse in Figure 6.1. Modeling the dependencies among frames also allows us to reduce the size of the graph. For example, *SemDepGraph* can retain frames that are along a dependency path from a designated entity mention to the root of the dependency tree. In this way, frames that are syntactically more subordinate can be dropped because they are semantically peripheral.

**SemLexGraph**: *Vanilla SemGraph* and *SemDepGraph* express lexical information only in the nodes that represent the frame targets. To investigate the contribution of other lexical material, for every phrase that fills a frame element, *SemLexGraph* contains nodes for each content word in the phrase. An edge connects each such lexical node to the frame element. As shown in Figure 6.5a), here the additional lexical nodes consist of *accreditation*, *renewal*, *our*, *work*, and *members*.

**SemDepLexGraph**, or simply **OmniGraph**, incorporates both the lexical items and the syntactic dependency information.

**Directed Graphs** exist for all our semantic graph variations listed above, where all edges become directed. The direction of an edge is determined by syntactic dependency as follows: 1) the frame in a subordinate clause depends on a frame in its superordinate clause; 2) frame elements depend on

their frames; 3) words that fill a frame element depend on the frame element; 4) designated entity nodes depend on the frame element where they are the role filler. Figure 6.6b) shows the directed *OmniGraph* for the sample sentence.

Figure 6.8 demonstrates another example of *OmniGraph* construction for a more complex sentence. Recall that the criteria that guided our design decisions for *OmniGraph* were to 1) represent lexical, syntactic and frame semantic information; 2) have a more general topology than trees, e.g., with cycles, and no distinctive root or leaf nodes; and support feature engineering through 4) extensibility and 5) use of an efficient and flexible kernel. *OmniGraph* consists of three levels of linguistic features: words, syntactic dependencies, and frame semantics. These three levels of representation have independently been found useful for text classification and opinion mining. In particular, frame semantics [Fillmore, 1976], which generalizes from words and phrases to abstract scenarios, or frames, has proved useful for diverse tasks, including polarity classification of financial news [Xie *et al.*, 2013], opinion mining [Kim and Hovy, 2006], and social network analysis [Agarwal *et al.*, 2014].

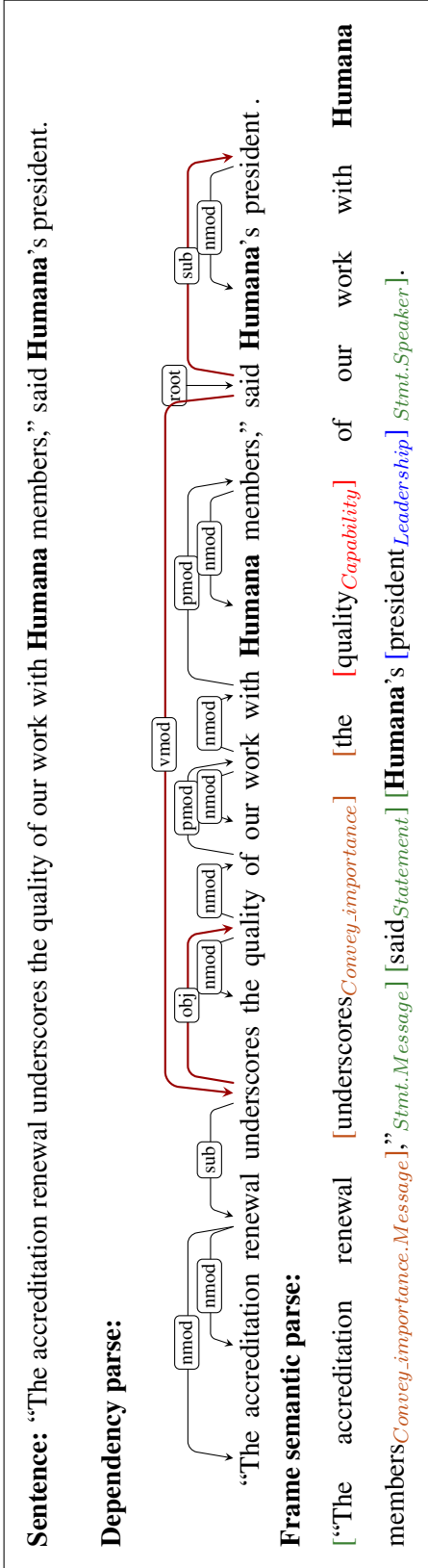


Figure 6.7: Example sentence, the dependency parse, and the frame semantic parse. The red edges in the dependency parse helps recover the interactions among frames.

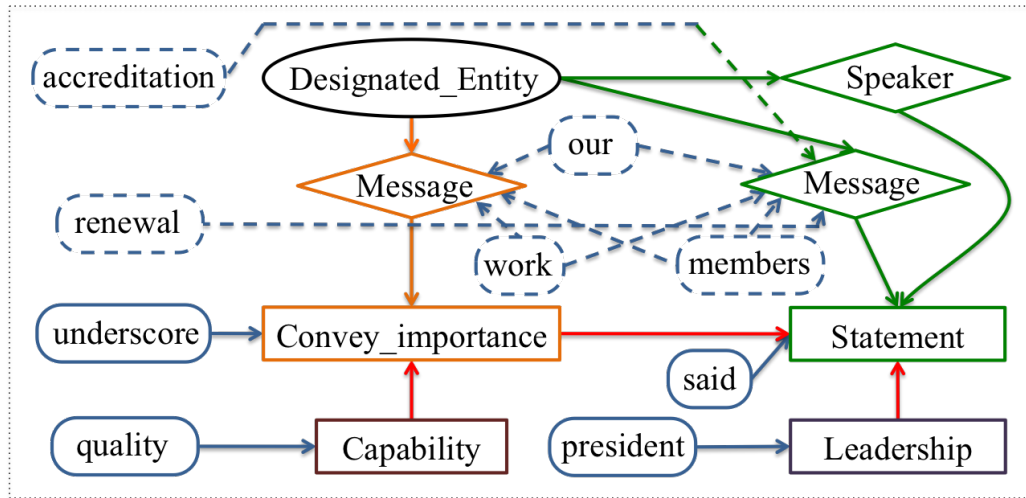


Figure 6.8: OmniGraph representation that includes lexical, dependency, and semantic information for *Humana* of the sample sentence in Figure 6.7.

## 6.4 Graph Kernels to Measure Graph Similarity

Among a variety of graph kernels as described in Section 2.3, we selected the Weisfeiler-Lehman (WL) graph kernel [Shervashidze *et al.*, 2011] for SVM learning and extensive feature exploration. It can measure similarity between graphs with respect to different neighborhood sizes specified by the user. This allows us to test many classes of *SemGraph* features for minimal engineering cost. It also has a lower computational complexity compared to other graph kernels.

The idea behind the WL graph kernel is that at each degree  $n$  of neighborhood, all nodes are relabeled with their neighborhoods, then graph similarity is measured. For example, to represent first degree neighbors, the immediate neighborhood of the *Convey\_importance* node is used to relabel the node as  $\{Convey\_importance \rightarrow Capability, Message, Statement, underscore\}$ . We first illustrate with a toy example, then provide the algorithm.

Figure 6.9 illustrates how to calculate the WL graph kernel between graphs  $G$  and  $G'$  for degrees of neighbor up to 1 ( $h=1$ ). Iteration  $i=0$  for degree of neighbor 0 (stepsize 0) compares only the nodes of the original graphs. Nodes with label 0 have one match; nodes with label 1 have two matches. This gives a total similarity of three. The neighborhoods for each node are then augmented to compute similarity when iteration  $i=1$ , which compares the nodes and their first degree neighbors.

New labels (i.e. 3, 4, and 5) are assigned as shown in the figure, and similarity is computed for the relabeled nodes. Therefore,  $k^{h=1}(G, G') = k(G_0, G'_0) + k(G_1, G'_1) = 3 + 2 = 5$ .

The WL kernel computation is based on the Weisfeiler-Lehman test of isomorphism [Weisfeiler and Lehman, 1968], which iteratively augments the node labels by the sorted set of their neighboring node labels, and compresses them into new short labels, called multiset-labels. WL graph kernel applies the idea of neighbor augmentation to iteratively measure the similarity between graphs using dynamic programming.

Define the  $n$ -degree neighborhood of a node  $v$  as the set of nodes exactly  $n$  steps away from  $v$ . For graph  $G = (V, E, \ell) = (V, E, l_0)$  denote a graph of  $i$ -degree neighbor as  $G_i = (V, E, l_i)$ . Define WL graph sequence of degree up to  $h$   $\{G_0, G_1, \dots, G_h\} = \{(V, E, l_0), (V, E, l_1), \dots, (V, E, l_h)\}$ , where  $V$  are vertices,  $E$  are edges,  $l$  are labels of vertices, and  $G_0 = G$ . Note that neither  $V$  nor  $E$  ever change in this sequence, but only the labels  $l$ .

The general WL graph kernel with up to  $h$  degree neighbor is defined as  $k_{WL}^{(h)}(G, G') = k(G_0, G'_0) + k(G_1, G'_1) + \dots + k(G_h, G'_h)$ , where  $\{G_0, \dots, G_h\}$  and  $\{G'_0, \dots, G'_h\}$  are the WL sequences of graphs  $G$  and  $G'$  respectively.  $k$  is the kernel that counts the number of common subgraph patterns.

In the kernel computation between graphs  $G$  and  $G'$ , define  $\Sigma_i \subseteq \Sigma$  as a set of strings that occur as node labels in the  $i^{th}$  iteration for the  $i$ -degree neighbor computation. In particular,  $\Sigma_0$  is the set of original node labels of  $G$  and  $G'$ . Without loss of generality, assume that every  $\Sigma_i = \{\sigma_1^{(i)}, \dots, \sigma_{|\Sigma_i|}^{(i)}\}$  is ordered.  $c_i(G, \sigma_j^{(i)})$  is the number of occurrences of the string  $\sigma_j^{(i)}$  in the graph  $G$ .  $\sigma_j^{(i)}$  is used as an id for a substructure in the  $i^{th}$  iteration (of the  $i$ -degree neighbor graph). It serves as a storage of computed information in dynamic programming.

Algorithm 1 summarizes the procedure to compute the kernel matrix for  $N$  graphs. It takes

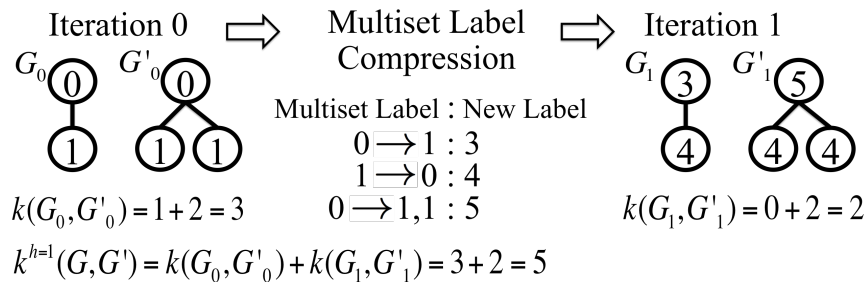


Figure 6.9: Toy example of the WL graph kernel

**Algorithm 1: Weisfeiler-Lehman graph kernel**


---

```

1 Procedure compute_WL_graph_kernel ( $\{G_1, \dots, G_N\}, h$ )
2 for  $i \in 0 : h$  do
3   for each node  $v$  in  $\{G_1, \dots, G_N\}$  do
4     // multiset-label determination
5     if  $i == 0$  then
6        $M_0(v) \leftarrow l_0(v) = \ell(v)$ 
7     else
8       Assign a multiset-label  $M_i(v)$  to  $v$ , which consists of the multiset  $\{l_{i-1}(u) | u \in \mathcal{N}(v)\}$ .
9     end
10
11     // sorting each multiset
12     Sort elements in  $M_i(v)$  in ascending order and concatenate them into a string  $s_i(v)$ .
13     Add  $l_{i-1}(v)$  as a prefix to  $s_i(v)$ , i.e.  $s_i(v) \leftarrow l_{i-1}(v) + s_i(v)$ .
14   end
15
16   // label compression
17   Sort all of the strings  $s_i(v)$  for all  $v$  in  $\{G_1, \dots, G_N\}$  in ascending order.
18   Map each string  $s_i(v)$  to a new compressed label using a function  $f : \Sigma^* \rightarrow \Sigma_i$  such that  $f(s_i(v)) = f(s_i(w))$  if and
19   only if  $s_i(v) = s_i(w)$ , where  $\Sigma_i$  is the set of newly added labels at iteration  $i$ , and  $\Sigma_i = \{\sigma_1^{(i)}, \dots, \sigma_{|\Sigma_i|}^{(i)}\}$ .
20
21   // relabeling
22   Set  $l_i(v) \leftarrow f(s_i(v))$  for all nodes  $v$  in  $\{G_1, \dots, G_N\}$ .
23
24   // augment feature space
25    $\mathcal{F} \leftarrow \mathcal{F} \cup \Sigma_i$ ;
26 end
27 for each pair of graphs  $(G_i, G_j)$  in  $\{G_1, \dots, G_N\}$  do
28    $k_{WL}^{(h)}(G_i, G_j) = \langle \Phi_{WL}^{(h)}(G_i), \Phi_{WL}^{(h)}(G_j) \rangle$ ,
29   where  $\Phi_{WL}^{(h)}(G) = (c_0(G, \sigma_1^{(0)}), \dots, c_0(G, \sigma_{|\Sigma_0|}^{(0)}), \dots, c_h(G, \sigma_1^{(h)}), \dots, c_h(G, \sigma_{|\Sigma_h|}^{(h)}))$ 

```

---

into account different levels of the node-labelings, from the original labelings to increasingly large  $h$ -degree neighborhoods (stepsizes). The full kernel for a given  $h$  is then the sum of the kernel computations for each stepsize from 0 to  $h$ .

**Analysis of Complexity** For  $N$  graphs, let  $m$  be the maximum number of edges of a graph, and  $n$  be the maximum number of nodes. Determining the multiset-labels takes  $O(Nm)$ . Sorting each multiset takes  $O(Nn + Nm)$ , which can be done via counting sort. Computing  $\Phi_{WL}^{(h)}$  on all  $N$  graphs in  $h$  iterations is thus  $O(Nhm)$ , assuming  $m > n$ . To get pairwise  $k_{WL}^{(h)}$  requires  $O(N^2hn)$ , as each  $\Phi_{WL}^{(h)}$  of a graph has at most  $hn$  non-zero entries. It brings the overall runtime to  $O(Nhm + N^2hn)$ .

**Relation to *SemTree*** Our constructed semantic graph with Weisfeiler-Lehman graph kernel learning is a natural extension to the semantic tree representation. Figure 6.10a shows a subtree pattern of  $h = 3$  rooted at the node of *Designated Entity*. Colored arrows are shown for different step sizes. Unfolding of this subtree pattern results in the substructure of Figure 6.10b. The dashed area is equivalent to the semantic tree of Figure 5.2c.

## 6.5 WL Graph Kernel Computation Example

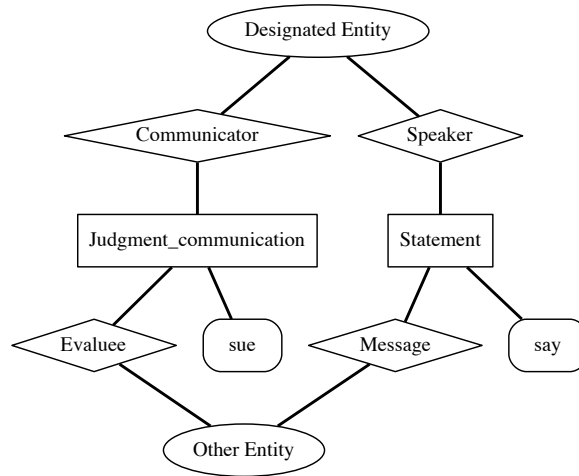
We use the same set of data instances in Table 5.1 as an example. *SemGraph* representation with WL Graph Kernel learning is also able to achieve the requirement that measures instance 3 to instance 1 with a higher similarity score than to instance 2 ( $k(G3, G1) > k(G3, G2)$ ), and measures instance 4 to instance 2 with a higher similarity score than to instance 1 ( $k(G4, G2) > k(G4, G1)$ ).

Figure 6.11 shows the procedure for computing the kernel between instance 1 and instance 2. Figure 6.11a & 6.11b assign initial labels ( $l_0$ ) to the original graphs shown, and it results in the graphs (Figure 6.11c & 6.11d) after iteration 0. The kernel for iteration 0 is a comparison between the two graphs only by nodes:

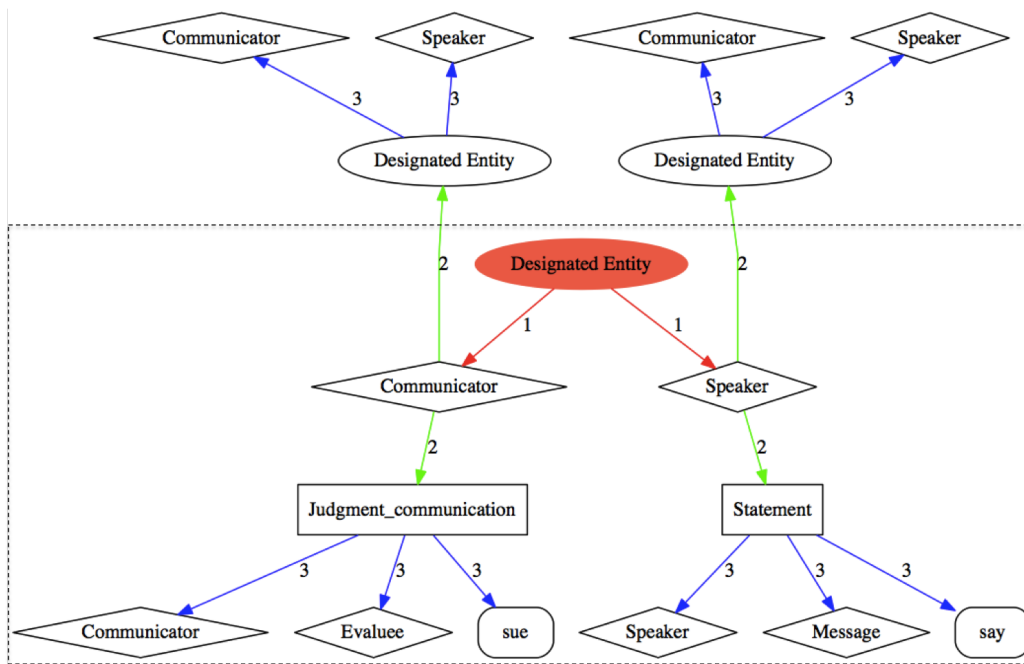
$$\begin{aligned}\Phi_{WLsubtree}^{(0)}(G_1) &= (1, 1, 1, 1, 1, 1, 1, 1, 1, 1) \\ \Phi_{WLsubtree}^{(0)}(G_2) &= (1, 1, 1, 1, 1, 1, 1, 1, 1, 1) \\ k_{WLsubtree}^{(0)}(G_1, G_2) &= \langle \Phi_{WLsubtree}^{(0)}(G_1), \Phi_{WLsubtree}^{(0)}(G_2) \rangle = 10\end{aligned}$$

In iteration 1, the 1 degree neighbor connectivity is included. Multiset-labels are created, sorted, prefixed, and assigned to the nodes (Figure 6.11e & 6.11f). With label compression (Figure 6.11g), a





(a) A subtree pattern of the semantic graph in Figure 6.3a, rooted at node *Designated Entity* with subtree pattern of height 3.



(b) Unfolding the subtree pattern of height 3 rooted at node *Designated Entity*, generated from (a).

Figure 6.10: A subtree pattern of height 3 rooted at the node of *Designated Entity*, and the unfolding of this subtree pattern. The dashed area is equivalent to the semantic tree of Figure 5.2c.

new set of labels ( $l_1$ ) are created and assigned to the nodes, and it results in the graphs (Figure 6.11h & 6.11i) after iteration 1. The kernel is a comparison consists of both the nodes and their one-degree neighbor:

$$\Phi_{WLsubtree}^{(1)}(G_1) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1)$$

$$\Phi_{WLsubtree}^{(1)}(G_2) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1)$$

$$k_{WLsubtree}^{(1)}(G_1, G_2) = \langle \Phi_{WLsubtree}^{(1)}(G_1), \Phi_{WLsubtree}^{(1)}(G_2) \rangle = 14$$

Suppose we have instance 3 of Table 5.1. Figure 6.12 shows the procedure for kernel computation up to 1 degree neighbor. The kernel between instance 3 & 1, and the kernel between instance 3 & 2 are:

$$\Phi_{WLsubtree}^{(1)}(G_3) = (1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)$$

$$\Phi_{WLsubtree}^{(1)}(G_1) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0)$$

$$k_{WLsubtree}^{(1)}(G_3, G_1) = \langle \Phi_{WLsubtree}^{(1)}(G_3), \Phi_{WLsubtree}^{(1)}(G_1) \rangle = 7$$

$$\Phi_{WLsubtree}^{(1)}(G_3) = (1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1)$$

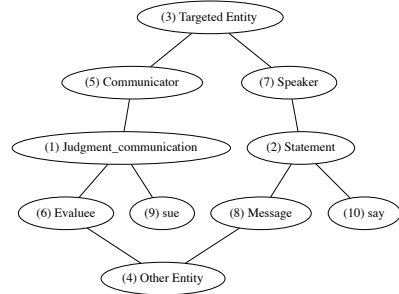
$$\Phi_{WLsubtree}^{(1)}(G_2) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0)$$

$$k_{WLsubtree}^{(1)}(G_3, G_2) = \langle \Phi_{WLsubtree}^{(1)}(G_3), \Phi_{WLsubtree}^{(1)}(G_2) \rangle = 5$$

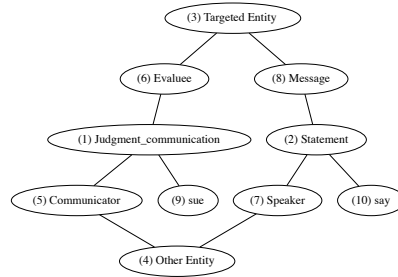
Consistent with our aims, instance 3 has a higher similarity score to instance 1 than instance 2.

## 6.6 Node Edge Weighting Graph Kernel

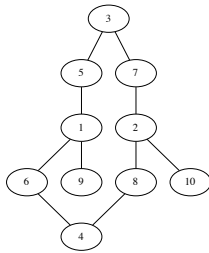
WL kernel is efficient at neighborhood augmentation but often results in coarse-grained features. For instance, in Figure 6.6, all nodes are treated equally and there is no distinction between different node types. The 1-degree WL feature for the Designated Entity (DE) node is  $\langle \text{DE} \rightarrow \text{Spkr}, \text{Msg}, \text{Msg} \rangle$ , i.e. DE fills a Speaker and two Message elements (one for the Statement frame and the other for



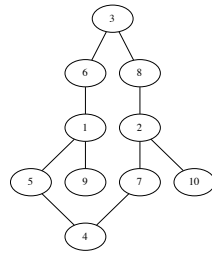
(a) Assign initial labels ( $l_0$ ) to  $G_1$ .



