Distributional Semantics for Robust Automatic Summarization

by

Jackie Chi Kit Cheung

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Copyright © 2014 by Jackie Chi Kit Cheung

Abstract

Distributional Semantics for Robust Automatic Summarization

Jackie Chi Kit Cheung

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2014

Large text collections are an important resource of information about the world, containing everything from movie reviews and research papers to news articles about current events. Yet the sheer size of such collections presents a challenge for applications to make sense of this data and present it to users. **Automatic summarization** is one potential solution which aims to shorten one or more source documents while retaining the important information. Summarization is a complex task that requires inferences about the form and content of the summary using a semantic model.

This dissertation examines the feasibility of **distributional semantics** as the core semantic representation for automatic summarization. In distributional semantics, the meanings of words and phrases are modelled by the contexts in which they appear. These models are easy to train and have found successful applications, but they have until recently not been seriously considered as contenders to support semantic inference for complex NLP tasks such as summarization because of a lack of evaluation methods that would demonstrate their benefit.

I argue that current automatic summarization systems avoid relying on semantic analysis by focusing instead on replicating the source text to be summarized, but that substantial progress will not be possible without semantic analysis and domain knowledge acquisition. To overcome these problems, I propose an evaluation framework for distributional semantics based on first principles about the role of a semantic formalism in supporting inference. My experiments show that current distributional semantic approaches can support semantic inference at

a phrasal level invariant to the constituent syntactic constructions better than a word overlap baseline.

Then, I present a novel technique to embed distributional semantic vectors into a generative probabilistic model for domain modelling. This model achieves state-of-the-art results in slot induction, which also translates into better summarization performance. Finally, I introduce a text-to-text generation technique called sentence enhancement that combines parts of heterogeneous source text sentences into a novel sentence, resulting in more informative and grammatical summary sentences than a previous sentence fusion approach. The success of this approach relies crucially on distributional semantics in order to determine which parts may be combined.

These results lay the groundwork for the development of future distributional semantic models, and demonstrate their utility in determining the form and content of automatic summaries.

Acknowledgements

Graduate school has been a memorable journey, and I am very fortunate to have met some wonderful people along the way. First of all, I'd like to thank my advisor, Gerald Penn, who has been extremely patient and supportive. He let me pursue the research topics that I fancied while providing guidance to my research and career. After all this time, I am still amazed by the depth and breadth of his research vision, and his acumen in selecting meaningful research problems. I can only hope that enough of his taste in research and his sense of humour has rubbed off on me.

I'd also like to thank the other members of my committee. Graeme Hirst has been an influential mentor. His insightful questions and comments have caused me to reflect on the nature of computational linguistics research. Hector Levesque brought in a much needed outside perspective from knowledge representation. I'd like to thank Suzanne Stevenson for interesting discussions in seminars, meetings, and at my defense. My external examiner, Mirella Lapata, has been a great inspiration for my research, and provided valuable feedback on this dissertation.

During my degree, I spent two highly productive terms at Microsoft Research Redmond in the Speech and Natural Language Processing groups. I'd like to thank my internship mentors Xiao Li, Hoifung Poon, and Lucy Vanderwende for fruitful discussions and collaborations, and the other members of the Speech and NLP groups for lively conversations over lunch.

I am grateful to have been supported by the Natural Sciences and Engineering Research Council of Canada and by a Facebook PhD Fellowship.

Graduate student life would have been drab without the enrichment of colleagues and friends. The Department of Computer Science and the Computational Linguistics group in particular are full of smart and friendly people who filled my days with joy. Jorge Aranda got me hooked on board games. Giovanna Thron and Siavosh Benabbas threw awesome parties. Jono Lung and Aditya Bhargava taught me about finance. Elizabeth Patitsas is a good friend who owns two cats and is very fit. Andrew Perrault introduced me to climbing. Erin Delisle

will do a great job organizing games. Aida Nematzadeh and Amin Tootoonchian are the kindest people that I know, and put up with my attempts to learn Farsi. George Dahl keeps me (and the CL community) honest. The lunch crew (Afsaneh Fazly, Libby Barak, Katie Fraser, Varada Kolhatkar, and Nona Naderi) accepted me into their fold. Michael Guerzhoy is a conspirator in further bouts of not-working. Yuval Filmus is brilliant and has a good perspective on life. Timothy Fowler, Eric Corlett, and Patricia Araujo Thaine are great meeting and dinner buddies. Dai Tri Le Man and Sergey Gorbunov kept my appreciation of music alive. Dan Lister, Sam Kim, and Mischa Auerbach-Ziogas made for formidable Agricola opponents. There are many others both inside and outside of the department who I am lucky to have met. I am also lucky to have "old friends" from Vancouver like Hung Yao Tjia, Muneori Otaka, Joseph Leung, Rory Harris and Sandra Yuen to catch up with.

Finally, my family is my rock. My brother Vincent, my sister Joyce and her new family help me put things in perspective. My parents, Cecilia and David, are always there to support me through all of my hopes and dreams, whining and boasting, failures and successes. They remind me that life is more than just about research, and are the foundation that gives me confidence to lead my life.

To my parents.