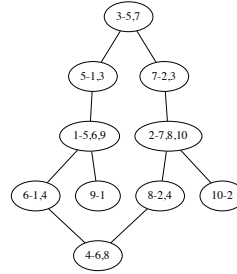
(b) Assign initial labels ( $l_0$ ) to  $G_2$ .



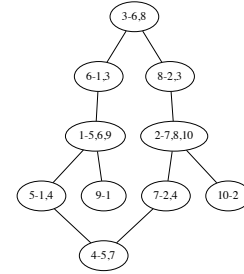
(c) After iteration 0,  $G_1$  with  $l_0$ .



(d) After iteration 0,  $G_2$  with  $l_0$ .



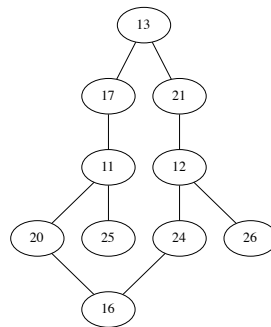
(e) In iteration 1,  $G_1$  with  $l_1$ .



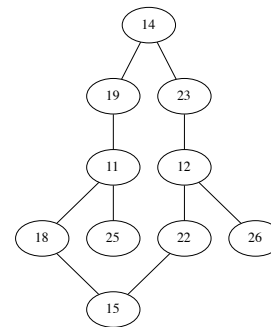
(f) In iteration 1,  $G_2$  with  $l_1$ .

1-5,6,9 $\rightarrow$ 11	5-1,4 $\rightarrow$ 19
2-7,8,10 $\rightarrow$ 12	5-1,4 $\rightarrow$ 20
3-5,7 $\rightarrow$ 13	5-1,4 $\rightarrow$ 21
3-6,8 $\rightarrow$ 14	5-1,4 $\rightarrow$ 22
4-5,7 $\rightarrow$ 15	5-1,4 $\rightarrow$ 23
4-6,8 $\rightarrow$ 16	5-1,4 $\rightarrow$ 24
5-1,3 $\rightarrow$ 17	5-1,4 $\rightarrow$ 25
5-1,4 $\rightarrow$ 18	5-1,4 $\rightarrow$ 26

(g) Label compression.



(h) After iteration 1,  $G_1$  with  $l_1$ .



(i) After iteration 1,  $G_2$  with  $l_1$ .

Figure 6.11: Procedure of the computation for Weisfeiler-Lehman graph kernel with  $h=1$  between instance 1 ( $G_1$ ) and instance 2 ( $G_2$ ) of Table 5.1.

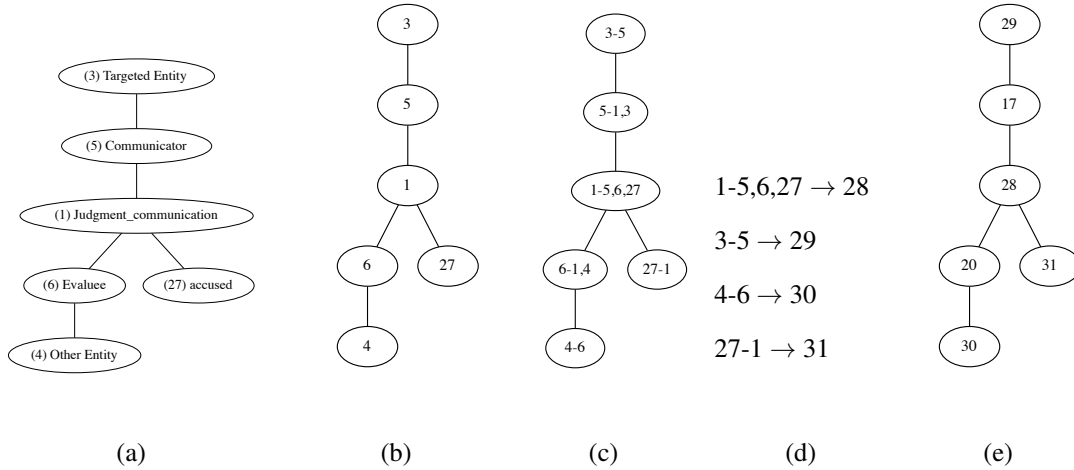


Figure 6.12: Procedure of the computation for Weisfeiler-Lehman graph kernel with  $h=1$  for instance 3 of Table 5.1 ( $G_3$ ). (a) Assign initial labels; (b) After iteration 0; (c) Sorted and prefixed multiset-label; (d) Label compression; and (e) After iteration 1.

the Convey\_importance frame). No credit for partial matching is given when this graph instance is compared to another instance where DE just fills the Message element of the Convey\_importance frame. In order to generate finer-grained features to allow partial match, and to take advantage of the type information of nodes and edges, we propose the node edge weighting (NEW) graph kernel.

Node edge weighting graph kernel also measures subgraph similarities through neighborhood augmentation. It explores OmniGraph by progressively evaluating the subgraphs of different degrees of neighborhood. Therefore, a huge amount of features are automatically being evaluated. For example, Figure 6.13 displays a sample of features that are evaluated by Node Edge Weighting graph kernel for the OmniGraph in Figure 6.8. Specifically, the neighborhoods for each node start with the individual nodes, and are augmented to each of their first degree neighbors, and then to higher degree neighbors.

The kernel computation can be broken down into node kernels and edge kernels, which measure the similarity for both nodes and edges. Node and edge kernels are weighted Kronecker delta kernels ( $\delta(\cdot, \cdot)$ ) that return whether the two objects being compared are identical. Define  $w_{\mathcal{F}_n}$  for the weight of node  $n$  of feature type  $\mathcal{F}$ , node label  $\mathcal{L}$ , and  $w_{\langle \mathcal{F}_{fr} \rightarrow \mathcal{F}_{to} \rangle}$  for the weight of edge  $e$  with from-node of feature type  $\mathcal{F}_{fr}$  and to-node of feature type  $\mathcal{F}_{to}$ . We have

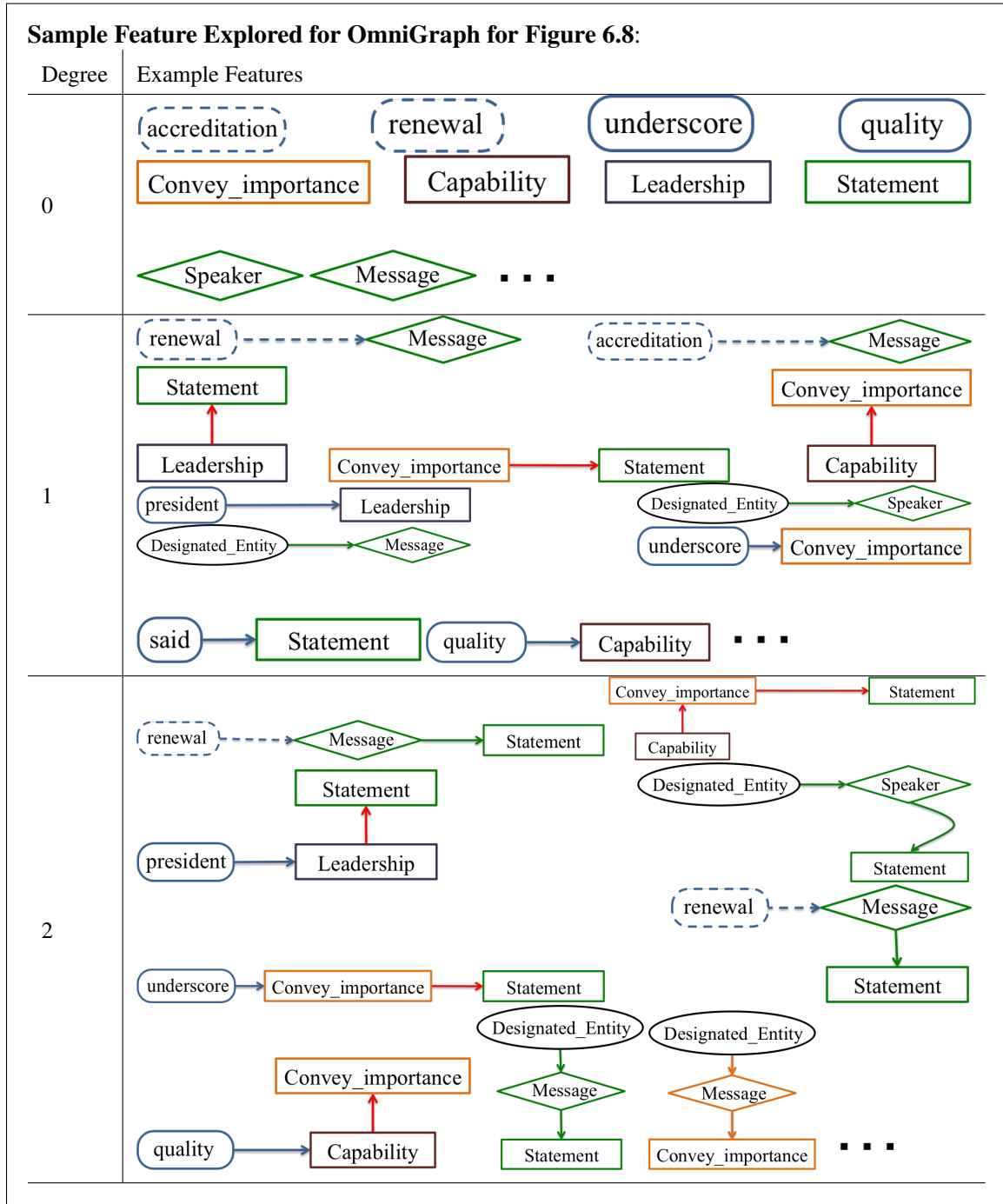


Figure 6.13: Subgraph features up to 2 degree of neighbors that are explored by Node Edge Weighting graph kernel.

$$k_{node}(n, n') = w_{\mathcal{F}_n} \cdot \delta(\mathcal{F}_n, \mathcal{F}_{n'}) \cdot \delta(\mathcal{L}_n, \mathcal{L}_{n'}) \tag{6.1}$$

$$k_{edge}(e, e') = w_{\langle \mathcal{F}_{fr} \rightarrow \mathcal{F}_{to} \rangle} \cdot \delta(\mathcal{F}_{fr}, \mathcal{F}_{fr'}) \cdot \delta(\mathcal{F}_{to}, \mathcal{F}_{to'}) \tag{6.2}$$

Define  $k^p(G, G')$  to be the basis kernel for  $p$ -degree neighborhood, the kernel between graph  $G$  and  $G'$  is computed by recursion as in Equation 6.3.

$$k^p(G, G') = \sum_{\text{all paths of length } p \in G, G'} k_{node}(n_p^G, n_p^{G'}) \prod_{i=1}^{p-1} k_{edge}(e_i^G, e_i^{G'}) k_{node}(n_i^G, n_i^{G'}) \tag{6.3}$$

Dynamic programming can be used to improve the efficiency. Each entry in the dynamic programming table is a tuple of  $\langle G, G', n_i^G, n_i^{G'} \rangle$ , where  $n_i^G$  and  $n_i^{G'}$  are nodes in graph  $G$  and  $G'$ .

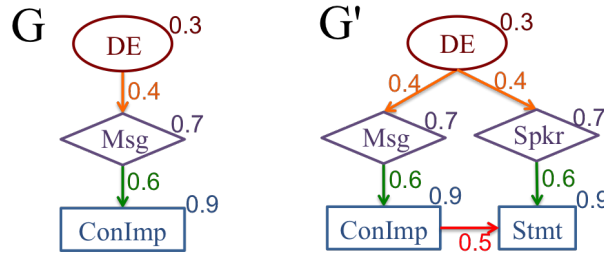


Figure 6.14: Toy example of the node edge weighting (NEW) graph kernel

The toy example in Figure 6.14 illustrates how to calculate the NEW graph kernel between graphs  $G$  and  $G'$ . As with WL GK, NEW GK compares different degrees of node neighborhoods up to  $p$  degrees of neighbors, and the final kernel is a sum of all basis kernels. For  $p=0$  degree of neighborhood, only the nodes of the original graphs are compared. Nodes with labels DE, Msg and ConImp all have one match. With node weighting,  $k^{p=0}=0.3+0.7+0.9=1.9$ . For  $p=1$ , each node plus its one-degree neighbors are compared, and the relations between the nodes. Path  $DE \rightarrow Msg$  has a match. With node and edge weighting,  $k^{p=1}=0.3*0.4*0.7=0.084$ . For the same reason,  $k^{p=2}=0.3*0.4*0.7*0.6*0.9=0.045$ . There is no match for three degrees of neighbors,  $k^{p=3}=0$ .

Each basis kernel that corresponds to different neighborhood sizes is then normalized by the maximum of the evaluation between each graph and itself. For each graph kernel  $k_G^p$  we have a normalized graph kernel  $\hat{k}_G^p$ :

$$\hat{k}^p(G, G') = \frac{k^p(G, G')}{\max(k^p(G, G), k^p(G', G'))} \quad (6.4)$$

This normalization ensures that a graph will always match itself with the highest value of 1 and other graphs with values between 0 and 1. The final kernel is an interpolation of basis kernels:

$$k(G, G') = \sum_p \alpha_p \hat{k}^p(G, G') \quad (6.5)$$

where  $\sum_p \alpha_p = 1$ . Combining basis kernels is a common problem in machine learning and several multiple kernel learning techniques have been developed to allow benefits from multiple kernels [Smits and Jordaan, 2002; Bach *et al.*, 2004]. It is also analogous to the decay function in the tree kernel of [Moschitti, 2006] where longer paths from the root get assigned lower weights. Here we do not restrict our model to a fixed decay rate, but allow more flexible weighting that can be learned from cross-validating training examples.

Our graph kernel learning is a general technique that measures the similarity between graphs. The algorithm can also be applied to other graph representation of data instances, such as *AMR*, or Abstract Meaning Representation [Cai and Knight, 2013; Banarescu *et al.*, 2013; Flanigan *et al.*, 2014].

## **Part III**

# **Experiments**



Our general goal is to ground information derived from NLP techniques applied to textual datasets in real world observations. Natural language semantics is used as a means to learn the semantic relations that are important in the real world domain, to understand what is relevant for the objectives of the practitioner. The benefit of entity-driven text analysis on real world problems allows us to model entities through text and can help predict outcomes for the entities.

In Chapter 7, we describe our experiments in a financial domain to investigate whether our methodology applied to large-scale analysis of financial news can improve our understanding of a company's fundamental market value, and whether linguistic information derived from news produces a consistent predictive power with the potential to benefit more comprehensive financial models. We align stock price data with news articles for companies in S&P500 - the Standard & Poor's 500 is an equity market index that includes 500 large companies listed on NYSE<sup>1</sup> and NASDAQ.<sup>2</sup> We formulate the task in two ways: binary classification and bipartite ranking problems. The task is to predict a company's change of price on a given day, based on the news from the previous day. This setting is aligned with many existing NLP research in the financial domain that predicts price change from news articles, e.g. [Luss and d'Aspremont, 2008], and [Feldman *et al.*, 2011b]. Predicting the price change from news is also analogous to sentiment analysis, but encompasses a more general semantic analysis.

In Chapter 8, we present another experiment to test the performance of our methods for fine-grained sentiment analysis. We work on a recently introduced sentiment analysis dataset - the GoodFor/BadFor (gfbf) corpus, which is part of MPQA (multi-perspective question answering).<sup>3</sup> Based on the two annotation tasks in the dataset, we formulate two classification problems. One task is to classify whether the agent and the event mentioned in the sentence is benefactive or malefactive on the affected entity (the object). The other task is to identify if the writer has a positive or negative attitude towards the agent and the object in the sentence.

---

<sup>1</sup>New York Stock Exchange.

<sup>2</sup>National Association of Securities Dealers Automated Quotations.

<sup>3</sup>MPQA: <http://mpqa.cs.pitt.edu/>.

## Chapter 7

# Financial News Analytics

In this chapter we describe our experiments in a financial domain to investigate whether our methods can be used for entity-driven text analytics, and forecast real world phenomenon in the stock market. More specifically, we use textual news articles to forecast the price change of the company mentions in the news. We test if our methods applied to large-scale analysis of financial news can produce a consistent predictive power, and improve our understanding of a company's fundamental market value. We hypothesize that the mileage to be gained by frame semantic feature is significant. We compare different representations and learning methods to determine experimentally what is the best way to explore and make use of these frame semantic features.

Most of the NLP literature on semantic frames addresses how to build robust semantic frame parsers, with intrinsic evaluation against gold standard parses. There have been fewer applications of semantic frame parsing for extrinsic tasks. To test for measurable benefits of semantic frame parsing, this study poses the following questions:

1. Are semantic frames useful for document representation of financial news on a large scale?
2. What aspects of frames are most useful?
3. What is the relative performance of document representation that relies on frames comparing to conventional data representations?
4. What improvements could be made to best exploit semantic frames?

We will first provide a background of our problem domain, and introduce our dataset and labeling methods. We then present the results and discussions of our methods in vector, tree, and graph-based representation and learning. We found that a rich structured representation that incor-

porates different levels of linguistic information, such as lexical items, syntactic dependencies, and semantic frames leads to a significant improvement over the baseline and benchmark methods. At the end, we also present a task on company mention detection that tests if coreference resolution is necessary in this domain.

## 7.1 Background

There has been a long-running debate on the relation between the media and the market, whether news can move the price of publicly traded companies or whether the market already incorporates news information. Evidence that the media can influence the market was presented in an influential paper by [Tetlock, 2007], where he quantified the impact of news pessimism on both price and trading volume, using a valence dictionary to measure pessimism. Because financial media provide a rich vein for NLP to mine, recent research has been investigating the use of NLP techniques for financial analysis tasks, such as price prediction for individual companies [Feldman *et al.*, 2011b; Lee *et al.*, 2014], for market sectors [Xie *et al.*, 2013] or for sets of stocks [Bar-Haim *et al.*, 2011; Creamer *et al.*, 2013], and default risk analysis based on financial reports [Kogan *et al.*, 2009]. This study looks at prediction of the price change from news for a large set of companies across market sectors. We hypothesize that detecting the semantic roles of targeted companies and the language used to describe company-involved events can benefit the price prediction task, and improve analysts' understanding of the status of companies.

One of the biggest challenges of the financial domain is the unpredictability of the market. In general, the work in NLP that uses news to predict price does well if it achieves better than 50% accuracy [Lee *et al.*, 2014; Bar-Haim *et al.*, 2011; Creamer *et al.*, 2013; Xie *et al.*, 2013]. Unlike many other domains, however, low accuracy can have great value in a high-volume trading strategy.

This work is also related to sentiment analysis. We mine opinions about entities of interest, which later feeds a ranking model. [Schumaker *et al.*, 2012] treat stock price prediction as a sentiment analysis problem to distinguish positive and negative financial news. [Tetlock, 2007] and [Tetlock *et al.*, 2008] quantify pessimism of news using General Inquirer (GI), a content analysis program. [Feldman *et al.*, 2011b] applies sentiment analysis on financial news using rule-based information extraction and dictionary-based prior polarity scores. In this study, our model addresses

a fine-grained sentiment analysis task that distinguishes different entities mentioned in the same sentence, and their distinct roles in sentiment bearing semantic frames.

## 7.2 Corpus, Data Instances, and Labeling Methods

GICS	Sector	$\mathcal{C}$	$\mathcal{N}$	$\mathcal{S}$	$\mathcal{I}$	$\mathcal{N}/\mathcal{I}$	$\mathcal{S}/\mathcal{I}$
10	Energy	39	1,823.42	3,440.00	690.74	2.64	4.92
15	Materials	27	1,276.07	3,225.86	605.93	2.11	5.32
20	Industrials	44	2,373.53	5,120.67	716.93	3.31	7.14
25	Consumer Discretionary	67	2,869.31	5,375.37	825.67	3.48	6.51
30	Consumer Staples	26	2,332.89	3,892.81	832.71	2.80	4.67
35	Health Care	39	1,962.20	2,978.71	725.22	2.71	4.11
45	Information Technology	49	3,554.90	6,161.49	857.64	4.14	7.18
55	Utilities	30	1,194.24	3,290.64	653.24	1.83	5.04

Table 7.1: Description of news data.

Reuters is an international news agency and a provider of financial market data. A typical Reuters news is generated through a sequence of processes. When a newsworthy event occurs, the first part of a story may be an *alert*, a short sentence in upper-case that contains the facts and essential detail. Often several *alerts* are filed in quick succession. A *newsbreak* is generally created 5-20 minutes after any *alerts*. *Newsbreaks* comprise a headline (often different from the *alert*) and perhaps two to four paragraphs of body text putting the facts into BODY context and making them meaningful. An *update* may be filed 20-30 minutes after a *newsbreak*. *Updates* comprise a headline (sometimes different to the headline in the original *newsbreak*) and 6-20 paragraphs of body text with further information about the event. The update may be refreshed as the story develops each subsequent *update* replaces the previous *update*, but the original *alert(s)* and *newsbreak* remain. This news provide us a high quality large-scale textual dataset.

An information extraction pipeline is used to process the data, as shown in Figure 7.1. News full text is extracted from HTML. Full text is analyzed into sentences for tokenization, dependency parsing and semantic parsing. The timestamp of the news is extracted for a later alignment with stock price information, which will be discussed in a later section.

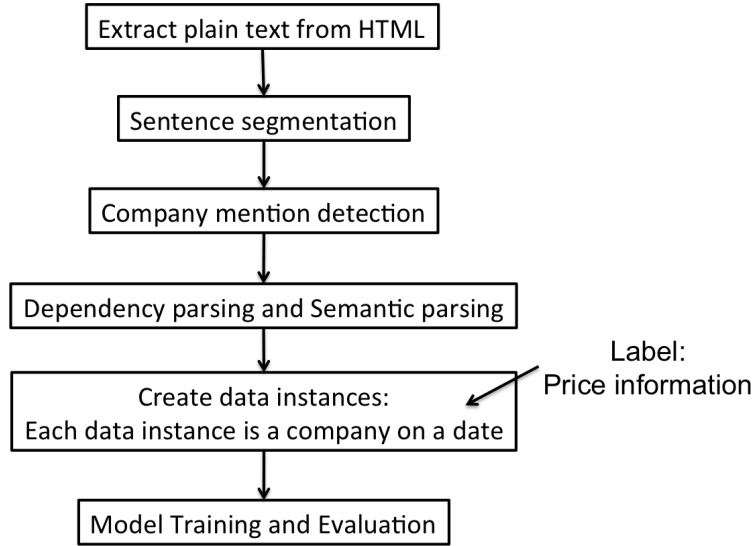


Figure 7.1: Pipeline of our experiments on Reuters news data.

Reuters news data from 2007 to 2013 that covers eight GICS (Global Industry Classification Standard) sectors are used in our experiments. GICS is a standardized classification system for equities developed jointly by Morgan Stanley Capital International (MSCI) and Standard & Poor's (S&P). The GICS structure consists of 10 sectors, 24 industry groups, 68 industries and 154 sub-industries into which S&P has categorized major public companies. The Financial sector (GICS40) is not included in this study. This sector is usually excluded from financial analytics because prices are usually not associated to market events and are not often used as investment instruments. The Telecommunications sector (GICS50) yields insufficient data because there are few companies, and companies are added and removed to the S&P 500 during our time frame. Table 7.1 describes our data.  $\mathcal{C}$  is the number of companies being modeled in each sector;  $\mathcal{N}$  is the average number of news items per company;  $\mathcal{S}$  is the average number of sentences per company;  $\mathcal{I}$  is the average number of data instances per company. In our experiment, a data instance  $\mathcal{I}$  is all the news associated to a company on a day; the number of  $\mathcal{I}$  depends on the number of days that the company is mentioned in the news.  $\mathcal{N}/\mathcal{I}$  is the average number of articles per data instance (one day of news); and  $\mathcal{S}/\mathcal{I}$  is the average number of sentences per data instance. As shown, the average number of sentences per data instance ( $\mathcal{S}/\mathcal{I}$ : sentences about a company on a given day) ranges from four to seven.

A data instance is a  $\langle Company, Date \rangle$  tuple that corresponds to all the news associated to a company on a day. We align publicly available daily stock price data from Yahoo Finance with

the Reuters news using a method to avoid back-casting. In particular, we use the daily adjusted closing price - the price quoted at the end of a trading day (4PM US Eastern Time), then adjusted by dividends, stock split, and other corporate actions. We create two types of labels for news documents using the price data, to label the existence of a change and the direction of change.

$$change = \begin{cases} +1 & \text{if } \frac{|p_{t(0)+\Delta t} - p_{t(-1)}|}{p_{t(-1)}} > r \\ -1 & \text{otherwise} \end{cases}$$

$$polarity = \begin{cases} +1 & \text{if } p_{t(0)+\Delta t} > p_{t(-1)} \text{ and } change = +1 \\ -1 & \text{if } p_{t(0)+\Delta t} < p_{t(-1)} \text{ and } change = +1 \end{cases}$$

$p_{t(-1)}$  is the adjusted closing price at the end of the last trading day, and  $p_{t(0)+\Delta t}$  is the price of the end of the trading day after the  $\Delta t$  day delay. Only the instances with price change are included in the *polarity* task. Based on the finding of a one-day delay of the price response to the information embedded in the news by [Tetlock *et al.*, 2008], we use  $\Delta t = 1$  in our experiment. In this study, we use the threshold ( $r$ ) of 2% that corresponds to a moderate fluctuation.

### 7.3 Overall Experiments

Reuters news data from 2007 to 2013 for the eight GICS<sup>1</sup> sectors shown in Table 7.1 are used in our experiments.<sup>2</sup>

An OmniGraph is created for each data instance consisting of a forest of the graphs for each sentence about a company on a given date. For all data instances of a company, we use cross-validation on 80% of the data for training to parametrize the model, and test on the 20%. Grid search is used to determine the weights of the basis kernels that correspond to different sizes of relational features and the types of features. The experiment consists of a first phase to determine the best parameters for each company. In the second phase, the selected parameters are used to test the prediction performance for each company. We report average accuracy per company for each sector, along with the majority baseline accuracy and three benchmarks, as described below.

---

<sup>1</sup>Global Industry Classification Standard.

<sup>2</sup>Two sectors are not included in this study. The Financial sector (GICS40) is usually excluded from financial analytics because prices are usually not associated to market events and are not often used as investment instruments. The Telecommunications sector (GICS50) yields insufficient data.

GICS	Sector	Baseline	BOW	SemTreeFWD	DepTree	OmniGraph <sup>WL</sup>	OmniGraph <sup>NEW</sup>
10	Energy	53.95±3.36	52.56±3.97	53.53±4.84	53.00±4.31	56.71±5.17*	56.90±4.20*
15	Materials	55.00±2.88	53.18±5.23	52.73±5.60	54.08±4.10	56.42±3.85*	56.49±3.26*
20	Industrials	54.25±3.85	52.89±5.91	52.90±5.21	53.10±3.49	55.29±4.44*	56.16±5.56*
25	Cons. Disc.	54.32±4.18	53.91±4.73	54.09±5.86	54.75±4.62	56.81±5.93*	57.49±5.63*
30	Cons. Staples	54.85±3.24	52.82±4.07	53.78±3.76	53.21±3.30	56.23±3.40*	60.64±10.7*
35	Healthcare	56.44±4.51	52.75±3.86	54.31±6.46	53.94±4.49	57.31±4.48	59.31±5.28*
45	IT	53.95±4.07	52.42±3.64	52.79±6.84	52.80±4.07	55.38±4.92*	55.69±5.61*
55	Utilities	53.82±2.75	51.66±4.24	51.75±5.23	52.56±4.45	54.86±5.70	55.30±5.32

Table 7.2: Mean accuracy by sector for the majority class baseline, three benchmarks, and two graph kernel learnings on OmniGraph. The cases where the sector mean is significantly better than the baseline are marked by \*. OmniGraph is significantly better than all three benchmarks in all cases.

### Baseline and Benchmark Methods

We use the majority class label as a baseline. We also introduce three benchmark methods for comparison. (1) *BOW*-a vector space model that contains unigrams, bigrams, and trigrams. (2) *DepTree*-a tree space representation where dependency parse of the document are encoded into a tree representation. (3) *SemTreeFWD*-an enriched hybrid of vector and tree space model that contains semantic frames, lexical items, and part-of-speech-specific psycholinguistic dictionary-based features based on [Xie *et al.*, 2013], trained with Tree Kernel SVM [Moschitti, 2006]. *SemTree* results on a smaller dataset are presented in Section 7.5.

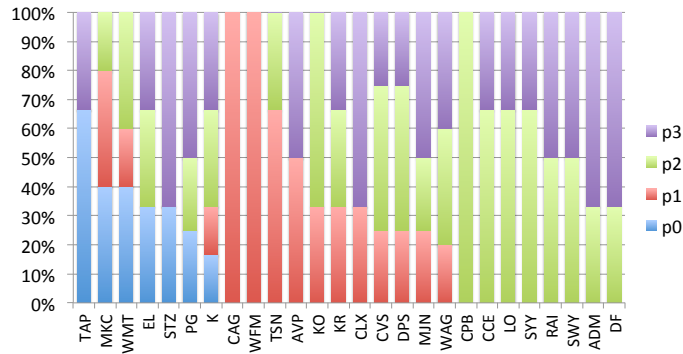
### Results

OmniGraph with NEW GK learning shows a strong impact of relational features. Figure 7.2 summarizes the results for the GICS30 Consumer Staples - a sector with medium size of news data. The stacked bar chart in Figure 7.2a) gives the breakdown for each company of the basis kernel coefficients of all stepsizes from 0 to 3, learned from the training data. Stepsizes greater than 0 correspond to relational features. On average, only 9% of the features are non-relational ( $p=0$ ).

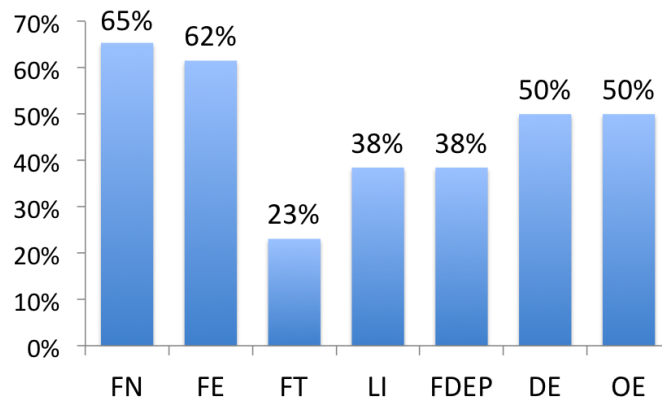
The proportion of all companies that use each of the following seven feature types appears in Figure 7.2b). The most important features are frame names (FN) and frame elements (FE): more than 60% of the companies need them to obtain the best performance. The next most required features are about entities - designated entities (DE) and other entities (OE). The importance of both DE and OE suggests that relations between companies are useful in the price prediction task. More than one third of the companies need the feature for dependencies between frames (FDEP), often involving complex sentences where multiples frames are evoked. The lexical item features (LI) have a contribution similar to FDEP. Note that LI features from OmniGraph are more than words; they include dependencies of words and the frame elements that they fill ( $p=1$ ), and the frames to which the frame elements belong ( $p=2$ ). Consistent with our expectation, frame target (FT) is the least preferred feature, because it is the lexical unit that has been generalized by the frame name. The other sectors show the same general trends.

Table 7.2 summarizes the average accuracy for all eight sectors of the majority class baseline, three benchmarks, and the two OmniGraph models. All models are trained on 80% of the data and tested on 20% for each company. Model parametrization is done by cross-validation on the training data. Both versions of OmniGraph significantly outperform the three benchmarks. The cells with





(a) Stepsizes (i.e. degrees of relational features) by companies:  $p_0$  is for non-relaitonal features;  $p > 0$  are for relational features.



(b) Percentage of companies requiring each feature

Figure 7.2: Parametrizing OmniGraph<sup>NEW</sup> for companies in Consumer Staples sector. It shows a) the breakdown by stepsize for each of the 26 companies, and b) the total proportion across companies of node-edge weights for each feature type.

asterisks represent a difference from the baseline that is statistically significant. OmniGraph<sup>WL</sup> beats the baseline with statistical significance in six out of eight sectors, and OmniGraph<sup>NEW</sup> does so in seven out of eight sectors. Note that none of the benchmarks outperforms the baseline.

Despite the excellent performance of BOW for topical classification tasks, for this price prediction task it does poorly. Both DepTree and SemTreeFWD outperform BOW, which indicates that features derived from dependency syntax and semantic frame parsing improve performance. DepTree directly represents the dependency parse with both dependencies and words as nodes, without semantic information. The limitation of SemTree comes from its entity-centric representation, the root node is the designated entity. The semantic frames without DE mentions are discarded. A heterogeneous combination of trees and vectors are used for learning. In contrast, OmniGraph with graph kernel learning learns a model in a more uniform and effective way. Between WL and NEW learning on OmniGraph, NEW produces the best results. We suspect this is due to the high granularity of the features it generates, and its flexibility in assigning different weights to nodes and edges, depending on the node and edge feature types.

### Discussion

To understand the difference in performance across document representations, we analyzed predictive features from each kind of document representation investigated here. Using mutual information to rank features for each method, we found that the more expressive the representation, the more insight the features provide. This is illustrated in Figure 7.3, which presents predictive features from vector space (VS), the dependency tree (DT); SemTree (ST); OmniGraph (OG).

Consider features 1, 2, 3, each of which is predictive due to news that refers to a company's change of price. Feature 1 is an individual lexical item (*fell*). Feature 2, a frame name triggered by lexical units such as *fell*, is somewhat more general. Neither captures the important relational information represented in features 3 and 5. Feature 3, which also has the lexical item *fell*, captures the important dependency relation that the designated entity (or its shares) should be the subject.

Feature 4 is a SemTree feature that simultaneously illustrates the benefit of frame semantics and relational features: although the sentence mentions soaring prices, the important feature from this sentence is that the designated entity fills the victim role of the Cause\_Harm frame.

The OmniGraph features 5 and 6 illustrate the superior expressivity of this representation. Feature 5 is a 2-degree neighbor subgraph that consists of three frames and their inter-dependencies.



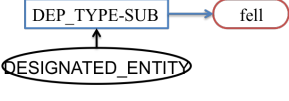

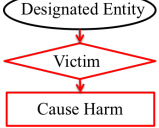

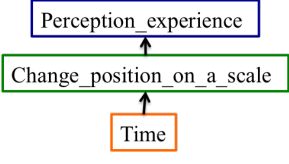

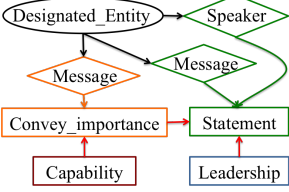

	Graph Features	Feature Types	Example Sentences	Label
1 VS	fell	Lexical item	<b>Exxon Mobil</b> fell 2.8 percent to \$86.49 and led the S&P 500's decline.	 GICS10
2 VS	Change_position_on_a_scale	Frame	<b>Wyndham</b> have seen profits soar in recent years.	 GICS25
3 DT		Syntactic dependency, Entity, Lexical item	<b>Exxon Mobil</b> fell 2.8 percent to \$86.49 and led the S&P 500's decline. After the bell, a number of high-profile tech companies reported results, including <b>International Business Machines</b> , whose shares fell 3.5 percent to \$200.01.	 GICS10 GICS45
4 ST		Frames, Frame elements	<b>Hershey</b> has been hurt by soaring prices from cocoa, energy and other commodities.	 GICS30
5 OG		Dependencies among frames	<b>Wyndham</b> have seen profits soar in recent years as robust demand has allowed them to steadily raise rates. <b>Family Dollar</b> and <b>Walmart</b> are also expected to see same store sales growth over the next 60 days.	 GICS25 GICS30
6 OG		Frames, Frame elements, Dependencies among frames	“This milestone highlighted the <b>Boeing</b> KC-767's ability to perform refueling operations under all lighting conditions,” said George Hildebrand, <b>Boeing</b> KC-767 Japan program manager. “This benchmark underlines how <b>Intel</b> can collaborate to innovate and drive real performance and total cost of ownership benefits for our clients,” said Nigel Woodward, <b>Global Director</b> , Financial Services for <b>Intel</b> .	 GICS20 GICS35 GICS45

Figure 7.3: Sample OmniGraph features (OG) that have predictive power within or across sectors, compared with those from vector space (VS), from dependency trees (DT), and from SemTree (ST).

This feature represents that the designated entity experiences a change over a time period. Note that it applies to the same sentence shown with Feature 2. It is more reliably predictive, however, than a lexical item or dependency relation because it is more precise. Although the frame *Change\_position\_on\_a\_scale* alone does not specify the polarity (e.g. evoked by *soar* or *fell*), this relational feature as a whole does not predict a negative class. We found that the language pattern of this feature rarely occurs in a negative description, and this could be a benefit of our relational features. Feature 6 is a very complex positive feature that generalizes over multiple sectors, and it is the feature that corresponds to the example sentences in Motivation section.

The example sentence with Feature 4 illustrates the potential limitations of BOW for the classification task addressed here, to predict the price change of companies. It contains the lexical item *soar*, which can be predictive of positive price change, as illustrated by Feature 2. However, the more important information is that the designated entity (*Hershey*) has been hurt by rising prices of the commodities it depends on. In contrast, OmniGraph Feature 5, a stepsize 2 feature that would co-occur with the sorts of stepsize 0 features illustrated by Feature 1 and Feature 2, captures a pattern in which reference to soaring prices is predictive when it is part of a perceived trend.

We use the standard model in SEMAFOR for frame semantic parse without retraining the model. In an evaluation of the semantic parse quality, in general, SEMAFOR parses capture most of the important frames for our purposes. On a randomly selected 40 sample sentences, two researchers working independently evaluated the semantic parses, with approximately 80% agreement. Some of the inaccuracies in frame parses result from errors prior to the SEMAFOR parse, such as tokenization or dependency parsing errors. The average sentence length for the sample was 33.3 words, with an average of 14 frames per sentence, 3 of them with a GICS company as a role filler. For the frames containing a designated object (company), on average, about half the frames with a designated object were correct, and two thirds of those frames we judged to be important. Besides errors due to incorrect tokenization or dependency parsing, we observed that about 8% to 10% of frames were incorrectly assigned to due word sense ambiguity. We also notice there is much room for improvement to have a domain-specific semantic model. Another 200 randomly selected sentences were further evaluated, and the similar trend is observed. Given the satisfying performance of the general purpose semantic parser, we project that a domain-specific semantic parser can further reduce the noise and get better frame structure patterns. For example for sentence *Citigroup raises Monster to*

*buy*. The current parse is [Citigroup<sub>Goods</sub>] raises [Monster<sub>Buyer</sub>] to [buy<sub>Commerce.buy</sub>]. However, *buy* here is an analyst's recommendation rating for investment from Citigroup on company Monster, rather than Monster being the buyer in a Commerce.buy scenario. Future work can create an annotated corpus on financial news, and retrain a domain-specific parser to improve the semantic parsing results.

Our use of natural language processing in financial news has a potential impact on the existing financial models. For example, it can be used to bridge financial analysts who are doing fundamental analysis and *quants* who are doing quantitative modeling. In fundamental analysis, financial advisors, credit analysts, or traders read financial articles to look for investment opportunities, while quantitative analysts apply mathematical models to explain financial market and to make predictions. Application of NLP on financial news to price prediction could ultimately provide a discovery mechanism to generate hypotheses to better explain how news about companies affects their price. A potential application is for the Binomial Option Pricing Model [Cox *et al.*, 1979], a numerical method based on a binomial tree that spans the option's valuation date and the expiration date with a fixed time interval and a fixed probability for upward and downward price change. The use of financial news to predict price movement allows a dynamic price change prediction and arbitrary time intervals based on the release of news. A possible improvement of the experimental setup is to incorporate the movement of the market, in contrast to the use the absolute change of price for labeling. Partly because we are predicting the short term price movement, i.e. price change of a one day delay, incorporation of the market movement does not exhibit significant difference in a preliminary experiment.

The next few sections describe side experiments that improve our understandings of different representations and learnings on financial news, and a study on name entity recognition and coreference resolution for company mention detection.

## 7.4 Vector Space Results and Discussion

To better understand more detailed performance of the vector space and the tree space models, we ran a set of experiments on three sectors: Consumer Staples, Information Technology and Telecommunication. The choice of the three sectors is due to our familiarity with the companies in these

sectors and an expectation of different characteristics they may exhibit. Table 7.3 summarizes the results for the vector space model. FWD performs nearly as well as the BOW baseline on the *change* task, and outperforms BOW on the *polarity* task. We observe that frames whose names are discriminative are consistent with predictive BOW features: e.g., the Offering frame versus the unigram *offer*; the Earnings\_and\_Losses frame versus the bigram *quarterly profit*; the Commerce\_sell frame versus the trigram *disappointing december sales*. Comparing frame names (F) and frame targets (FT), F is predictive in the *change* task and F+FT gives good performance in the *polarity* task. For example, for sentences (a) *Private equity group Champ sees strong upside for the Constellation Brands business* and (b) *Walgreens posted weaker-than-expected August sales at stores open more than a year*, Exertive\_force frame (F) alone is effective to identify both sentences to have an influence on *change*, but the frame targets (FT) of *strong* and *weaker-than-expected* further specify the *polarity*. Inverse document frequency (IDF) adjusted weights do not improve the performance. This may be indicative of an interesting difference from typical topic-based text categorization and information retrieval tasks. Prior-polarity-like pDAL features alone do not have a consistent pattern, but they often lead to improvement when combined with other features.

## 7.5 Tree Space Results and Discussion

Classification and ranking tasks are designed to experiment with and evaluate the performance of the following models:

1. Vector space model with bag-of-words, bag of semantic frame features (including frame names, lexical units, and frame elements), and word affect features based a psycholinguistic dictionary (i.e. Dictionary of Affect in Language, DAL).
2. Semantic tree (*SemTree*) model with the designated entity to be the root of the tree with its semantic roles and frame information appended as additional features.

### Classification Task and Evaluation Metrics

The classification experiments are carried out in two settings: (1) use full dataset and run cross-validation; (2) split the data into training and test and report the performance on the test set. We test the influence of news to predict (1) a change in stock price (change task), and (2) the polarity of change (increase vs. decrease; polarity task).

methods	class	change					polarity				
		pre	rec	f1	acc	MCC	pre	rec	f1	acc	MCC
BOW	+	37.73	44.39	40.79	60.43	0.1146	52.88	50.63	51.73	52.4	0.0482
	-	73.27	67.54	70.29			51.94	54.19	53.04		
BOW+pDAL	+	38.26	44.86	41.30	60.78	<b>0.1221</b>	53.53	51.16	52.32	53.09	0.0620
	-	<b>73.47</b>	67.85	70.55			52.67	55.03	53.83		
F	+	41.3	19.76	26.73	<b>66.68</b>	0.0949	53.88	58.33	56.02	53.92	0.0781
	-	71.06	87.53	78.44			53.96	49.45	51.61		
F+pDAL	+	41.17	21.38	28.15	66.42	0.0984	53.86	57.84	55.78	53.77	0.0769
	-	71.22	86.43	78.09			53.86	49.83	51.77		
F+FT	+	36.38	37.86	37.10	60.52	0.0836	55.27	54.73	54.99	54.93	0.0987
	-	71.89	70.59	71.23			54.61	55.15	54.88		
F+FT+pDAL	+	36.13	38.72	37.38	60.10	0.0816	<b>55.55</b>	53.86	54.69	<b>55.10</b>	<b>0.1023</b>
	-	71.88	69.59	70.72			<b>54.68</b>	56.36	55.51		
F+FT+FE	+	35.72	39.50	37.52	59.53	0.0772	55.12	55.14	55.13	54.84	0.0968
	-	71.80	68.43	70.07			54.56	54.54	54.55		
F+FT+FE+pDAL	+	35.73	39.99	37.74	59.41	0.0780	55.03	53.11	54.06	54.58	0.0917
	-	71.85	68.04	69.89			54.15	56.06	55.08		
F+FT+FE+BOW+pDAL	+	36.44	46.88	41.01	58.51	0.0997	55.08	54.69	54.88	54.77	0.0953
	-	72.96	63.68	68.01			54.45	54.84	54.65		

Table 7.3: FWD results for consumer staples sector for test year 2010.

In this experiment setup where one year is used for training and to predict the following year, there is high variance across years in the proportion of positive labels, and often highly skewed classes in one direction or the other. The average ratios of +/- classes for change and polarity over the six years data are 0.73 (std=0.35) and 1.12 (std=0.25), respectively. Because the time frame for our experiments includes an economic crisis followed by a recovery period, we note that the ratio between increase and decrease of price flips between 2007, where it is 1.40, and 2008, where it is 0.71. Accuracy is very sensitive to skew: when a class has low frequency, accuracy can be high using a baseline that makes prediction on the majority class. Given the high data skew, and the large changes from year to year in positive versus negative skew, we use a more robust evaluation metric, Matthews correlation coefficient (MCC) [Matthews, 1975], to avoid the bias of accuracy due to data skew, and to produce a robust summary score independent of whether the positive class is skewed to the majority or minority.  $MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$ , where TP, FP, TN and

FN are for true or false positive or negative.

### Bipartite Ranking Task and Evaluation Metrics

We assume that at best, there may be a weak predictive effect of news on price for a particular data instance, e.g. a company on a date, based on results found in previous work [Lee *et al.*, 2014; Bar-Haim *et al.*, 2011; Creamer *et al.*, 2013]. A ranking task is proposed to solve more practical problems. For example, given a particular date, investors (or traders) want to know which company will have an upward or downward movement. We cast the problem as a bipartite ranking task that ranks companies in two directions according to their probability to affect the direction of change in price. Bipartite ranking positions data points in a sequential order according to the probability that a data instance is classified as the positive class. Data at the top of the ranked list correspond to the positive class predictions, and data at the bottom are the negative class.

SVM outputs  $f(\mathbf{x}) = \sum_{i=1}^N y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - b$ , which is an uncalibrated weight that indicates the distance from the hyperplane computed by SVM classifier. This value can be converted into a probabilistic score for ranking. We follow [Wahba and Wahba, 1998] to use the logistic function to convert the general output for SVM into probabilistic form:

$$p(y_i = 1 | f(\mathbf{x}_i)) = \frac{1}{1 + \exp(-f(\mathbf{x}_i))}$$

where  $f(x)$  is the standard output for SVM.

The evaluation metrics for this ranking task include receiver operating characteristic (ROC) curves. In addition, we quantify the performance at the two ends of the bipartite ranked list, such as precision@top- $K$ , which is a standard evaluation in information retrieval to assess query result rankings. For example, when  $K = 100$  for the positive class, we report the precision of the top 100 items of the ranked list, while for the negative class, we report the precision of the bottom 100 items. The other three metrics include mean reciprocal rank (MRR), discounted cumulative gain (DCG), and PNorm scores [Rudin, 2009].

$$MRR(f) = \sum_i \frac{1}{Rank(i)} = \sum_i \frac{1}{\sum_k \mathbf{1}_{[f(\mathbf{x}_i) \leq f(\tilde{\mathbf{x}}_k)]} + \sum_i \mathbf{1}_{[f(\mathbf{x}_i) \leq f(\mathbf{x}_i)]}} \quad (7.1)$$



$$DCG(f) = \sum_i \frac{1}{\ln(1 + Rank(i))} = \sum_i \frac{1}{\ln(1 + \sum_k \mathbf{1}_{[f(\mathbf{x}_i) \leq f(\tilde{\mathbf{x}}_k)]} + \sum_i \mathbf{1}_{[f(\mathbf{x}_i) \leq f(\mathbf{x}_i)]})} \quad (7.2)$$

$$R_{p,\ell}(f) = \sum_{k=1}^K \left( \sum_{i=1}^I \ell(f(\mathbf{x}_i) - f(\tilde{\mathbf{x}}_k)) \right)^p \quad (7.3)$$

MRR (eq. 7.1) and DCG (eq. 7.2) are two weighted versions of AUC that favor the top (or the bottom) of the list. Higher values are preferred. PNorm score (eq. 7.3) corresponds to the loss of the  $l_p$ -norm objective function, where  $p$  controls the degree of concentration at the top (or one end) of the ranked list. The set of instances with positive labels is  $\{\mathbf{x}_i\}_{i=1,\dots,I}$ . The negative instances are  $\{\tilde{\mathbf{x}}_k\}_{k=1,\dots,K}$ . At the heart of this derivation,  $l_p$ -norm is used to interpolate between the  $l_1$ -norm (the AUC), and the  $l_\infty$ -norm (the values of  $R_{max}$ ). Lower values are preferred.

### Results

Our results have shown advantages of the tree space model, as shown in Table 7.4. The features encoded in this tree representation can capture interesting characteristics in different datasets (e.g. news articles about the companies in different sectors). Our post-analysis of the model also shows the benefits of semantic structural features.

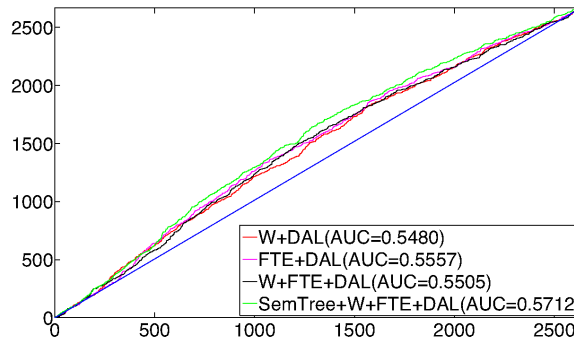
To analyze which were the best performing features within sectors, we extracted the best performing frame fragments for the *polarity* task using a tree kernel feature engineering method presented in [Pighin and Moschitti, 2009]. The algorithm selects the most relevant features in accordance with the weights estimated by SVM, and uses these features to build an explicit representation of the kernel space. Figure 7.4 shows the best performing *SemTree* fragments of the *polarity* task for the consumer staples sector.

+ (Target(jump))	- (PHENOMENON( <i>Perception_active</i> (Target)(PERCEIVER_AGENTIVE)(PHENOMENON)))
+ (RECIPIENT( <i>Receiving</i> ))	- (TRIGGER( <i>Response</i> ))
+ (VICTIM( <i>Defend</i> ))	- (Target( <i>cuts</i> ))
+ (PERCEIVER_AGENTIVE( <i>Perception_active</i> (Target)(PERCEIVER_AGENTIVE)(PHENOMENON)))	- (VICTIM( <i>Cause_harm</i> (Target( <i>hurt</i> ))(VICTIM)))
+ (DONOR( <i>Giving</i> (Target)(THEME)(DONOR)))	
+ (Target( <i>beats</i> ))	

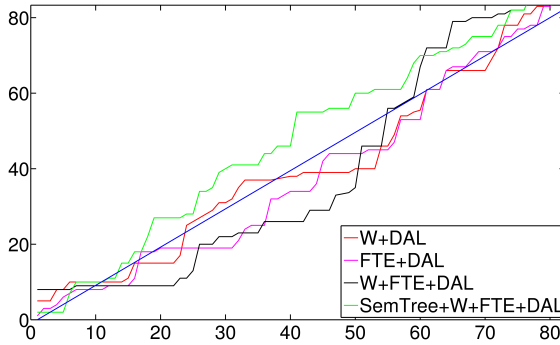
Figure 7.4: Best performing SemTree fragments for increase (+) and decrease (-) of price for consumer staples sector across training years.

test years	Change				Polarity			
	BOW	sLDA	FWD	SemTreeFWD	BOW	sLDA	FWD	SemTreeFWD
	Consumer Staples				Consumer Staples			
2008-2010	0.1015	0.0774	0.1079	0.1426	0.0359	0.0383	0.0956	0.1054
2011-2012	0.1663	0.1203	0.1664	0.1736	0.0938	0.0270	0.1131	0.1285
5 years	0.1274	0.0945	0.1313	0.1550	0.0590	0.0338	0.1026*	0.1147*
	Information Technology				Information Technology			
2008-2010	0.0580	0.0585	0.0701	0.0846	0.0551	0.0332	0.0697	0.0763
2011-2012	0.0894	0.0681	0.1076	0.1273	0.0591	0.0516	0.0764	0.0857
5 years	0.0705	0.0623	0.0851	0.1017	0.0567	0.0405*	0.0723*	0.0801*
	Telecommunication Services				Telecommunication Services			
2008-2010	0.1501	0.1615	0.1497	0.2409	0.0402	0.0464	0.0821	0.0745
2011-2012	0.2256	0.2084	0.2191	0.4009	0.0366	0.0781	0.0611	0.0809
5 years	0.1803	0.1803	0.1774	0.3049	0.0388	0.0591	0.0737*	0.0770*

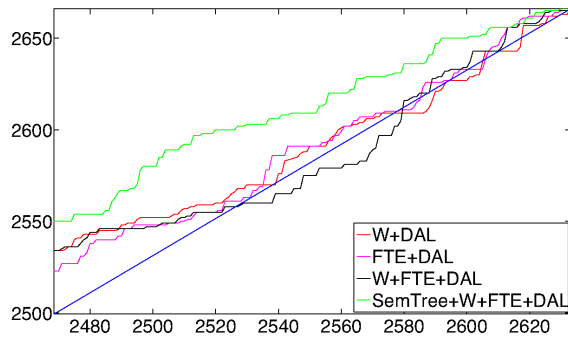
Table 7.4: Average MCC for the change and polarity tasks by feature representation, for 2008-2010; for 2011-2012; for all 5 years and associated p-values of ANOVAs for comparison to BOW.



(a) Raw SVM: Full ROC.



(b) Raw SVM: Head ROC.



(c) Raw SVM: Tail ROC.

Figure 7.5: ROC curves for the polarity task.

Data Representation	P@10	P@20	P@50	P@100	MRR	DCG	PNorm64
positive class (increase of price)							
W+DAL	0.7	0.5	0.52	0.46	0.354	298.167	7.31E+220
FTE+DAL	0.6	0.45	0.38	0.45	0.355	298.245	7.87E+220
W+FTE+DAL	0.8	0.45	0.44	0.46	0.354	298.189	7.90E+220
SemTree+W+FTE+DAL	0.5	0.5	0.54	0.55	0.357	298.414	6.46E+220
negative class (decrease of price)							
W+DAL	0.6	0.55	0.54	0.46	0.350	294.502	3.14E+220
FTE+DAL	0.6	0.75	0.54	0.49	0.351	294.594	2.87E+220
W+FTE+DAL	0.8	0.6	0.54	0.51	0.351	294.530	3.08E+220
SemTree+W+FTE+DAL	1.0	0.75	0.68	0.63	0.353	294.777	1.87E+220

Table 7.5: Evaluation that concentrates on positive and negative predictions by Precision@TopK, DCG, MRR, and PNorm (lower is better).

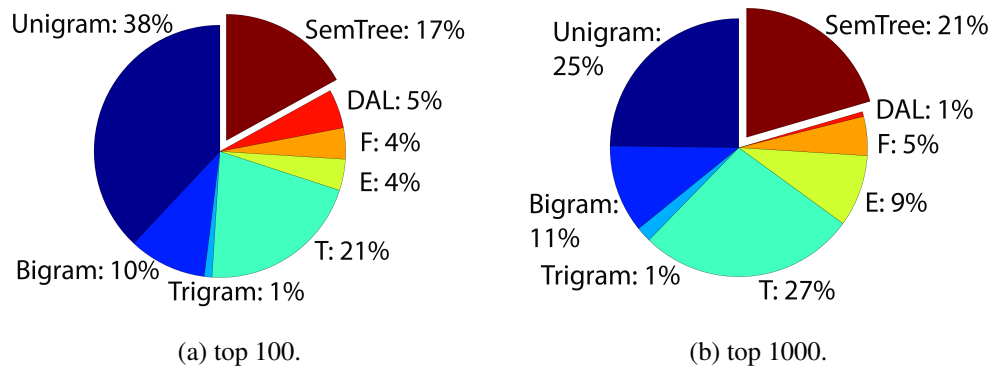


Figure 7.6: Ratio of feature types at top 100 and top 1000 ranked list by information gain for 2010 polarity prediction.

Recall the hypothesis that there exist differences in semantic frame features across sectors. This shows up as large differences in the strength of features across sectors. More strikingly, the same feature can differ in polarity across sectors. For example, in consumer staples, (EVAL-UEE(*Judgment\_communication*)) has positive polarity, compared with negative polarity in the information technology sector. The examples we see indicate that the positive cases pertain to aggressive retail practices that lead to lawsuits with only small fines, but whose larger impact benefits the bottom line. A typical case is the sentence, *The plaintiffs accused Wal-Mart of discriminating against disabled customers by mounting “point-of-sale” terminals in many stores at elevated heights that cannot be reached.* Lawsuits in the IT sector, on the other hand, are often about technology patent

disputes, and are more negative.

*SemTree* features capture the differences between semantic roles for the same frame, and between the same semantic role in different frames. For example, the PERCEIVER\_AGENTIVE role of the *Perception\_active* frame contributes to prediction of an increase in price, as in *R.J. Reynolds is watching this situation closely and will respond as appropriate*. Conversely, a company that fills the PHENOMENON role of the same frame contributes to prediction of a price decrease, as in *Investors will get a clearer look at how the market values the Philip Morris tobacco businesses when Altria Group Inc. “when-issued” shares begin trading on Tuesday*. When a company fills the VICTIM role in the *Cause\_harm* frame, this can predict a decrease in price, as in *Hershey has been hurt by soaring prices for cocoa, energy and other commodities*, whereas filling the VICTIM role in the *Defend* frame is associated with an increase in price, as in *At Berkshire’s annual shareholder meeting earlier this month, Warren Buffett defended Wal-Mart, saying the scandal did not change his opinion of the company*.

We compare the ranking performance of using all features against other combinations of features without *SemTree* features. Figure 7.5a illustrates the receiver operating characteristic (ROC) curves of the full ranked list from different data representations. It also presents the Area Under the ROC Curve (AUC) that corresponds to each representation. As can be seen, the representation with *SemTree* features has higher AUC scores than the others. Its curve starts out slightly better and more stable, neck-and-neck with the other three curves at the beginning, and gradually outperforms the others all the way to the bottom of the ranked list. Figure 7.5b and 7.5c zoom in to the head and tail of the full ranked list.

The head of the ranked list is associated to the positive class (increase of price) and the tail of the list is associated to the negative class (decrease of price). The predictions at these extremes are more important than at the middle of the ranking. The second to the fifth columns of Table 7.5 provide the precision at top K for both classes. For predicting the positive label, W+FTE+DAL correctly captures 8 instances from its top 10 items, which is the best among all methods; while *SemTree* features starts to lead the performance after P@20. Prediction on the negative class is generally better than prediction on the positive class. In 3 out of 4 cases, *SemTree* features are 20% better than the second best method. We quantify the performance at the two ends of the bipartite ranked list by reporting mean reciprocal rank (MRR), discounted cumulative gain (DCG), and PNorm scores

[Rudin, 2009]. MRR and DCG are two weighted versions of AUC that favor the top (or the bottom) of the list. Higher values are preferred. PNorm score corresponds to the loss of the  $l_p$ -norms objective function, where  $p$  controls the degree of concentration to the top (or the end) of the ranked list. Lower values are preferred. As can be seen in Table 7.5, the proposed method has better ranking performance for these metrics.

For feature analysis, we compare the ratios of feature types by their discriminative power. As shown in Figure 7.6, *SemTree* features represent 21% of the top 1000 features ranked by information gain for polarity classification in 2010. This is representative for the other classifiers as well.

## 7.6 Graph Space Results and Discussion

This experiment investigates the potential of a variation of semantic graph representations for large-scale semantic analysis. Because the WL kernel facilitates exploration of a wide range of features that differ regarding the neighborhood of the graph, the first phase of the experiment tests the efficacy of different stepsizes for each of the eight semantic graph variants (undirected and directed versions of vanilla *SemGraph*, *SemDepGraph*, *SemLexGraph*, and full *OmniGraph*). Stepsizes from 0 to 3 are used here to provide a more direct comparison to the *SemTree* representation. Paths from the root node of *SemTree* loosely correspond to paths of up to stepsize 3 from the designated entity nodes in vanilla *SemGraph*. The WL kernel, however, considers the neighborhood three steps from all nodes, not just the designated entity nodes.

The second experiments assess average performance across companies in a sector, using the best stepsize identified for each pairing of an *OmniGraph* with a given company in the first phase. The goal is to identify which configurations of *OmniGraph* perform best in a sector. Both phases evaluate with leave-one-out cross-validation.

Our current results in Table 7.6 show the advantage of *OmniGraph*'s ability to efficiently capture a wide range of semantic dependencies, as well as the ease of testing different extents of the graph around the nodes of interest. Table 7.6 shows the undirected version of the 4 variants of *OmniGraph*, for a few companies from the *Energy* sector. Numbers in boldface identify the best performing stepsize for each *OmniGraph* variant; the underlined values are the best performance across all variants for a given company. For example, the best performance for *ConocoPhillips (COP)* for each variant

uses  $h=3$ , and *SemDepGraph* performs best among the 4 variants of *OmniGraph*. In a majority of cases there is improvement after including at least 1-degree neighbors, and sometimes the best performance is at  $h=1$  or  $h=2$ . No single stepsize consistently performs best across companies, or across *OmniGraph* variants. In a statistical test for all companies in this sector, including at least 1-degree neighbors performs significantly better than using only 0-degree neighbors.

Table 7.7 presents Phase II results that test which *OmniGraph* variant performs best for a sector. Numbers in boldface identify which of the 8 variants has the best mean accuracy for the sector. T-tests that compare the means of each *OmniGraph* to the baseline indicate that in 4 out of 8 sectors, at least one *OmniGraph* variant has significantly better accuracy than the baseline. Not shown in the table, we observe that the benchmarks, *BOW* and *SemTreeFWD*, never beat the baseline. With a higher accuracy in the majority of cases, no single variation of *OmniGraph* consistently significantly outperforms the baseline. More expressiveness representation does not always lead to higher accuracy. Including syntactic dependency information alone (*SemDepGraph*) helps more often than including lexical information alone (*SemLexGraph*): *SemDepGraph* leads to higher mean accuracy in 3 of the 8 sectors, while *SemLexGraph* accuracy is superior in only 1 of the sectors. T-tests to compare mean accuracy of *SemDepGraph* and *SemLexGraph* indicate that *SemDepGraph* is significantly better in 4 of the 16 cases (8 sectors, directed versus undirected graphs), and vanilla *SemLexGraph* is never significantly better than *SemDepGraph*.

The directed versions of *OmniGraph* achieve the best mean accuracy in 3 of 8 sectors, and in one of these cases (GICS15: Energy), the difference is statistically significant. One advantage to the directed versions is efficiency. They reduce the kernel computation asymptotically by a half, since they only allow the flow of information to pass edges through one direction.

Ticker	Baseline	Vanilla SemGraph			SemDepGraph			SemLexGraph			OmniGraph		
		h=0	h=1	h=2	h=3	h=0	h=1	h=2	h=3	h=0	h=1	h=2	h=3
BHI	53.01	48.93	55.25	<b>56.28</b>	55.77	48.93	<b>50.58</b>	50.41	50.58	50.91	51.90	<b>52.56</b>	52.23
COP	53.20	53.31	55.99	56.16	<b>56.98</b>	53.31	56.31	56.62	<b>57.10</b>	51.58	54.73	54.73	<b>55.05</b>
CVX	50.22	52.92	54.61	57.24	<b>57.68</b>	52.92	<b>54.86</b>	53.78	54.64	51.19	<b>52.48</b>	51.84	51.40
...	...	...	...	...	...	...	...	...	...	...	...	...	...
OXY	55.52	53.48	<b>53.63</b>	48.90	49.21	53.48	54.04	54.04	<b>55.99</b>	54.87	53.76	<b>55.15</b>	55.15
VLO	53.99	51.14	<b>53.32</b>	49.67	48.51	51.14	55.08	<b>55.54</b>	54.93	50.99	<b>54.32</b>	53.57	53.26

Table 7.6: A breakdown of performance by stepsizes ( $h$ ) using WL graph kernels for 4 variants of *SemGraph*. It shows the leave-one-out accuracies for some sample companies in the Energy sector.

GICS	Sector	Baseline	directed	Vanilla SemGraph	SemDepGraph	SemLexGraph	OmniGraph
10	Energy	53.95±3.36	n	54.94±5.71	<b>55.93</b> ±5.60*	55.71±6.16*	55.80±5.89*
			y	54.69±5.09	55.61±5.50	55.10±5.80	55.51±5.57
15	Materials	55.00±2.88	n	56.20±4.07	55.08±4.47	55.34±5.66	55.28±5.69
			y	<b>57.07</b> ±4.30*	55.29±4.09	55.10±6.42	55.11±6.30
20	Industrials	54.25±3.85	n	55.24±6.21	54.96±5.54	54.35±5.15	54.41±5.36
			y	54.94±6.17	<b>55.32</b> ±5.41	53.94±5.36	54.58±5.18
25	Consumer Discretionary	54.32±4.18	n	54.52±5.96	55.66±7.19	55.56±4.28*	<b>55.67</b> ±4.41*
			y	54.27±6.26	55.46±5.66	55.36±4.58*	55.40±4.77*
30	Consumer Staples	54.85±3.24	n	55.74±4.98	53.97±3.88	53.14±6.06	53.21±5.71
			y	<b>56.13</b> ±4.95	54.35±3.39	52.67±5.54	52.83±5.63
35	Health Care	56.44±4.51	n	55.61±6.62	<b>57.89</b> ±5.39	55.32±5.31	55.29±5.39
			y	55.51±6.70	57.55±5.54	55.33±5.49	55.48±5.38
45	Information Technology	53.95±4.07	n	<b>56.18</b> ±4.50*	54.07±5.03	53.81±6.06	53.76±6.15
			y	55.59±4.16*	54.10±3.98	53.55±6.20	53.77±5.91
55	Utilities	53.82±2.75	n	54.87±6.28	54.03±4.53	<b>55.16</b> ±5.62	55.00±5.48
			y	54.10±5.47	54.59±5.08	55.01±5.51	54.68±5.33

Table 7.7: The means and standard deviations of the leave-one-out accuracy over the companies in each of the eight GICS sectors. The performance of all variations of *OmniGraph* are shown. Boldface values are the best performance across different *OmniGraphs*. \* indicates a p-value<.05 compared to baseline.



## 7.7 Company Mention Detection

The goal of this task is to study the performance of named entity recognition (NER) for company mention detection in financial news analytics. The two issues we address are 1) to resolve variant names to the same company (e.g., *Eli Lilly and Company*, *Eli Lilly*, *Eli Lilly & Co.*, *Lilly & Co.*), and 2) to resolve coreferent expressions consisting of noun phrases and pronouns (e.g., *Eli Lilly and Company is an American global pharmaceutical company with headquarters in Indianapolis, Indiana. The company also has offices in Puerto Rico and 17 other countries. Their products are sold in 125 countries. It was founded in 1876.*). We refer to this task as *company mention detection*.

Improved company mention detection will not necessarily improve price prediction from news. This is an extremely challenging prediction problem with many confounding factors. For example, news items that provide novel information about a company potentially have more impact on price than news items that provide old information. Accurate company mention detection might incorporate a higher proportion of sentences that provide old information, which could hurt rather than benefit prediction of price change. Given the complexity of factors involved in testing whether more accurate company mention detection improves prediction of stock price change, it is possible that results would vary, depending on the type of feature representation used. To make our test more general, we use our same framework to compare alternative document representations as described in the previous sections. Because this framework compares several kinds of vector and tree space representations, it serves as a more general test of the impact of improved company mention detection.

One of the challenges in mining financial information from news is that the domain of publicly traded corporate entities is extremely heterogeneous. For example, the features that prove predictive vary markedly across sectors, and can even predict opposite direction of price change in different sectors, such as retail versus industrials. It is also well known that the performance of NLP techniques varies across domains. Domain adaptation has been addressed in parsing [Ravi *et al.*, 2008; McClosky *et al.*, 2010; Roux *et al.*, 2012] and language modeling [Bulyko and Ostendorf, 2003; Sarikaya *et al.*, 2005]. Sensitivity to domain is undoubtedly true as well of NER and coreference. This suggested to us that to evaluate the effect on performance of existing NLP tools for improving company mention detection, it is important to assess performance sector by sector. We find that

<p><b>Company name:</b> Baker Hughes Inc</p> <p><b>Ticker:</b> BHI</p> <p><b>Company divisions:</b> Baker Hughes Drilling Fluids, Baker Oil Tools, Baker Petrolite, etc.</p> <p>—————<b>Example sentence 1</b> (a company found by named entity recognition) —————</p> <p>&lt;company ticker='BHI' type='SP500' sector='energy'&gt;Baker Hughes Inc&lt;/company&gt; lowered estimates in mid-July to \$1.12-\$1.14 per share.</p> <p>—————<b>Example sentence 2</b> (company divisions found by named entity recognition) —————</p> <p>Wall, 54, comes from &lt;company ticker='BHI' type='SP500' sector='energy'&gt;Baker Hughes&lt;/company&gt;, where he served since 2005 as group president, completion &amp; production, responsible for the combined activities of &lt;company ticker='BHI' type='SP500' sector='energy'&gt;Baker Oil Tools&lt;/company&gt; and &lt;company ticker='BHI' type='SP500' sector='energy'&gt;Baker Petro-lite&lt;/company&gt; divisions.</p> <p>—————<b>Example sentence 3</b> (company found by coreference resolution) —————</p> <p>&lt;company ticker='BHI' type='SP500' sector='energy'&gt;Baker Hughes&lt;/company&gt; said &lt;company ticker='BHI' type='SP500' sector='energy'&gt;it&lt;/company&gt; supplied products to customers in Myanmar. ... Although &lt;company ticker='BHI' type='SP500' sector='energy'&gt;it&lt;/company&gt; did not have an office or operations there, &lt;company ticker='BHI' type='SP500' sector='energy'&gt;it&lt;/company&gt; was constantly reviewing &lt;company ticker='BHI' type='SP500' sector='energy'&gt;its&lt;/company&gt; presence in nations around the globe.</p>
---

Figure 7.7: Example company and news sentences.

extension of the NER component of our framework and integration of a coreference toolkit dramatically improves recall, but much more so for one sector in particular. Manual assessment of samples of the data suggests that precision remains high. The impact on prediction, however, is not uniform. Predictive accuracy improves primarily for one of the three sectors, using the more expressive tree space representation. Improving prediction is not necessarily dependent on the number of mentions captured, but rather on the quality of the content surrounding company mentions.

### **Motivation**

Company mention detection is a challenging task. Consider the example in Figure 7.7. *Baker Hughes Inc* is a company that provides oil and gas services in the *Energy* sector. Example sentence 1 mentions the full name of the company and an exact match can identify it. The challenges occur when companies mentioned in the articles are referred to by a more abbreviated version of their full name, such as *Baker Hughes* or *Baker*, as in example sentence 2. Further problems lie in the fact that some of these abbreviated mentions name other entities, such as a person, or are generic words, such as the word *baker*, when it introduces a person of that occupation. We had to consider if increasing the recall to capture these cases would outweigh the negative effect of a decrease in precision. Accordingly, we looked at how frequently abbreviated name strings are in fact used to refer to a company versus a different entity. Additionally, there are instances where sub-branches of a company are mentioned, and it is questionable as to whether these are important instances to capture. *Baker Hughes*, in example sentence 2, has divisions *Baker Oil Tools* and *Baker Petrolite*, which are mentioned in the same news article, but an exact match by full name cannot capture these mentions. The question of whether news reports about subsidiary units affect the main company's price is a complex one that we do not address here.

Further improvement of company mention detection requires coreference resolution, especially to detect mentions in different sentences, as shown in example sentence 3 of Figure 7.7. Coreference resolution was not used in many previous studies on financial news analytics, including [Rosenfeld and Feldman, 2007; Feldman *et al.*, 2011a]. We found that the Stanford CoreNLP coreference parser [Lee *et al.*, 2013], a state-of-art coreference resolution toolkit that works well on the CoNLL Shared Task, does not lead to good results when directly applied. It introduces many mention chains that are irrelevant to the company entities, and some chains contain heterogeneous noun phrases that are not appropriate for our company mention annotation task. However, it has a modular design that

GICS	Sector	$\mathcal{C}$	$\mathcal{N}$	$\mathcal{S}$	$\mathcal{T}$
10	Energy	40	5,373	109,277	2,014,085
15	Materials	26	2,295	53,595	953,133
20	Industrials	58	8,325	238,570	3,780,129

Table 7.8: Description of news data for company mention detection.

supports relatively easy re-design, as described below.

### Related Work

This experiment on company mention detection focuses on improving the processing pipeline to improve the overall knowledge discovery framework for financial news. Capturing named entities is essential for making accurate predictions because we rely on named entity recognition to select company relevant news information for price modeling. Named entity recognition is a major area of interest in text mining. A large resource that supports this task is the Heidelberg Named Entity Resource, a lexicon that links many proper names to named entities [Wolodja Wentland and Hartung, 2008]. It is not used in our study because its coverage is limited: it fails to capture enough mentions for our targeted company list, which is based on the S&P 500. As a result, we require a more general and comprehensive method. In addition to named entity recognition, we also incorporated a coreference resolution step to further improve the performance of text mining procedure. There are coreference parsers that use various approaches in attempts to attain optimal performance. The coreference resolution model that our method builds on is the Stanford CoreNLP parser [Manning *et al.*, 2014]. Named entity recognition and coreference resolution are the two key components in our company mention detection task. We leverage state-of-art tools to maximize compatibility and stock market predictability for the financial news domain.

### Data

Our company mention detection experiment relies on a subset of our financial news dataset for the year 2007 on the first three sectors in GICS: 40 companies in GICS 10 of *Energy* such as *Hess* and *Exxon Mobile*, 26 companies in GICS 15 of *Materials* such as *Du Pont*, and 58 companies in GICS 20 of *Industrials* such as *Boeing* and *General Electric*. Table 7.8 describes our data.  $\mathcal{C}$  is number of companies in each sector;  $\mathcal{N}$  is the number of news items;  $\mathcal{S}$  is the number of sentences; and  $\mathcal{T}$  is the number of words.

### Framework

We rely on a framework introduced in the previous section to test the effectiveness of company mention detection. This framework to capture news impact on the financial market consists of three main components, as shown in Figure 7.8: (1) text processing, (2) data instance formation, and (3) model learning and evaluation. In the text processing component, a four-stage NLP pipeline is used. The title and full text of the news article are first extracted from the HTML documents from Reuters News Web Archive. The sentence segmentation stage splits the full text into sentences. The company mention detection stage then identifies if any company of interest is mentioned in the sentence. In this study, we focus on a finite list of companies in the S&P 500.

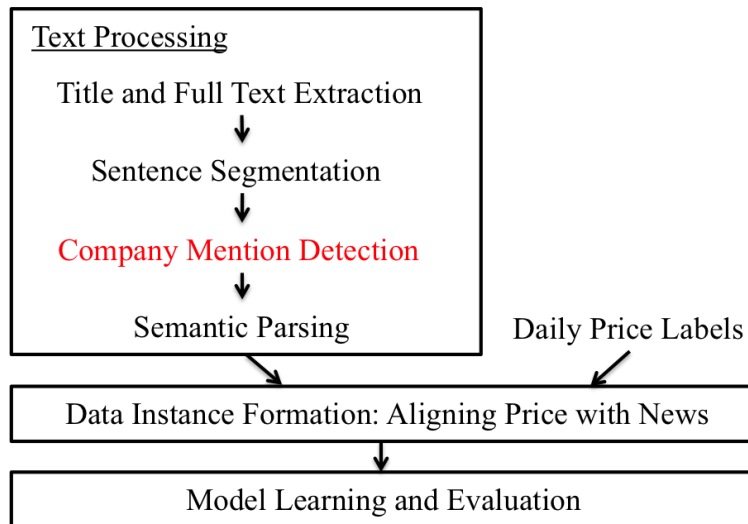


Figure 7.8: Framework of the text mining on financial news for stock market price prediction.

The sentences with at least one S&P 500 company mention are parsed and used for text mining. Therefore, the company mention detection task provides the data foundation for the whole framework. How to improve the coverage of the company mention detection in a way that improves prediction is the main focus in this experiment.

After text processing, we align public available daily stock price data from Yahoo Finance with the textual news data. Recall that the task is to predict the change in price of a company on a date based on the analysis of the preceding day's news. A data instance is all the news associated with a company on a given day, and consists of the companies whose price changed above a threshold between the closing price on the day of the news and the closing price on the following day.

In the learning and evaluation component, rich vector space models are used to test the price prediction performance. These vector space models include bag-of-words models, semantic frame features, and part-of-speech based word affective features. A model that encodes rich structured semantic information, *SemTreeFWD*, is also used for model learning and evaluation. It is an enriched hybrid of vector and tree space models that contains semantic frames, lexical items, and part-of-speech-specific affective features trained with Tree Kernel SVM [Moschitti, 2006].

### **Company Mention Detection**

Our Company Mention Detection module attempts to identify all named entities, variants of these names, and coreferential expressions, then replaces the original strings with a unique identifier. For the identifiers, we use the company tickers, character codes between length of one to five, to identify publicly traded companies. Our initial NLP pipeline used a rule-based method for partial matching on the full company names that only recognized a limited number of the variant names for a company. We have expanded its NER (Named Entity Recognition) rules to capture a much wider range of name variants. We also tested the Stanford CoreNLP coreference parser, and modified it to achieve optimal performance for our domain. This section describes the original and our new NER module, and the changes we made to Stanford coreference parser.

To obtain a lower bound for NER, we used an Exact Match method, defined as matching the exact string to the official names of the S&P 500 companies. This ensures 100% precision, but recall is low. Our initial approach for NER relies on a few conservative rules. These rules focus on the structure of the company names, which can consist of two types of tokens. The words that make up the unique name of the company are the general name elements. The second type are the generic endings, a predefined set of possible suffixes that are optionally included in company names. A generic ending, when included, will be the last token of a company name. It uses the generic endings *Company*, *Corporation*, *Incorporation*, and *Limited*, as well as their abbreviations.

Our initial NER module applies three rules, Exact Match to the company official name, a rule for the generic endings in the Exact Match, and one for the name elements. The second rule applies if there is a generic ending: the program substitutes, one at a time, each generic element in our predefined list for the original generic element, and finally a null element, and searches for each of these new candidate name strings in the text; note that if the null element is substituted, then the new search string consists only of a sequence of name elements with no generic ending. The third rule,

which triggers after the second, truncates the sequence of name elements by iteratively removing the last name element unless the sequence of name elements is length two. After each truncation step, the second rule is re-applied. The process terminates at the first word of a company name.

The proposed Company Mention Detection module incorporates the initial NER module described above, but extends the set of rules so that it does not terminate when the sequence of name elements is length one. Through random sampling and visual inspection, we found that it would be beneficial to include the first word. To maintain high precision, we hard-coded rules for companies where there was a strong possibility that the first token of their names could be mistaken for another entity.

The proposed Company Mention Detection module also incorporates the Stanford CoreNLP parser, which outputs lists of entities that corefer, called coreference chains [Manning *et al.*, 2014]. The Stanford parser was trained on various corpora where the average F-measure was about 60%, which is considered a high score for this task. Furthermore, this parser was intended to be easy for others to modify, either by removing or adding methods to capture coreference patterns. Initially, the Stanford parser seemed ineffective for our dataset due to some inaccuracies in the results. It captured many more instances than it should have, thus decreasing precision. By observing the list of entities in the coreference chains, we noticed that there were some incorrect linkings. First, distinct companies were sometimes linked with each other, such that an incorrect ticker was assigned to one of the companies. Second, the parser captured predicate nominative instances, which are not relevant for our purposes. Third, there were general incorrect linkings between company names and other words in the text.

To address these issues, we re-structured the components of the Stanford CoreNLP coreference parser. The original algorithm goes through ten passes, or sieves, to capture different kinds of coreference phenomena for each iteration [Lee *et al.*, 2013]. By exploring the sieves in the coreference toolkit, we were able to identify the ones causing problems in our data, and to manually tune the parser to meet our needs. The three passes that decreased the accuracy of the mention detection algorithm are called Precise Constructs, Strict Head Match 3 and Relaxed Head Match. There are a few rules incorporated into Precise Constructs, but the main one causing issues in our data was the predicate nominative condition, which, when capturing an entity, also captures the text following a linking verb [Lee *et al.*, 2013]. For example, a sentence that mentions the *ConocoPhillips* company

says, *ConocoPhillips is an international, integrated petroleum company with interests around the world*. Precise Constructs gives the output *ConocoPhillips is ConocoPhillips*.

Strict Head Match 3 removes a word inclusion constraint used in Strict Head Match 1, where all the non-stop words of one entity must match the non-stop words that appear in the previous one. By removing this sieve and thereby imposing this constraint, our program avoids generating incorrect linkages between entities. Strict Head Match 3 removes this constraint since the score for the dataset the Stanford team tested it on improved. Relaxed Head Match allows any word in the main entity to match with entities in other coreference chains. As a result, for the company *Air Products*, the original algorithm incorrectly recognized *these products* to be the company entity. Once these three sieves were eliminated, we observed a significant improvement.

The passes that remained in the coreference parser include Speaker Identification, Exact String Match, Relaxed String Match, Strict Head Match 1, Strict Head Match 2, Proper Head Word Match and Pronoun Match. The Speaker Identification sieve detects the speakers in the text and captures any pronouns that refer to them. In Exact String Match, the parser captures the exact string of entities, similar to the idea of our Exact Match method, but with the additional property of including modifiers and determiners. Relaxed String Match removes the text following the head words of two entities, and links them together if the remaining strings match. Strict Head Match 1 uses the heads of the entities and imposes constraints to determine if the mentions are coreferent. Strict Head Match 2 eliminates a restriction used in Strict Head Match 1, where in this property, modifiers in one entity must match the modifiers in the previous entity in order to be linked together. Proper Head Word Match links proper nouns that have the same head word, but also has specific restrictions imposed on these entities. Pronoun Match focuses on pronominal rules and imposes agreement constraints to capture the entities that are compatible. These seven sieves [Lee *et al.*, 2013] provided the results we needed for capturing additional correct instances.

### **Experiment**

Before conducting our experiment with the Company Mention Detection module, we did some probes on the data to shape our expectations for performance gains. Taking a randomly selected company, and ten randomly selected documents that mention the company, we counted how many company mentions were captured by each of the three methods: *Exact Match*, the *Initial NER* and the proposed *Company Mention Detection (CMD)*. Percentage results for the 54 mentions this



Methods	Precision	Recall	F-measure
Exact Match	100.0%	17.0%	29.0%
Initial NER	100.0%	57.4%	72.9%
CMD	90.0%	76.6%	82.8%

Table 7.9: A manual evaluation for company detection in a preliminary experiment.

yielded are displayed in Table 7.9. As shown, *CMD* yielded greatly improved recall at a reasonable sacrifice in precision, and an overall increase in F-measure of 13.6%, compared to the *Initial NER*. Interestingly, the incorrect instances for *CMD* were not entirely wrong: they all referred to units within the company. We count them as incorrect, however, because of our focus on predicting stock price for the S&P 500 (parent) companies. As noted above, what happens to one unit of a company may not necessarily affect public perception of the company as a whole. We, therefore, do not regard sub-companies as correct instances for the purposes of our experiment.

The Exact Match method has a very low F-measure since it only captures the full name of a company. Except for the first time it is mentioned in a news article, a company is usually not referred to by its full name. Instead, variations of company names are frequently used. Clearly, the Initial NER method far outperforms this baseline, yet leaves much room for improvement in recall.

As described in section 7.7, *CMD* further expanded the NER so as to search for abbreviated name strings that include only the first word of the full named entity string of the companies. For the company *Baker Hughes Inc.*, this would lead to the inclusion of mentions by the single name *Baker*. Although in the general case, this could introduce imprecision, if a document already contains the

GICS	Initial NER	CMD	Increase
10	8,646	11,252	30.14%
15	5,445	6,336	16.36%
20	15,286	17,865	16.87%
Total	29,377	35,453	20.68%

Table 7.10: Counts of company mentions by sentence.

full company name, it is likely that use of the first name token in the full name (e.g., *Baker*) would be a company mention. In addition, *CMD* also captures many coreferential expressions for company mentions. For example, one article says, *Baker Hughes said it supplied products to customers*; where the original NER rules capture *Baker Hughes*. *CMD* also captures *it*. As shown in Table 7.10, *CMD* captures many additional instances of company mentions. This also leads to some gains in stock price prediction, as will be reported in the next section.

The experiment uses as input the data described in Table 7.8 consisting of all the news in three market sectors from Reuters news archive for 2007. Recall, we use the framework described in Section 7.7 because it allows us to test the impact of improved F-measure for *CMD* across multiple document representations. The five document representations we test in the experiment are: 1) *BOW*, which refers to bag-of-words with unigram counts; 2) *BOW (n-gram)*, for BOW with unigram, bigram and trigram counts; 3) *FW* which is like *BOW (n-gram)* but also includes Frame Semantic elements (see next paragraph); 4) *FWD* consists of *FW* plus a prior polarity on words from the Dictionary of Affect in Language (DAL score; see next paragraph); 5) and lastly, *SemTreeFWD*, which is a tree structure that uses the FWD features combined with a tree kernel.

Three of the five document representations make use of features from frame semantics [Fillmore, 1976]. Frame semantics aims for a conceptual representation that generalizes from words and phrases to abstract scenarios, or frames, that capture explicit and implicit meanings of sentences. The three basic feature types from frame semantics are frame name, frame target, and frame element. Each frame is evoked by a frame target, or lexical unit, for example, *sue* or *accuse* evoke the *Judgement Communication* frame, which describes a lawsuit scenario. Its frame elements, or semantic roles, are *Communicator*, *Evaluee*, and *Reason*. *FW* and *FWD* uses bag-of-frames (including frame names, frame targets, and frame elements) features in a vector space representation, while *SemTreeFWD* encodes relational structures between the company entity and the semantic frame features in a tree representation, in addition to *FWD*. The semantic parsing we use to extract frame features is SEMAFOR<sup>3</sup> [Das and Smith, 2011; Das and Smith, 2012], a statistical parser that uses a rule-based frame target identification, a semi-supervised model that expands the predicate lexicon of FrameNet for semantic frame classification, and a supervised model for argument identification.

---

<sup>3</sup><http://www.ark.cs.cmu.edu/SEMAFOR>

GICS	Sector	type	BOW	BOW (n-gram)	FW	FWD	SemTreeFWD
10	Energy	Initial NER	59.94±16.38	61.18±15.43	59.99±14.46	59.05±16.58	64.26±14.95
		CMD	58.54±17.32	61.11±15.34	58.67±15.76	58.44±18.40	<b>64.87±15.04</b>
15	Materials	Initial NER	58.23±15.53	59.74±14.33	62.10±14.24	62.69±15.28	68.62±14.72
		CMD	<b>61.82±15.18</b>	<b>60.63±15.33</b>	<b>63.23±13.71</b>	<b>63.12±15.01</b>	67.18±13.37
20	Industrials	Initial NER	56.70±14.81	55.47±13.86	53.86±13.43	54.29±14.31	57.25±16.88
		CMD	<b>60.13±14.04*</b>	<b>58.19±13.44*</b>	<b>55.37±13.31</b>	<b>55.75±13.54</b>	56.36±18.38

Table 7.11: Averaged test accuracy for each company by sector that uses 80% of the data for training 20% for testing. Boldface identifies a higher *CMD* mean and \* identifies the *CMD* that is significantly better than the *Initial NER* with  $p\text{-value} < 0.05$ .

*FWD* and *SemTreeFWD* contain word affect features based on DAL, the Dictionary of Affect in Language [Whissel, 1989]. It is a psycholinguistic resource designed to quantify the undertones of emotional words that includes 8,742 words annotated for three dimensions: pleasantness, activation, and imagery. We use the average scores, in terms of the three dimensions, for all words, verbs, adjectives, and adverbs in a vector space for feature representation.

The experiments assess the performance of predicting the direction of price change across companies in a sector. Recall that a data instance in our experiment is all the news associated with a company on a given day, and consists of the companies whose price changed above a threshold between the closing price on the day of the news and the closing price on the following day. In this experiment, we use the threshold of 2% that corresponds to a moderate fluctuation. A binary class label  $\{-1, +1\}$  indicates the direction of price change on the next day after the data instance was generated from the news. For each company, 80% of the data is used for training and 20% for testing. We report the averaged accuracy and standard deviation of the test data for both the *Initial NER*, as a benchmark, and our *CMD* on a sector-by-sector basis.

## Results

The experiment addresses two questions: 1) Does *CMD* improve the coverage of company mentions in the domain of interest? 2) Does our *Company Mention Detection* improve accuracy of prediction on the task to identify the direction of price change? Based on our probe of the data where we could manually assess precision (Table 7.9 in section 7.7), we expected a large increase in coverage. Projecting from the results of this manual probe, we assume that an increase in recall

comes with an acceptable (small) degradation in precision. Yet, because there is no gold standard data set, we cannot assess precision of CMD for the full dataset. Prediction accuracy is the true test of performance on the benefit of increased coverage of company mentions using CMD, but is only a very indirect measure of precision. As noted above, stock price prediction from news is a challenging task with a great deal of noise in the input. Results presented here show a substantial increase in coverage, and statistically significant increases in prediction accuracy for some but not all of the experimental conditions.

As background to interpret the results, it is important to consider the relation between the increased number of mentions versus the number of data instances per company, and the differences across sectors in the average number of data instances per company. Again, each data instance consists of all the news for a given company on a given day. Therefore, new data instances will be added only if CMD identifies news for a given company on a day that was not identified before. If new sentences for a given day are identified, however, then we expect that *BOW* and *BOW (n-gram)* are very likely to be enriched, and prediction could improve in these two cases. If new mentions in an existing sentence are identified, this should not improve *BOW* and *BOW (n-gram)* because all the relevant feature positions in the vector (unigram, n-gram) will already have had values, and the values will not change. In contrast, if new mentions occur not in the same sentence but in new clauses within or across sentences, the representations that use semantic frame parsing (FW, FWD, SemTreeFWD) could be enriched if the new clauses contain words that trigger new frames, and the new mentions fill their roles.

We found that CMD did not increase the number of data instances. This result suggests that if a news item mentions a relevant company, at least one mention will be either an exact match to the full name string, or a near match based on the conservative NER rules. On the other hand, there were substantial gains in the total number of sentences. Table 7.10 reports the absolute numbers of sentences with company mentions from the original NER module compared with those for the Company Mention Detection module. At increases of between 16% and 17%, the Materials and Industrials sectors already show large increases; the increase for the energy sector is nearly double that of the two other sectors. This difference between the GICS 15 and 20 versus GICS 10 reflects the underlying domain differences from sector to sector, which accounts to some degree for the difficulty of the prediction task. We further note that the number of data instances per company

differs substantially across the three sectors. The mean and standard deviation for each sector are as follows, respectively: GICS 10,  $\mu = 24.37$ ,  $\sigma = 15.80$ ; GICS 15  $\mu = 20.80$ ,  $\sigma = 15.52$ ; GICS 20:  $\mu = 16.16$ ,  $\sigma = 18.96$ . Based on these figures, we expect the gains for GICS 15 and 20 to be similar, and the gains for GICS 10 to be larger for the semantic frame representations.

Table 7.11 gives the average accuracy per sector of the CMD combined with the five document representation methods introduced in the previous section. (Note: None of these results significantly beat the baseline accuracy given by the average over the majority class for each company, but the standard deviations for this baseline—as for the results in Table 7.11—are quite high. This does not diminish the comparison of the different representations, and the question of whether CMD can improve performance.) Prediction accuracy improved for the BOW representations. The numbers in boldface are the cases where the average accuracy for CMD is higher than for the original NER, and the cells with an asterisk indicate cases where a t-test of the difference is statistically significant. As shown, the two cases where there is a statistically significant improvement are for the two BOW representations for the sector with the fewest average data instances per company, namely Industrials. When using NER, the BOW representations already had very competitive performance, and CMD increases their performance. This suggests that the new sentences that are identified with CMD add new vocabulary that is predictive. The two vector-based representations with frames also have higher accuracy, but the increase is not statistically significant. For the tree-based representation (SemTreeFWD), the performance degrades somewhat. The performance of the frame-based representations suggests that the new sentences for Industrials do not add new frames, or possibly add new frames that have semantic conflicts with the frames that were found earlier. The same general pattern holds for the Materials sector.

The one case where the SemTreeFWD performance improves is for the Energy sector, but the improvement is not statistically significant. We can only speculate that this sector is the only one where SemTreeFWD shows greater accuracy because this is the sector where the number of additional sentences is substantially larger.

The two questions posed by our experiment can be answered briefly as follows: 1) CMD improves the coverage of company mentions dramatically at the sentence level: the number of additional sentences per sector increases on average by over 20%. This does not, however, increase the number of data instances; 2) CMD has a statistically significant impact on predictive accuracy

only for the Industrials sector, for the two BOW representations. In the next section we discuss the ramifications of these results.

### **Conclusion on Company Mention Detection**

Evaluation of coreference performance generally involves assessment of the accuracy of coreference as an independent module. Here we provide an evaluation of coreference as an independent module (intrinsic), and as part of an end-to-end system that aims at a real world prediction task (extrinsic). The results presented in the preceding section provide a very dramatic and concrete demonstration that large gains for coreference as a stand-alone module do not necessarily result in system gains. They also demonstrate the importance of considering the overall integration of information for data representation.

Of the fifteen conditions in Table 7.11, the two conditions where we find statistically significant improvements from CMD pertain to the two data representations that are relatively less rich, *BOW* and *BOW (n-gram)*, for the sector with the fewest data instances. There are marginal improvements that are not statistically significant for FW and FWD, and a degradation for SemTreeFWD. This indicates to us that the new sentences added for the Industrials sector add new features to the BOW feature vector, but do not add as much in the way of frame features. Continuing with this sector, the differences between the five document representations are not as great for NER as they are with CMD, and the unigram BOW representation in the CMD condition ends up with the highest accuracy for the ten conditions. The same general trend for the vector representations holds in Materials as for Industrials, but without statistical significance. For Materials, however, SemTreeFWD remains the representation with the highest accuracy among all five.

Energy, which had a much more substantial gain in number of sentences, has a different pattern. There are no gains for the vector based representations. Energy is also the sector with the greatest number of data instances per company. Here we speculate that the addition of new sentences does not add new vocabulary: with such a large number of data instances per company already, vocabulary coverage was perhaps already high. SemTreeFWD shows a small gain in accuracy that is not statistically significant.

In our view, rich semantic and pragmatic data mining for large scale text mining should aim for information that supports more informed decision making, or in other words, is actionable. To summarize the results of the experiment presented here, a substantial increase in coverage for the task of

detecting mentions of relevant entities on a large scale prediction task does not necessarily translate to gains in the actionable value of the information gained. Further, the experiment demonstrates the interdependence of semantic and pragmatic data mining with feature representation, and with the end goals of the data mining task.

## Chapter 8

# GoodFor/BadFor Corpus Analytics

To test the performance of our representation and learning methods on entity driven analytics, we experimented on a recently introduced, publicly available dataset - the GoodFor/BadFor Corpus<sup>1</sup> [Deng *et al.*, 2013], which is part of MPQA. MPQA, multi-perspective question answering, is a corpus that contains news articles from a wide variety of news sources manually annotated for opinions and other private states. The original MPQA dataset is created to facilitate the research on general sentiment analysis and opinion mining, which contains documents of foreign and U.S. news sources that were identified by human searches and by an information retrieval system. The dataset we use in this study is an newly introduced GoodFor/BadFor dataset in MPQA. The annotation of the GoodFor/BadFor dataset investigates whether the event mentioned in a sentence has either positive or negative affect on the the event object. The creation of the dataset is to facilitate the research on fine-grained sentiment analysis that distinguish the opinion holder and the object.

We first introduce the GoodFor/BadFor dataset, then describe two classification tasks related to the two annotation tasks on this corpus. We then present our results followed by discussions.

### 8.1 Introduction

Sentiment analysis is a popular topic and a fast growing area in NLP research. Traditional sentiment analysis is interested in classifying the overall sentiment of a document, where bag-of-words and dictionary based valence scoring are often used. While recent studies are on fine-grained sentiment

---

<sup>1</sup><http://mpqa.cs.pitt.edu/corpora/gfbf/>



analysis, for example, to identify the opinion and the opinion holder and the target object, which requires more of a deeper understanding of the syntactic and semantic level analysis. This work on entity-driven text analytics also contributes to the fine-grained sentiment analysis. Our methods can be applied to the sentiment analysis task that targets at an object of interest, e.g. designated entities, in a sentence, and we make use of lexical, syntactic dependency, and semantic information. In particular, our models even allow us to create different representations for different designated entities for the same sentence, which has been shown successful in financial news analytics in the previous experiment. To test the generality of the methods, we here experiment on a recently introduced, publicly available dataset in sentiment analysis - the GoodFor/BadFor dataset<sup>2</sup> [Deng *et al.*, 2013], which is part of the MPQA corpus.

GoodFor/BadFor dataset is an annotated corpus created for the study of fine-grained opinion mining and sentiment analysis. Two annotation tasks are involved in the GoodFor/BadFor (gfbf) corpus: 1) benefactive/malefactive event annotation, and 2) writer attitude annotation. The benefactive/malefactive task asks annotators to identify the affected entity (the object) and the entity causing the event (the agent), and label either the agent and the event has a positive or negative affect to the object. For ease of communication, the terms GoodFor and BadFor are used for benefactive and malefactive events, respectively. On other other hand, the writer attitude task asks annotators to identify the writer's attitude towards the agents and toward the objects. We derived two binary classification tasks from these two annotation scheme. The first task is to classify the benefactive or malefactive affect on the object, and the second task is to identify the positive or negative writer attitude towards both the agent and the object. The next two sections will provide some example annotations, and introduce how we create our data representation for each of the classification tasks.

## 8.2 Benefactive/Malefactive Identification Task

Our first classification task is to identify whether the agent and the event mentioned in the sentence are benefactive (good for) or malefactive (bad for) to the object. When the corpus is being annotated, annotators are asked to interpret sentences and words with respect to the context in which they appear, and do not take words out of context but just judge them as they are being used in that

---

<sup>2</sup><http://mpqa.cs.pitt.edu/corpora/gfbf/>

particular sentence. Many of the circumstances should be judged by intuition based on the most common fundamental ethical values. There are also several guidelines for the annotation tasks [Deng *et al.*, 2013], and these include:

- To exist is good.

(1) This strategy has already failed in Britain, where [politicians<sub>Agent</sub>] desperate to tame rising health costs created [a National Institute for Health and Clinical Effectiveness<sub>Object←BENEFACTIVE</sub>].

(2) Then [he<sub>Agent</sub>] would help [us<sub>Object←BENEFACTIVE</sub>] develop the creative new ideas that would allow us to work together to face the big issues, not sidestep them.

- To destroy is bad.

(3) By utilizing peer review practices which would not stand muster under standard constitutional law, [hospital and health systems<sub>Agent</sub>] can label anyone a disruptive, unruly or uncooperative physician and destroy [their ability to work<sub>Object←MALEFACTIVE</sub>].

(4) On the other hand, we have [an opposition<sub>Agent</sub>] that wants to get rid of the law - and then dismantle [Medicare and Medicaid<sub>Object←MALEFACTIVE</sub>] along with it," Sebelius said.

- To assist is good.

(5) Despite a lot of noise and confusion over opponents' claims, a quick look at the facts shows that [this reform law<sub>Agent</sub>] is well on its way to helping protect [working families and the middle class<sub>Object←BENEFACTIVE</sub>].

(6) Then [he<sub>Agent</sub>] would help [us<sub>Object←BENEFACTIVE</sub>] develop the creative new ideas that would allow us to work together to face the big issues, not sidestep them.

Annotators are also asked to interpret sentences and words with respect the context in which they appear, and do not take words out of context but just judge them as they are being used in that particular sentence.

To create SemTrees or OmniGraphs for the experiment, we treat the *Object* in the annotated sentence as the designated entity. For example, for sentence *This strategy has already failed in Britain, where politicians desperate to tame rising health costs created a National Institute for Health and Clinical Effectiveness*, the semantic parsing is in Figure 8.1. The SemTree and OmniGraph representation for the target object is illustrated in Figure 8.2 and 8.4. Note that the dependencies among frames in OmniGraph are recovered as shown in Figure 8.3, which is constructed based on the syntactic dependency parsing.

**Sentence:** This strategy has already failed in Britain, where politicians desperate to tame rising health costs created a National Institute for Health and Clinical Effectiveness.

**Benefactive/Malefactive Annotation:**  
 This strategy has already failed in Britain, where [politicians<sub>Agent</sub>] desperate to tame rising health costs created [a National Institute for Health and Clinical Effectiveness<sub>Object←MALEFACTIVE</sub>].

**Frame semantic parse:**  
 [This strategy<sub>Success\_or\_failure.Agent</sub>] has already [failed<sub>Success\_or\_failure</sub>] [in Britain, where [politicians<sub>Desiring.Experiencer</sub>] [desperate<sub>Desiring</sub>] [to [tame<sub>Conquering</sub>] [rising<sub>Motion\_directional</sub>] health [costs<sub>Motion\_directional.Theme</sub>]Desiring.Event] [created<sub>Intentionally\_create</sub>] [a National Institute for Health and Clinical Effectiveness<sub>Intentionally\_create.Created\_entity</sub>]Success\_or\_failure.Goal].

Figure 8.1: Example sentence, its benefactive/malefactive annotation, and the frame semantic parse.

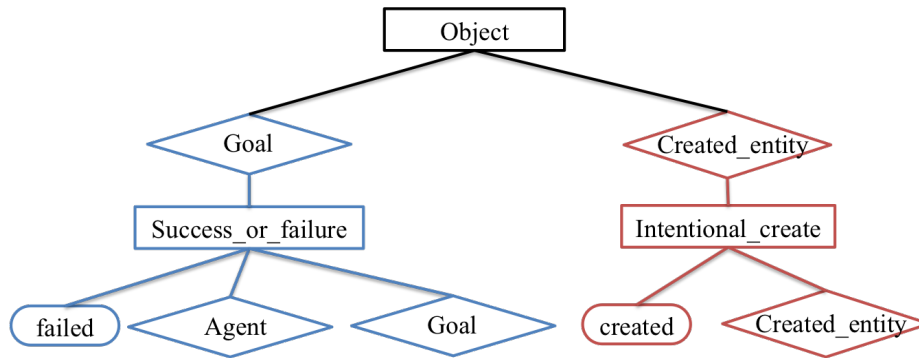


Figure 8.2: SemTree representation for the object *a National Institute for Health and Clinical Effectiveness* of the sample sentence in Figure 8.1.

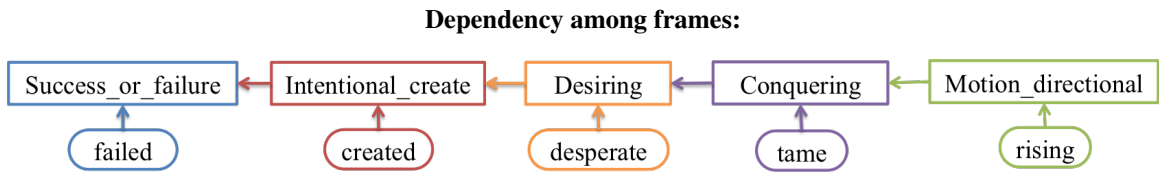


Figure 8.3: The dependencies among semantic frames, which is constructed based on syntactic dependency parsing.

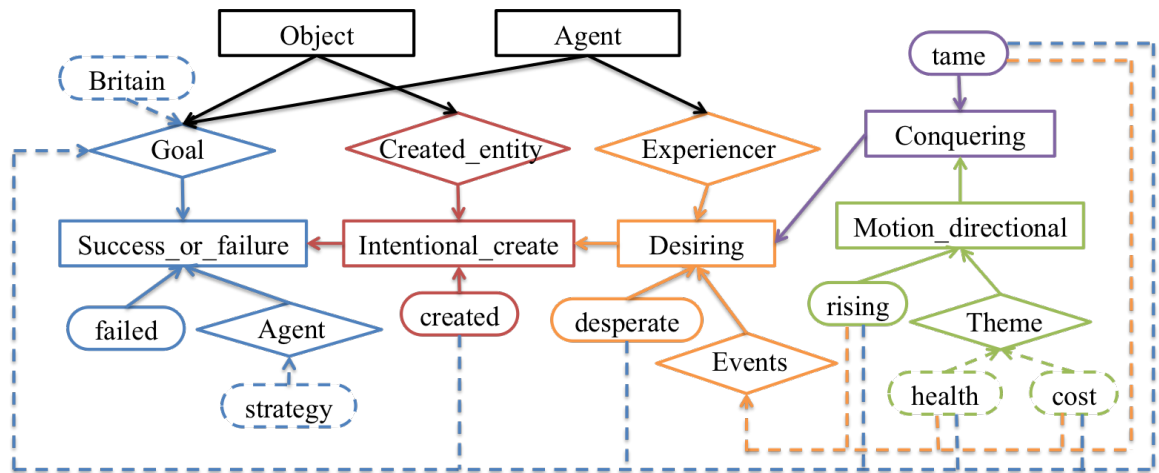


Figure 8.4: OmniGraph representation that includes lexical, dependency, and semantic information for the object *a National Institute for Health and Clinical Effectiveness* of the sample sentence in Figure 8.1.

### 8.3 Writer Attitude Detection Task

The other classification task is to detect the positive or negative writer attitude towards the agent and the object. In the annotation, annotators are asked to mark whether there is a positive or negative attitude of the writer revealed in that particular sentence. Similar to the benefactive/malefactive task, the annotation requires the annotators not to 'over' use their world knowledge, and only find mark those that reveal the speaker's attitude within the sentences.

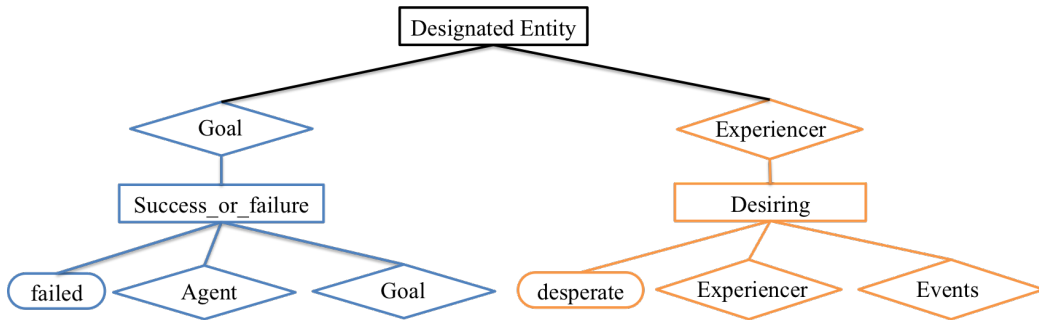
Figure 8.5 shows an example sentence with its annotation for both the agent and the object.

**Sentence:** This strategy has already failed in Britain, where politicians desperate to tame rising health costs created a National Institute for Health and Clinical Effectiveness.

**Writer Attitude Annotation:**  
 This strategy has already failed in Britain, where [politicians<sub>Agent←NEGATIVE</sub>] desperate to tame rising health costs created [a National Institute for Health and Clinical Effectiveness<sub>Object←NEGATIVE</sub>].

Figure 8.5: Example sentence and its writer attitude annotation.

**SemTree Representation for the Agent:**



**SemTree Representation for the Object:**

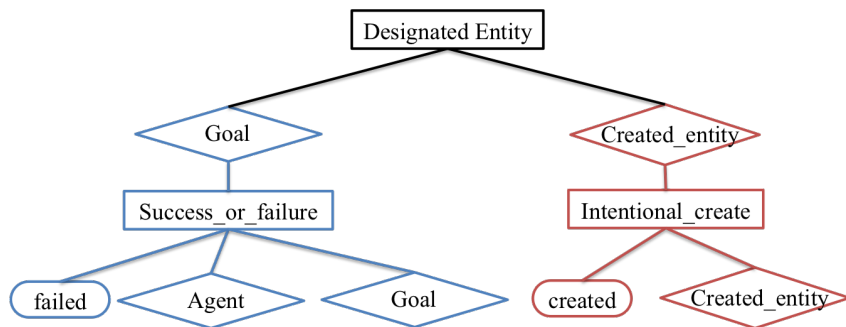
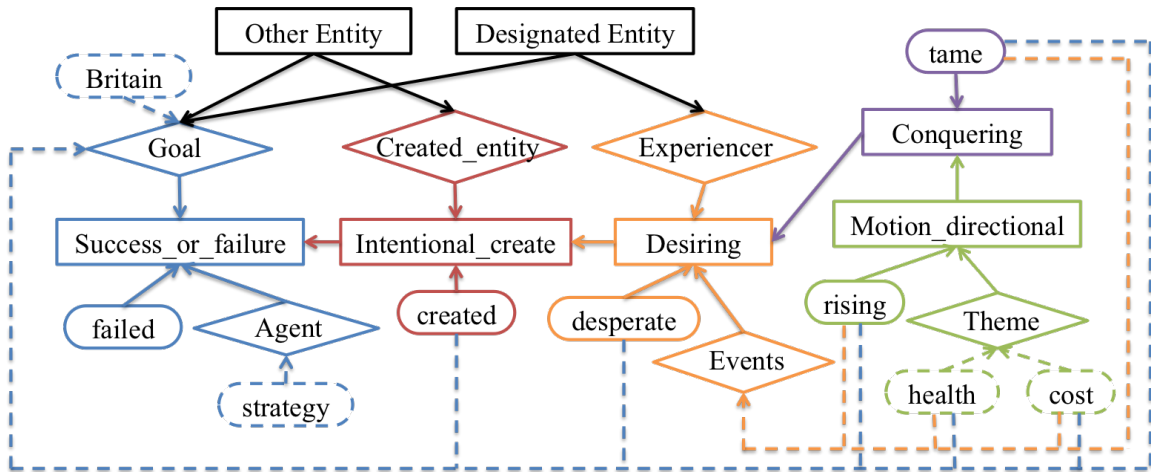


Figure 8.6: SemTree representations for the agent and the object, respectively.

**OmniGraph Representation for the Agent:**



**OmniGraph Representation for the Object:**

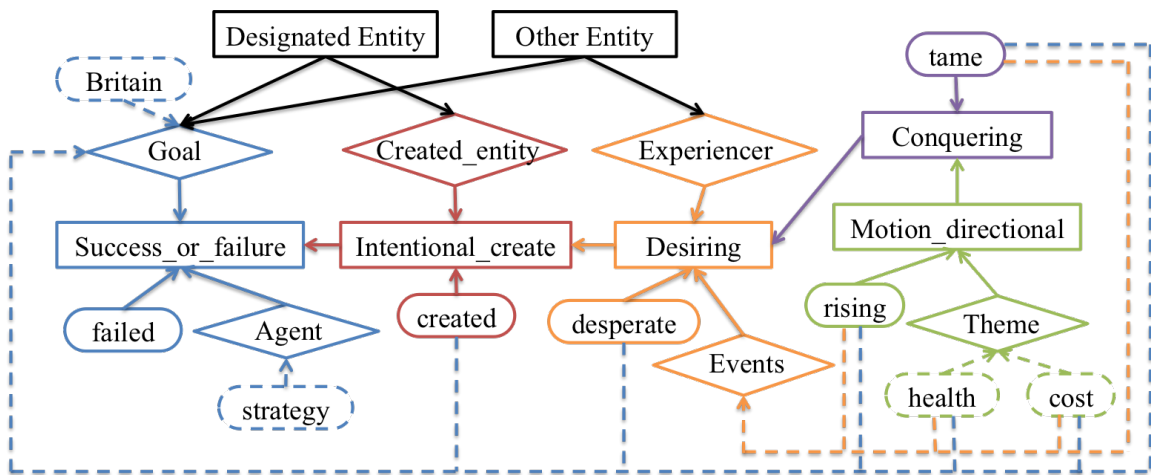


Figure 8.7: OmniGraph representations for the agent and the object, respectively.

## 8.4 Experiments and Results

The experiment includes two classification tasks introduced in the previous sections. Our goal is to test if our structured representation and learning can improve the traditional word based vector space model, and lead to good performance on this fine-grained sentiment analysis task.

We extracted the sentences from the corpus annotation. We ran MST parser for dependency parsing, and SEMAFOR for frame-based semantic parsing. The coverage of semantic frames in this dataset is wide. A total of 604 distinct frames are identified, and they approximately follow the Zipf's law. Figure 8.8 shows the plot of the distribution of the frames, the  $R^2$  of the log fit is 0.856. Table 8.1 shows the top 50 most frequent frames. These frames cover a wide range of scenarios. For example, the *Frequency*, *Quantity*, *Increment* (top 3 frames), and *Cause\_change\_of\_position\_on\_a\_scale* (8th ranked) frames are related to numbering and frequency generalizations. The *Medical\_conditions* (4th), *Education\_teaching* (7th), *Law* (19th) and *Reforming\_a\_system* (34th) frames reflect the topics in this corpus. *Purpose* (16th), *Intentionally\_act* (21st), *Intentionally\_create* (45th), and *Desirability* (46th) are related to behaviors and intentions. We hypothesize that these generalization of the frames and the characterization of the lexical items (frame targets) that evoked the frames are useful to the classification task. We further look into the frame targets (lexical items that evoked the frames) and Table 8.2 shows the frame targets distribution for each of the top 10 most frequent frames. We can see there is a wide variety of lexical items. Recall that some annotation guidelines are relevant to the quantities, e.g. to exist is good (*none* and *never* may have a negative affect), gain is good and loss is bad (*more* and *increase* may be positive while *reduce* and *few* may be negative). We expect the inclusion of semantic frame related features in data representation to be beneficial to the classification task.

In the experiment setup, we use the percentage of the majority class as the baseline, and compare the same five methods in our previous experiment. (1) *BOW* - a vector space model that contains unigrams, bigrams, and trigrams. (2) *DepTree* - a tree space representation where dependency parse of the sentence is encoded into a tree representation. (3) *SemTreeFWD* - an enriched hybrid of vector and tree space model that contains semantic frames, lexical items, and part-of-speech-specific psycholinguistic dictionary-based features, trained with Tree Kernel SVM [Moschitti, 2006]. (4) & (5) - OmniGraph representation trained with Weisfeiler-Lehman graph kernel and Node Edge Weighting graph kernel.

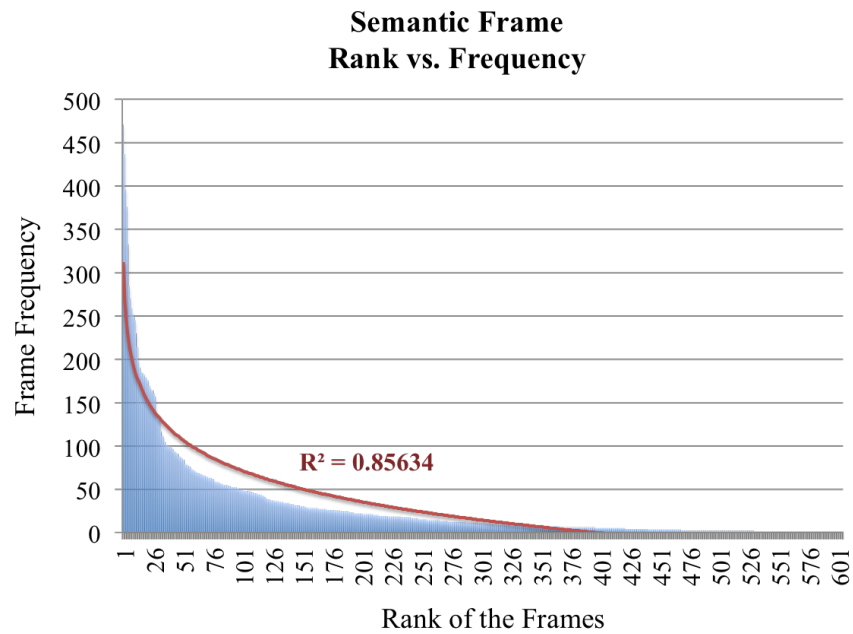


Figure 8.8: Distribution of semantic frames that are identified in the GoodFor/BadFor dataset. The trendline is a log fit, with  $R^2 = 0.856$ .



Rank	Frame Name	Freq	Rank	Frame Name	Freq
1	Frequency	471	26	Commerce_pay	160
2	Quantity	437	27	Businesses	157
3	Increment	376	28	Likelihood	138
4	Medical_conditions	333	29	Grant_permission	132
5	Expensiveness	284	30	Causation	132
6	Statement	270	31	Possession	131
7	Education_teaching	259	32	Topic	116
8	Cause_change_of_position_on_a_scale	252	33	Required_event	111
9	Buildings	249	34	Reforming_a_system	110
10	Capability	245	35	Cure	104
11	Political_locales	230	36	Age	101
12	People	214	37	Removing	100
13	Project	196	38	Observable_body_parts	100
14	Temporal_collocation	190	39	Importance	99
15	Continued_state_of_affairs	186	40	Expertise	99
16	Purpose	184	41	Desiring	97
17	Relational_quantity	183	42	Fields	96
18	Cardinal_numbers	181	43	Relative_time	93
19	Law	179	44	Locale_by_use	93
20	Calendric_unit	176	45	Intentionally_create	91
21	Intentionally_act	175	46	Desirability	91
22	Leadership	169	47	Sufficiency	87
23	Change_position_on_a_scale	166	48	Cause_to_make_progress	87
24	Time_vector	164	49	Protecting	86
25	Assistance	164	50	Measure_duration	85

Table 8.1: Top 50 most frequent frames in GoodFor/BadFor dataset.

Rank	Frame Name (Freq)	Frame Target (Freq)
1	Frequency(471)	not(89), n't(70), also(52), even(48), just(30), every(20), really(12), actually(12), rate(11), annual(10), instead(9), Even(8), Not(8), Instead(8), often(7), else(6), ever(6), easily(5), never(5), annually(4), exactly(4), simply(4), once(3), regular(3), definitely(3), Also(3), Once(2), always(2), typically(2), merely(2), generally(2), forever(2), surely(1), either(1), Never(1), Nevertheless(1), usually(1), ...
2	Quantity(437)	all(63), no(43), many(43), any(26), trillion(22), those(20), number(19), both(18), these(15), millions(15), several(11), No(10), amount(9), All(8), majority(8), nothing(8), hundreds(7), thousands(7), billions(6), fair(5), either(5), These(5), few(4), masses(4), measure(4), a few(4), transparent(4), a lot(4), numbers(3), none(3), percentage(3), size(3), dozen(2), amounts(2), also(2), lots(2), Few(2), None(2), ...
3	Increment(376)	more(125), that(96), other(55), That(22), further(18), additional(17), another(10), others(6), More(4), significantly(4), Other(4), Further(3), Others(2), fewer(2), targeted(2), Medicare-paid(1), nothing(1), something(1), Additional(1), extra(1), supplemental(1)
4	Medical_conditions(333)	health(244), sick(29), cancer(10), illness(8), ill(5), healthy(5), diseases(4), welfare(4), cold(3), syndrome(3), plaguing(2), illnesses(2), flu(2), obesity(2), hospitalizations(2), well-being(1), disease(1), pregnancy(1), hangover(1), intact(1), disabilities(1), exposure(1), headaches(1)
5	Expensiveness(284)	costs(91), cost(35), premiums(35), benefits(19), affordable(17), available(15), spending(13), free(12), cheaper(4), cost-containment(4), deductibles(4), costly(4), expensive(4), expenses(4), free-market(3), controversial(2), expense(2), exorbitant(2), Costs(1), fees(1), pricy(1), unaffordable(1), cost-reduction(1), costing(1), Cost(1), affordability(1), premium(1), attractive(1), overhead(1), government-subsidized(1), benefit(1), deductible(1)

Table 8.2: Frame targets (lexical items that evoked the frames) for the top 10 most frequent frames (part 1).

Rank	Frame Name (Freq)	Frame Target (Freq)
6	Statement(270)	said(42), denying(29), deny(20), claims(16), add(12), added(11), says(11), denied(10), adding(8), announced(7), explain(6), report(6), say(5), claim(4), proposed(4), Adding(4), saying(4), comment(4), warns(3), proposal(3), insist(3), affirmed(3), acknowledge(3), notes(2), observed(2), warning(2), warned(2), explained(2), reported(2), reporting(2), contend(2), declared(2), reports(1), contends(1), maintain(1), invoke(1), semantics(1), claiming(1), States(1), declares(1), proclaimed(1), state(1), credit(1), recounting(1), propose(1), Says(1), talked(1), remarking(1), touted(1), dismissing(1), denial(1), states(1), decrees(1), upheld(1), uphold(1), statement(1), confirms(1), explains(1), caution(1), adds(1), attributed(1), declaration(1), Denying(1), suggested(1), recommends(1), writes(1)
7	Education_teaching(259)	care(157), medical(56), health-care(13), counseling(11), education(2), student(2), educating(2), training(2), Medical(2), tech(1), educational(1), environmental(1), technical(1), learn(1), graduates(1), students(1), teaching(1), teaches(1), graduate(1), educate(1), lessons(1)
8	Cause_change_of_posi- tion_on_a_scale(252)	reduce(51), cut(31), reducing(27), increase(26), cuts(16), raise(12), promote(8), raising(8), push(7), boost(7), increased(6), promoting(5), reduces(4), lowering(4), limiting(4), increasing(4), growth(4), lowers(3), decreasing(3), reductions(2), layoffs(2), increases(2), limited(2), reduced(2), diminish(2), Increasing(1), eroding(1), inflation(1), lessen(1), Boost(1), pushing(1), CUTS(1), Pushing(1), Reducing(1), boosts(1)
9	Buildings(249)	insurance(163), benefits(28), health-insurance(9), hospitals(5), b(5), bonus(4), entitlement(4), salaries(4), housing(3), building(3), hotels(3), bureaucracy(3), hospital(3), shed(2), B(2), houses(2), benefit(1), Insurance(1), buildings(1), hotel(1), grapple(1), Hospital(1)
10	Capability(245)	can(118), could(39), ca(27), able(24), ability(11), would(7), worthy(3), quality(3), eligible(3), accountable(2), responsible(2), potential(2), hard-pressed(2), Can(1), incapable(1)

Table 8.3: Frame targets (lexical items that evoked the frames) for the top 10 most frequent frames (part 2).

	Benefactive/Malefactive	Writer Attitude
Baseline	56.65	55.61
BOW	67.13±2.68	66.61±1.90
DepTree	72.10±2.41	66.16±1.76
SemTreeFWD	72.51±2.22	65.32±2.05
OmniGraph <sup>WL</sup>	<b>83.17±1.93</b>	<b>73.10±1.64</b>
OmniGraph <sup>NEW</sup>	<b>82.42±2.04</b>	<b>74.24±1.58</b>

Table 8.4: Mean accuracy for Benefactive/Malefactive event and Writer Attitude tasks.

Table 8.4 summarizes the performance. For Benefactive/Malefactive task, BOW obtains a 10% improvement over the baseline. Structured representations significantly improves the vector based BOW. SemTreeFWD, which incorporates the semantic features and sentiment dictionary further improves the performance by another 5%. The dependency tree performs similar with a tiny lower average a bigger standard deviation. OmniGraphs with graph kernel learning (WL or NEW kernels) performs much better. The writer attitude task is a more difficult one. The baseline is a little lower, dependency trees and semantic trees representation haven't been able to improve BOW. However, OmniGraphs with both graph kernel learnings still have a significant improvement.

We conduct feature analysis to understand what types of features that contribute to the high performance of OmniGraphs. We linearize the OmniGraph features generated by Node Edge Weighting graph kernel, and rank features based on mutual information. We look at the top 100 features and count the number features that require each feature type, e.g. frame name or lexical item. Shown in Table 8.9 is the distribution of the counts for the Benefactive/Malefactive task, and Table 8.10 is for the Writer Attitude task. We see that frame name is consistently the most frequent feature type for both tasks, and the frame element (semantic role) is the second. These two types of features are the major boost of the performance for our semantic frame based model. They have been effectively used to generalize the meanings of different lexical items, and result in high predictive capability. Table 8.11 and 8.12 show example top ranked features for the Benefactive/Malefactive task, and Table 8.13 shows an example top ranked feature for the Writer Attitude task.

For the time being, there has been some work on goodFor/badFor dataset. [Deng and Wiebe, 2014a] described a rule-based conceptual framework for representing and analyzing opinion im-

plicatures. To understand implicatures, their system recognizes implicit sentiments (and beliefs) toward various events and entities in the sentence. [Deng and Wiebe, 2014b] applied Loopy Belief Propagation to propagate sentiments among entities. [Deng *et al.*, 2014] incorporate the inferences developed by implicature rules into Integer Linear Programming to jointly improve sentiment detection toward entities and disambiguate components of *gfbf* events. Their work cover multiple tasks on the dataset, such as detecting the sentiment spans for *gfbf* events, identifying which the noun phrase is the agent and which is the theme, and given a *gfbf* text span, which is its polarity, positive or negative? Our work only focus on the polarity detection task, and the different experimental setup may not allow a direction comparison. For example, they use a train/test split while we use cross-validation. [Deng and Wiebe, 2014a] does not report performance. Even though [Deng *et al.*, 2014] is not strictly comparable, none of their accuracies are above 0.7. We do not use rule-based framework for the representation of opinion implicatures as in [Deng and Wiebe, 2014a] and [Deng *et al.*, 2014]. However, our use of FrameNet to generalize the meaning of words and the use of graph kernel on OmniGraph automates the process of rich feature engineering, with a coverage of lexical items, semantic frame features, and dependencies among frames. The extracted semantic graph-based features can also facilitates the understanding of the problem domain and the relations between the target entity and the theme, such as in Figure 8.11, 8.12, and 8.13. These advantages may be the contribution that leads to the high accuracy. [Choi and Wiebe, 2014] addressed methods for creating a lexicon of positive/negative effect events to support opinion inference. They selected lexical units from FrameNet, such as *assemble*, *create* of the *Creating* frame to build a graph-based model in which each node is a WordNet sense, and edges represent semantic WordNet relations between sense. While our use of graph kernel learning on OmniGrpah automates the lexicon extraction for positive/negative effect events. In addition, our engineered features are not restricted to lexicons but contains relational features that combines lexical items, frame names, frame elements, and the designated entities. The above related work have been work on the Benefactive/Malefactive task, and our experiment also covers the Writer Attitude task in the GoodFor/BadFor corpus.

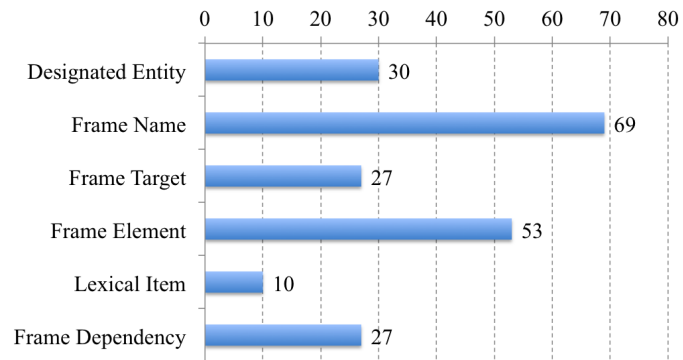


Figure 8.9: Number of the top 100 ranked features requiring each feature type for the Benefec-tive/Malefactive task.

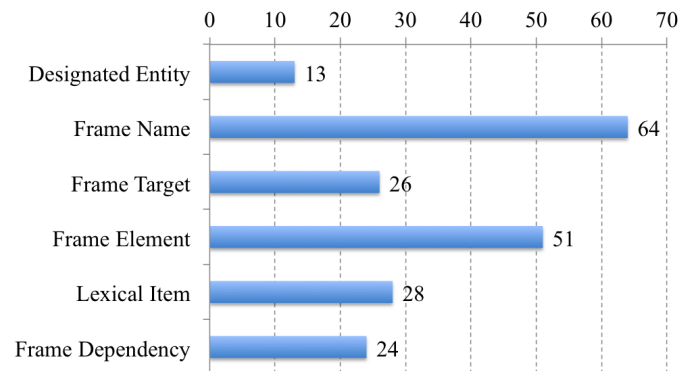
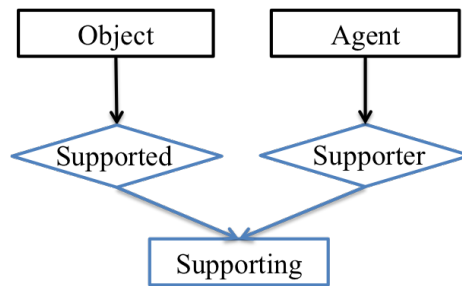


Figure 8.10: Number of the top 100 ranked features requiring each feature type for the Writer Attitude task.



**Feature Types:**

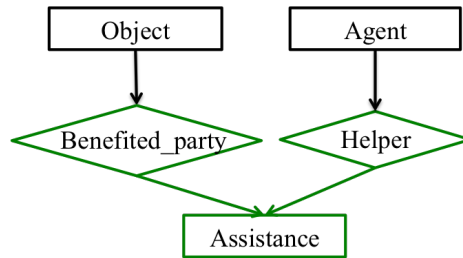
Object Entity	Frame Name	Frame Target	Frame Element	Lexical Item	Frame Dependency
✓	✓		✓		

**Example Sentences:**

[These programs<sub>Agent</sub>] bolster [nursing education at all levels<sub>Object</sub>], from entry-level preparation through the development of advanced practice nurses.

Bennet went on the record weeks ago saying [he<sub>Agent</sub>] would support [the bill<sub>Object</sub>] even if it cost him his job.

Figure 8.11: Graph features that predicts a positive polarity for the Object Entity in the Benefactive/Malefactive task.



**Feature Types:**

Object Entity	Frame Name	Frame Target	Frame Element	Lexical Item	Frame Dependency
✓	✓		✓		

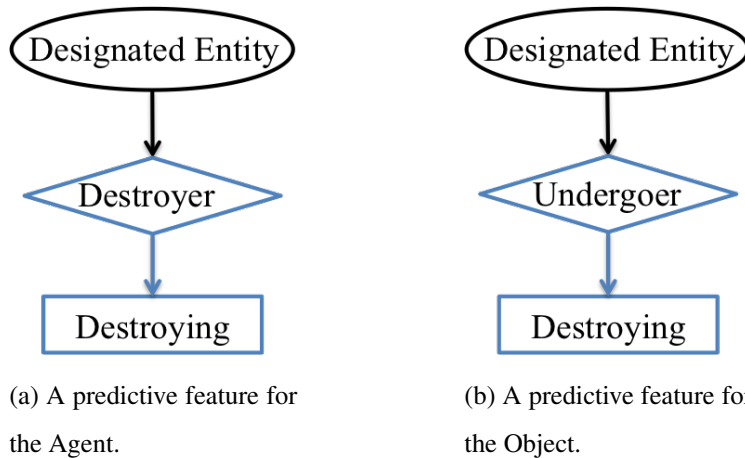
**Example Sentences:**

Looking ahead to the benefits health care reform will bring in future years, the law also established [a pregnancy assistance fund<sub>Agent</sub>] that will provide \$250 million over the next decade to help [pregnant and parenting women and teens with child care, housing, education and services for those victimized by domestic or sexual violence<sub>Object</sub>].

To sell ObamaCare and manufacture support, [desperate Democrats<sub>Agent</sub>] pandered to [the college set and their parents<sub>Object</sub>].

Figure 8.12: Graph features that predicts a positive polarity for the Object Entity in the Benefactive/Malefactive task.





**Feature Types:**

Designated Entity	Frame Name	Frame Target	Frame Element	Lexical Item	Frame Dependency
✓	✓		✓		

**Example Sentences:**

By utilizing peer review practices which would not stand muster under standard constitutional law, [hospital and health systems<sub>Agent</sub>] can label anyone a disruptive, unruly or uncooperative physician and destroy [their ability to work<sub>Object</sub>].

Figure 8.13: Graph features that predicts a negative polarity on the Designated Entity in the Writer Attitude task.

## **Part IV**

# **Conclusions**

## Chapter 9

# Conclusions

This thesis presents the study on entity-driven text analytics, where we specify the designated entity of interest, and use the information and signals from the real world to label entity mentions and make predictions. This work is also closely related to text forecasting and sentiment analysis. The methods we proposed build on frame semantics, a conceptual representation that generalizes from words and phrases to abstract scenarios, or frames, that capture explicit and implicit meaning of sentences. We demonstrate different approaches to incorporate the semantic features that make use of structures of data representation and learning models, including vectors, trees, and graphs.

The hypotheses behind our vector space model are that it is possible to 1) identify the underlying scenarios in text by generalizing the meanings of words when the word forms are different (e.g. *sue* and *accuse* both indicate a judgment communication scenario); 2) distinguish the word senses for the same word form (e.g. *right* for correctness or a legal entitlement); 3) capture the semantic roles (e.g. *Communicator*, *Evaluee*, and *Reason* roles in *Judgement Communication* frame); 4) improve sentiment-related tasks by incorporating the semantic orientations in words based on a psycholinguistic dictionary (e.g. Dictionary of Affect in Language, DAL). To test the hypotheses, we carry out experiments to incrementally combine different types of features and evaluate the classification performance. Our result in Table 7.3 shows our vector space features bring advantages over bag-of-words, and different features have different predictive power for different tasks (e.g. frame names is good for *change* task while frame target is good for *polarity* task).

The hypotheses behind our semantic tree space model are that, given a designated entity, it is possible to identify not only the scenario the entity locates in, but also the semantic role(s) it fills

(e.g. the designated entity *Oracle* fills both the *Communicator* and *Speaker* roles in sentence *Oracle sued Google, saying Google's Android mobile operating system infringes the copyrights of Java*). Both the semantic roles (frame elements) and scenarios (frames) can be neatly encoded in a tree structure where the designated entity is the root node, the semantic roles it fills are its immediate children, and the frames are the children of the semantic roles. We also hypothesize that the use of tree kernel learning on our tree space model can effectively measure the similarities between sentences with entity mentions and distinguish entities with different semantic roles, although they have the same features in vector space (e.g. to distinguish the different roles of *Oracle* and *Google* in sentence *Oracle sued Google*, which bag-of-words and bag-of-frames are incapable of). To test the hypotheses, we evaluate our tree space model against 1) bag-of-words model, 2) enriched vector space model that contains additional bag-of-frames and part-of-speech-specific psycholinguistic dictionary features, and 3) supervised topic models. We found that including *SemTree* on top of vector space model outperformed all three benchmarks (Table 7.4). We also carried out a post-hoc analysis by linearizing the tree kernel features and found that the tree structured semantic information provide insights for problem understanding and convenient model interpretation.

The hypotheses behind our graph space model are that, compared to trees, graphs provide a more flexible data structure with fewer topological constraints (e.g. allow cycles and no distinction between root and leaf nodes) that can encode fine-grained and richly varied features, such as semantic frames, semantic roles, word forms and dependency structures among frames. We can accurately measure data similarity using a fast kernel method, with improved classification performance. Also, through a fast linearization of graph kernel features, complex features with hierarchical structure can be extracted for model interpretation and feature selection. An advantage of the graph representation over trees is that in a graph any single node or group of nodes can be conveniently extracted and measured without treating their topological roles differently, such as root or leaf nodes. A substructure in trees is often a parent node with all its direct children or all descendants, depending on whether subset tree or subtree kernel is used. However, each node in a graph can be treated as a root node. The structural features from a graph can therefore include different sizes of the neighborhood when centered at each node. Another advantage of *OmniGraph* over *SemTree* is that the root node of *SemTree* is designated to be the single entity node and is used as a joint to connect the frame features (e.g. semantic roles, and frame names) of that entity, and the features of the frames without

a designated entity mention will not be included. *OmniGraph* is able to use all frames, even those without any entity mention. This can be done by connecting frames using the dependencies among them. These dependencies can be extracted based on syntactic dependency parsing. In *SemTree*, including dependencies among frames introduces loops. Furthermore, specifying the directions of the edges among nodes of different feature types based on syntactic dependencies allows us to obtain a more precise encoding of structural features by restricting the propagation of relational information through a particular pattern. For example, this can require that the frame feature in the subordinate clause depends on the frame feature in the main clause but not vice versa, or the words that fill a frame element should depend on the frame element but not that the frame element depends on the words. To test the hypotheses and the projected benefits of *OmniGraph*, we carry out experiments to compare a variety of realizations of *OmniGraph* to vector space model and tree space model by incrementally adding features in *OmniGraph*.

In the experiment of this study, we break down the components of the graph and analyze the contributing factors of each component. This analysis will provide us with an in-depth understanding of the model, and can also provide explanation about the real world phenomena that the model tries to predict. Our *OmniGraph* provides a unified representation of different types of features, and relies on a convolution graph kernel, a type of kernel that iteratively measures sub-parts of the graph, for support vector machine learning. Features are encoded in graphs as nodes and edges. The types of nodes include (1) entities, (2) frames names, (3) frame targets, (4) frame elements, and (5) lexical items. The types of edges include (1) ⟨entity, semantic role (frame element)⟩ relations, (2) ⟨frame target, frame⟩ relations, (3) ⟨frame element, frame⟩ relations, (4) ⟨lexical item, frame element⟩ relations, and (5) ⟨frame, frame⟩ (dependency) relations. The edges can be undirected or directed. When directed edge is applied, we use the syntactic dependencies to specify the direction of edges. Based on the characteristics of our graph structured representation, we select the Weisfeiler-Lehman (WL) graph kernel for machine learning and feature exploration. WL graph kernel efficiently measures the similarities among graphs by breaking down the similarity calculation for different neighborhood sizes. For 0-degree neighborhood, WL kernel measures the overlap of individual nodes, which is analogous to the bag-of- $X$  model. For  $K$ -degree neighborhoods, where  $K > 0$ , WL kernel measures the similarity of the nodes that are  $K$  edges away. The procedure is also called neighborhood augmentation. We observe that different node types (i.e. feature types of

the node, e.g. frame name, lexical item, etc.) and edge types (i.e. feature types of the relations between nodes, e.g. a designated entity fills a semantic role, dependencies among frames, etc.) have different contribution to the prediction task with different predictive ability. We propose a novel node edge weighting (NEW) graph kernel that allows the exploration of finer-grained subgraph features. NEW graph kernel assigns different weights to nodes and edges according to their feature types, and allows partial match on neighborhood comparisons when it calculates subgraph similarities.

We conduct experiments in financial news analytics to test the benefits of our proposed methods, which cover different feature types and different data representations and learning methods. We align stock price data with news articles for companies in S&P500, and use financial news to predict the price movement of company mentions in news. Our results show that our OmniGraph representation with WL and NEW graph kernel learning exhibit superior performance. The advantages of OmniGraph stem from the use of semantic frame features to generalize word meanings in a flexible and extensible graph structure, where rich relational linguistic information can be modeled and learned. In feature analysis, we found that the expressiveness of our subgraph features are much beyond the vector space word-based model. These structured features also bring insights to the problem domain. The superior performance of our graph structured representation and learning also exhibit in the GoodFor/BadFor dataset, where two tasks are involved. One task is to classify whether the agent and the event mentioned in the sentence is benefactive or malefactive on the affected entity (the object). The other task is to identify if the writer has a positive or negative attitude towards the agent and the object in the sentence. OmniGraph with WL and NEW graph kernel significantly outperform the baseline and the other benchmarks that rely on vectors and trees. Our methods can provide an exemplary approach for other fine-grained sentiment analysis tasks.

## **Part V**

# **Bibliography**

## Bibliography

- [Agarwal *et al.*, 2009] Apoorv Agarwal, Fadi Biadisy, and Kathleen Mckeown. Contextual phrase-level polarity analysis using lexical affect scoring and syntactic N-grams. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 24–32, Athens, Greece, March 2009. Association for Computational Linguistics.
- [Agarwal *et al.*, 2011] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media, LSM '11*, pages 30–38. Association for Computational Linguistics, 2011.
- [Agarwal *et al.*, 2014] Apoorv Agarwal, Sriramkumar Balabsubramanian, Anup Kotalwar, Jiehan Zheng, and Owen Rambow. Frame semantic tree kernels for social network extraction from text. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 211–219, 2014.
- [Bach *et al.*, 2004] Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proceedings of the Twenty-first International Conference on Machine Learning, ICML '04*, pages 6–, New York, NY, USA, 2004. ACM.
- [Bamman *et al.*, 2013] David Bamman, Brendan O'Connor, and Noah A. Smith. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1)*, pages 352–361, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.
- [Banarescu *et al.*, 2013] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking, 2013.



- [Bar-Haim *et al.*, 2011] Roy Bar-Haim, Elad Dinur, Ronen Feldman, Moshe Fresko, and Guy Goldstein. Identifying and following expert investors in stock microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1310–1319, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.
- [Bengtson and Roth, 2008] Eric Bengtson and Dan Roth. Understanding the value of features for coreference resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08*, pages 294–303, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.
- [Black and Scholes, 1973] Fishcer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637, 1973.
- [Blei *et al.*, 2003] David M. Blei, Andrew Ng, and Michael Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.
- [Borgwardt and Kriegel, 2005] Karsten M. Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining, ICDM '05*, pages 74–81, Washington, DC, USA, 2005. IEEE Computer Society.
- [Bulyko and Ostendorf, 2003] Ivan Bulyko and Mari Ostendorf. Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures. In *Proc. HLT-NAACL 2003*, pages 7–9, 2003.
- [Cai and Knight, 2013] Shu Cai and Kevin Knight. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 2: Short Papers*, pages 748–752, 2013.
- [Chan, 2003] Wesley S. Chan. Stock price reaction to news and no-news: drift and reversal after headlines. *Journal of Financial Economics*, 70(2):223 – 260, 2003.
- [Choi and Wiebe, 2014] Yoonjung Choi and Janyce Wiebe. +/-effectwordnet: Sense-level lexicon acquisition for opinion inference. In *Proceedings of the 2014 Conference on Empirical Methods*

*in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1181–1191, 2014.

[Chua *et al.*, 2009] Christopher Chua, Maria Milosavljevic, and James R. Curran. A sentiment detection engine for internet stock message boards. In *Proceedings of the Australasian Language Technology Association Workshop 2009*, pages 89–93, Sydney, Australia, December 2009.

[Collins and Duffy, 2001] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Proceedings of the 14th Conference on Neural Information Processing Systems*, 2001.

[Collins and Duffy, 2002] Michael Collins and Nigel Duffy. New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02*, pages 263–270, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[Cox *et al.*, 1979] John C. Cox, Stephen A. Ross, and Mark Rubenstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

[Creamer *et al.*, 2013] Germán G. Creamer, Yong Ren, Yasuaki Sacamoto, and Jeffrey V. Nickerson. News and sentiment analysis of the european market with a hybrid expert weighting algorithm. In *SocialCom'13*, pages 391–396. IEEE, 2013.

[Cruse, 1986] D. A. Cruse. *Lexical Semantics*. Cambridge University Press, September 1986.

[Culotta *et al.*, 2007] Aron Culotta, Michael Wick, Robert Hall, and Andrew McCallum. First-order probabilistic models for coreference resolution. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*, pages 81–88, 2007.

[Das and Smith, 2011] Dipanjan Das and Noah A. Smith. Semi-supervised frame-semantic parsing for unknown predicates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11*, pages 1435–1444, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

- [Das and Smith, 2012] Dipanjan Das and Noah A. Smith. Graph-based lexicon expansion with sparsity-inducing penalties. In *HLT-NAACL*, pages 677–687. The Association for Computational Linguistics, 2012.
- [Deerwester *et al.*, 1990] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshmn. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 1990.
- [Deng and Wiebe, 2014a] Lingjia Deng and Janyce Wiebe. A conceptual framework for inferring implicatures. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA-2014), June 27, 2014, Baltimore, Maryland, USA*, pages 154–159, 2014.
- [Deng and Wiebe, 2014b] Lingjia Deng and Janyce Wiebe. Sentiment propagation via implicature constraints. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 377–385, 2014.
- [Deng *et al.*, 2013] Lingjia Deng, Yoonjung Choi, and Janyce Wiebe. Benefactive/malefactive event and writer attitude annotation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 120–125, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.
- [Deng *et al.*, 2014] Lingjia Deng, Janyce Wiebe, and Yoonjung Choi. Joint inference and disambiguation of implicit sentiments via implicature constraints. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 79–88, 2014.
- [Devitt and Ahmad, 2007] Ann Devitt and Khurshid Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 984–991, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

- [Dolan *et al.*, 2004] William Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. *Proceedings of the 20th International Conference on Computational Linguistics*, 2004.
- [Ediger *et al.*, 2010] David Ediger, Karl Jiang, Jason Riedy, David A. Bader, and Courtney Corley. Massive social network analysis: Mining twitter for social good. In *Proceedings of the 2010 39th International Conference on Parallel Processing*, ICPP '10, pages 583–593, Washington, DC, USA, 2010. IEEE Computer Society.
- [Eisner, 1996] Jason M. Eisner. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 340–345, Stroudsburg, PA, USA, 1996. Association for Computational Linguistics.
- [Engelberg and Parsons, 2011] Joseph Engelberg and Christopher A. Parsons. The causal impact of media in financial markets. *Journal of Finance*, 66(1):67–97, 2011.
- [Fama and French, 1993] Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3 – 56, 1993.
- [Feldman *et al.*, 2011a] Ronen Feldman, Benjamin Rosenfeld, Roy Bar-Haim, and Moshe Fresko. The stock sonar - sentiment analysis of stocks based on a hybrid approach. In *Proceedings of the Twenty-Third Conference on Innovative Applications of Artificial Intelligence, August 9-11, 2011, San Francisco, California, USA*, 2011.
- [Feldman *et al.*, 2011b] Ronen Feldman, Benjamin Rosenfeld, Roy Bar-Haim, and Moshe Fresko. The stock sonar - sentiment analysis of stocks based on a hybrid approach. In *Proceedings of the Twenty-Third Conference on Innovative Applications of Artificial Intelligence, August 9-11, 2011, San Francisco, California, USA*, 2011.
- [Fillmore *et al.*, 2003] Charles J. Fillmore, Christopher R. Johnson, and Miriam R. L. Petruck. Background to Framenet. *International Journal of Lexicography*, 16(3):235–250, September 2003.

- [Fillmore, 1976] Charles J. Fillmore. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32, 1976.
- [Fisher *et al.*, 2011] Matthew Fisher, Manolis Savva, and Pat Hanrahan. Characterizing structural relationships in scenes using graph kernels. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 34:1–34:12, New York, NY, USA, 2011. ACM.
- [Flanigan *et al.*, 2014] Jeffrey Flanigan, Sam Thomson, Jaime G. Carbonell, Chris Dyer, and Noah A. Smith. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 1426–1436, 2014.
- [Forman, 2003] George Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, March 2003.
- [Gärtner *et al.*, 2003] Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Conference on Learning Theory*, pages 129–143, 2003.
- [Gentzkow and Shapiro, 2010] M. Gentzkow and J. M. Shapiro. What drives media slant? evidence from u.s. daily newspapers. *Econometrica*, 78(1):3571, 2010.
- [Gentzkow, 2006] M. Gentzkow. Television and voter turnout. *The Quarterly Journal of Economics*, 121(3):931972, 2006.
- [Gerber *et al.*, 2009] A. S. Gerber, D. Karlan, and D. Bergan. Does the media matter? a field experiment measuring the effect of newspapers on voting behavior and political opinions. *American Economic Journal: Applied Economics*, 1(2):3552, 2009.
- [Goyal and Daumé, 2011] Amit Goyal and Hal Daumé, III. Generating semantic orientation lexicon using large data and thesaurus. In *Proceedings of the 2Nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '11, pages 37–43, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
- [GuoDong and Jian, 2004] Zhou GuoDong and Su Jian. A high-performance coreference resolution system using a constraint-based multi-agent strategy. In *Proceedings of the 20th Interna-*

- tional Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.
- [Haghighi and Klein, 2009] Aria Haghighi and Dan Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1152–1161, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [Haider and Mehrotra, 2011] Syed Aqueel Haider and Rishabh Mehrotra. Corporate news classification and valence prediction: A supervised approach. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, pages 175–181, Portland, Oregon, June 2011. Association for Computational Linguistics.
- [Hassan and Radev, 2010] Ahmed Hassan and Dragomir Radev. Identifying text polarity using random walks. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 395–403, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [Haussler, 1999] David Haussler. Convolution kernels on discrete structures. Technical Report UCS-CRL-99-10, University of California at Santa Cruz, Santa Cruz, CA, USA, 1999.
- [Hoffmann *et al.*, 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 541–550, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
- [Horváth *et al.*, 2004] Tamás Horváth, Thomas Gärtner, and Stefan Wrobel. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 158–167, New York, NY, USA, 2004. ACM.
- [Kashima *et al.*, 2003] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 321–328. AAAI Press, 2003.

- [Kim and Hovy, 2006] Soo-Min Kim and Eduard Hovy. Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the Workshop on Sentiment and Subjectivity in Text*, SST '06, pages 1–8, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
- [Kogan *et al.*, 2009] Shimon Kogan, Dimitry Levin, Bryan R. Routledge, Jacob S. Sagi, and Noah A. Smith. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 272–280, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [Lappin and Leass, 1994] Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Comput. Linguist.*, 20(4):535–561, December 1994.
- [Lee *et al.*, 2011] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, CONLL Shared Task '11, pages 28–34, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
- [Lee *et al.*, 2013] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4), 2013.
- [Lee *et al.*, 2014] Heeyoung Lee, Mihai Surdeanu, Bill Maccartney, and Dan Jurafsky. On the importance of text analysis for stock price prediction. In *LREC'14*, Reykjavik, Iceland, may 2014.
- [Li, 2000] David X. Li. On default correlation: A copula function approach. *Journal of Financial Economics*, 9(4):43–54, 2000.
- [Luo *et al.*, 2004] Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla, and Salim Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

- [Luss and d'Aspremont, 2008] Ronny Luss and Alexandre d'Aspremont. Predicting abnormal returns from news using text classification. *CoRR*, abs/0809.2792, 2008.
- [Mahé and Vert, 2009] Pierre Mahé and Jean-Philippe Vert. Graph kernels based on tree patterns for molecules. *Journal of Machine Learning Research*, 75(1):3–35, April 2009.
- [Manning *et al.*, 2014] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the ACL*, pages 55–60, 2014.
- [Matthews, 1975] Brian W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, 1975.
- [McClosky *et al.*, 2010] David McClosky, Eugene Charniak, and Mark Johnson. Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 28–36, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [McDonald *et al.*, 2005a] Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 91–98, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
- [McDonald *et al.*, 2005b] Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 523–530, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.
- [Min *et al.*, 2013] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, 2013.



- [Mintz *et al.*, 2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [Mohammad and Turney, 2010] Saif M. Mohammad and Peter D. 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*, CAAGET '10, pages 26–34, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [Morton, 2000] Thomas S. Morton. Coreference for nlp applications. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 173–180, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.
- [Moschitti, 2006] Alessandro Moschitti. Making tree kernels practical for natural language learning. In *In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.
- [Ng and Cardie, 2002] Vincent Ng and Claire Cardie. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
- [Nicolae and Nicolae, 2006] Cristina Nicolae and Gabriel Nicolae. Bestcut: A graph algorithm for coreference resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 275–283, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
- [O'Connor *et al.*, 2013] Brendan O'Connor, Brandon M. Stewart, and Noah A. Smith. Learning to extract international relations from political context. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1)*, pages 1094–1104, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

- [Pighin and Moschitti, 2009] Daniele Pighin and Alessandro Moschitti. Reverse engineering of tree kernel feature spaces. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore*, pages 111–120, 2009.
- [Poesio *et al.*, 2004] Massimo Poesio, Rahul Mehta, Axel Maroudas, and Janet Hitzeman. Learning to resolve bridging references. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics, ACL '04, Stroudsburg, PA, USA, 2004*. Association for Computational Linguistics.
- [Purda and Skillicorn, 2014] Lynnette Purda and David Skillicorn. Accounting variables, deception, and a bag of words: Assessing the tools of fraud detection. *Contemporary Accounting Research*, 2014.
- [Raghunathan *et al.*, 2010] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10*, pages 492–501, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [Ravi *et al.*, 2008] Sujith Ravi, Kevin Knight, and Radu Soricut. Automatic prediction of parser accuracy. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 887–896, Honolulu, Hawaii, October 2008.
- [Recasens *et al.*, 2013] Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. The life and death of discourse entities: Identifying singleton mentions. In *Human Language Technologies: The 2013 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 627–633, Stroudsburg, PA, June 2013. Association for Computational Linguistics.
- [Riedel *et al.*, 2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III, ECML PKDD'10*, pages 148–163, Berlin, Heidelberg, 2010. Springer-Verlag.

- [Rosenfeld and Feldman, 2007] Benjamin Rosenfeld and Ronen Feldman. Using corpus statistics on entities to improve semi-supervised relation extraction from the web. In *ACL 2007, Proceedings of the 45th Annual Meeting of the ACL, June 23-30, 2007, Prague, Czech Republic, 2007*.
- [Roux *et al.*, 2012] Joseph Le Roux, Jennifer Foster, Joachim Wagner, Rasul Samad, Zadeh Kaljahi, and Anton Bryl. DUC-Paris13 systems for the SANCL 2012 shared task, 2012.
- [Rudin, 2009] Cynthia Rudin. The P-Norm Push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10:2233–2271, Oct 2009.
- [Ruppenhofer and Rehbein, 2012] Josef Ruppenhofer and Ines Rehbein. Semantic frames as an anchor representation for sentiment analysis. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, WASSA '12*, pages 104–109, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.
- [Rydberg and Shephard, 2003] Tina H. Rydberg and Neil Shephard. Dynamics of Trade-by-Trade Price Movements: Decomposition and Models. *Journal of Financial Econometrics*, 1(1):2–25, 2003.
- [Salleb-Aouissi *et al.*, 2011] Ansaif Salleb-Aouissi, Axinia Radeva, Rebecca J. Passonneau, Boyi Xie, Faiza Khan Khattak, Ashish Tomar, Hatim Diab, David Waltz, Mary McCord, Harriet McGurk, and Noemie Elhadad. Diving into a large corpus of pediatric notes. *ICML 2011 Learning from Unstructured Clinical Text Workshop*, 2011.
- [Sarikaya *et al.*, 2005] Ruhi Sarikaya, Agustín Gravano, and Yuqing Gao. Rapid language model development using external resources for new spoken dialog domains. In *International Congress of Acoustics, Speech, and Signal Processing (ICASSP)*, pages 573–576, Philadelphia, PA, USA, 2005. IEEE, Signal Processing Society.
- [Sayeed *et al.*, 2012] Asad B. Sayeed, Jordan Boyd-Graber, Bryan Rusk, and Amy Weinberg. Grammatical structures for word-level sentiment detection. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT '12*, pages 667–676, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

- [Scheible and Schütze, 2013] Christian Scheible and Hinrich Schütze. Sentiment relevance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1)*, pages 954–963, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.
- [Schumaker *et al.*, 2012] Robert P. Schumaker, Yulei Zhang, Chun-Neng Huang, and Hsinchun Chen. Evaluating sentiment in financial news articles. *Decision Support Systems*, 53(3):458 – 464, 2012.
- [Sharpe and Sharpe, 1970] William F Sharpe and WF Sharpe. *Portfolio theory and capital markets*, volume 217. McGraw-Hill New York, 1970.
- [Shervashidze *et al.*, 2009] Nino Shervashidze, S. V. N. Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten M. Borgwardt. Efficient graphlet kernels for large graph comparison. *Journal of Machine Learning Research - Proceedings Track*, 5:488–495, 2009.
- [Shervashidze *et al.*, 2011] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, November 2011.
- [Smith, 2010] Noah A. Smith. Text-driven forecasting, March 2010.
- [Smits and Jordaan, 2002] G.F. Smits and E.M. Jordaan. Improved svm regression using mixtures of kernels. In *Proceedings of 2002 International Joint Conference on Neural Networks, IJCNN’02*, pages 2785–2790. IEEE, 2002.
- [Stromberg, 2004] D. Stromberg. Radios impact on public spending. *The Quarterly Journal of Economics*, 119(1):189221, 2004.
- [Surdeanu *et al.*, 2012] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL ’12*, pages 455–465, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.
- [Takamatsu *et al.*, 2012] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting*

- of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 721–729, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.
- [Tetlock *et al.*, 2008] Paul C. Tetlock, Maytal Saar-Tsechansky, and Sofus Macskassy. More than Words: Quantifying Language to Measure Firms' Fundamentals. *The Journal of Finance*, 2008.
- [Tetlock, 2007] Paul C. Tetlock. Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *The Journal of Finance*, 2007.
- [Wahba and Wahba, 1998] Grace Wahba and Grace Wahba. Support Vector Machines, Reproducing Kernel Hilbert Spaces and the Randomized GACV, 1998.
- [Wale *et al.*, 2008] Nikil Wale, IanA. Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14(3):347–375, 2008.
- [Wang *et al.*, 2011] Shanshan Wang, Kaiquan Xu, Long Liu, Bing Fang, Shaoyi Liao, and Huaqing Wang. An ontology based framework for mining dependence relationships between news and financial instruments. *Expert System Application*, 38(10):12044–12050, September 2011.
- [Watkins, 2000] Chris Watkins. Dynamic alignment kernels. In A. Smola and P. Bartlett, editors, *Advances in Large Margin Classifiers*, chapter 3, pages 39–50. MIT Press, Cambridge, MA, USA, 2000.
- [Weisfeiler and Lehman, 1968] B. Weisfeiler and A. A. Lehman. A reduction of graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya, Ser. 2, no. 9*, 1968.
- [Whissel, 1989] Cynthia M. Whissel. The dictionary of affect in language. *Emotion: Theory, Research, and Experience*, 39(4):113–131, 1989.
- [Wilson *et al.*, 2009] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Comput. Linguist.*, 35(3):399–433, September 2009.

- [Wintrode and Khudanpur, 2014] J. Wintrode and S. Khudanpur. Limited resource term detection for effective topic identification of speech. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7118–7122, May 2014.
- [Wolodja Wentland and Hartung, 2008] Carina Silberer Wolodja Wentland, Johannes Knopp and Matthias Hartung. Building a multilingual lexical resource for named entity disambiguation, translation and transliteration. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.
- [Xie *et al.*, 2012] Boyi Xie, Rebecca J. Passonneau, Haimonti Dutta, Jing-Yeu Miaw, Axinia Radeva, Ashish Tomar, and Cynthia Rudin. Progressive clustering with learned seeds: An event categorization system for power grid. In *Proceedings of the 24th International Conference on Software Engineering & Knowledge Engineering (SEKE'2012)*, Hotel Sofitel, Redwood City, San Francisco Bay, USA July 1-3, 2012, pages 100–105, 2012.
- [Xie *et al.*, 2013] Boyi Xie, Rebecca J. Passonneau, Leon Wu, and Germán Creamer. Semantic frames to predict stock price movement. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1)*, pages 873–883, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.
- [Yang *et al.*, 2004] Xiaofeng Yang, Jian Su, Guodong Zhou, and Chew Lim Tan. An np-cluster based approach to coreference resolution. In *Proceedings of the 20th International Conference on Computational Linguistics, COLING '04*, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.
- [Yang *et al.*, 2008] Xiaofeng Yang, Jian Su, Jun Lang, Chew Lim, Tan Ting, and Liu Sheng Li. An entity-mention model for coreference resolution with inductive logic programming. In *Proceedings of the 46th Annual Meeting on Association for Computational Linguistics, ACL '08*, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.