Contents

1 Introduction **1**

1.1 Two Approaches to Semantics 3

1.2 Semantics in Automatic Summarization 5

1.3 Thesis Statement 7

1.4 Dissertation Objectives 8

1.5 Structure of the Dissertation 10

1.5.1 Peer-Reviewed Publications 12

2 Centrality in Automatic Summarization **13**

2.1 The Design and Evaluation of Summarization Systems 13

2.1.1 Classification of Summaries 14

2.1.2 Steps in Summarization 17

2.1.3 Summarization Evaluation 17

2.2 The Assumption of Centrality 23

2.2.1 Text Properties Important in Summarization 24

2.2.2 Cognitive Determinants of Interest, Surprise, and Memorability 26

2.2.3 Centrality in Content Selection 29

2.2.4 Abstractive Summarization 36

2.3 Summarizing Remarks on Summarization 38

3 A Case for Domain Knowledge in Automatic Summarization 40

3.1 Related Work . . . 42

3.2 Theoretical basis of the analysis . . . 43

3.2.1 Caseframe Similarity . . . 45

3.2.2 An Example . . . 46

3.3 Experiments . . . 46

3.3.1 Study 1: Sentence Aggregation . . . 47

3.3.2 Study 2: Signature Caseframe Density . . . 50

3.3.3 Study 3: Summary Reconstruction . . . 54

3.4 Why Source-External Elements? . . . 57

3.4.1 Study 4: Provenance Study . . . 58

3.4.2 Study 5: Domain Study . . . 60

3.5 Summary and Discussion . . . 62

4 Compositional Distributional Semantics 64

4.1 Compositionality and Co-Compositionality in Distributional Semantics . . . 64

4.1.1 Several Distributional Semantic Models . . . 66

4.1.2 Other Distributional Models . . . 68

4.2 Evaluating Distributional Semantics for Inference . . . 71

4.2.1 Existing Evaluations . . . 73

4.2.2 An Evaluation Framework . . . 75

4.2.3 Task 1: Relation Classification . . . 76

4.2.4 Task 2: Restricted QA . . . 78

4.3 Experiments . . . 80

4.3.1 Task 1 . . . 82

4.3.2 Task 2 . . . 84

4.4 Conclusions . . . 85

5 Distributional Semantic Hidden Markov Models **87**

5.1 Related Work . . . 89

5.2 Distributional Semantic Hidden Markov Models . . . 90

5.2.1 Contextualization . . . 94

5.2.2 Training and Inference . . . 96

5.2.3 Summary and Generative Process . . . 96

5.3 Experiments . . . 97

5.4 Guided Summarization Slot Induction . . . 98

5.5 Multi-document Summarization: An Extrinsic Evaluation . . . 100

5.5.1 A KL-based Criterion . . . 101

5.5.2 Supervised Learning . . . 102

5.5.3 Method and Results . . . 103

5.6 Discussion . . . 104

6 Sentence Enhancement for Automatic Summarization **107**

6.1 Sentence Revision for Abstractive Summarization . . . 107

6.2 Related Work . . . 109

6.3 A Sentence Enhancement Algorithm . . . 110

6.3.1 Sentence Graph Creation . . . 111

6.3.2 Sentence Graph Expansion . . . 112

6.3.3 Tree Generation . . . 115

6.3.4 Linearization . . . 117

6.4 Experiments . . . 118

6.4.1 Method . . . 118

6.4.2 Results and Discussion . . . 121

6.5 Discussion . . . 122

6.6 Appendix: ILP Encoding . . . 123

6.6.1 Informativeness Score . . . 123

6.6.2 Objective Function 123
6.6.3 Syntactic Constraints 124
6.6.4 Semantic Constraints 125

7 Conclusion **126**

7.1 Summary of Contributions 126
7.1.1 Distributional Semantics 126
7.1.2 Abstractive Summarization 127
7.2 Limitations and Future Work 128
7.2.1 Distributional Semantics in Probabilistic Models 128
7.2.2 Abstractive Summarization 129

Bibliography **130**

List of Tables

3.1 A sentence decomposed into its dependency edges, and the caseframes derived from those edges that are considered. 44

3.2 The average number of source text sentences needed to cover a summary sentence. 48

3.3 Signature caseframe densities for different sets of summarizers, for the initial and update guided summarization tasks. 51

3.4 Density of signature caseframes after merging to various thresholds. 53

3.5 Coverage of caseframes in summaries with respect to the source text. 56

3.6 The effect on caseframe coverage of adding in-domain and out-of-domain documents. 57

3.7 Results of the provenance study. 59

3.8 Results of the domain study. 95% confidence intervals are given in parentheses. 61

4.1 Task 1 dataset characteristics. 82

4.2 Task 1 results in AUC scores, averaged over the four subsets. 83

4.3 Task 2 results, in normalized rank scores. 84

5.1 The correspondence between nodes in DSHMM, the domain components that they model, and the related elements in the clause. 93

5.2 Slot induction results on the TAC guided summarization data set. 99

5.3 TAC 2010 summarization results by three settings of ROUGE. 102

5.4 Analysis of the most probable event heads and arguments in the most preferred (+) and dispreferred (−) events and slots after supervised training. 105

6.1 Results of the sentence enhancement and fusion experiments on TAC 2010 and TAC 2011. 121

List of Figures

1.1 An example of inference in proof-theoretic semantics. 4

1.2 An example of a vector representation of a word in distributional semantics, and of calculating the similarity between the vector representations of two words. 4

2.1 An informative, generic, extractive summary. 14

2.2 An indicative, generic, extractive summary of movie reviews. 15

2.3 A summary topic and a human-authored, informative, topic-focused abstractive summary from DUC 2005. 15

2.4 Linguistic quality questions used in DUC evaluations 19

2.5 Sentences exhibiting different levels of salience and relevance from DUC 2005 26

2.6 A sample template give in the TAC 2011 summarization task. 36

3.1 Sample pairs of similar caseframes by relation type, and the similarity score assigned to them by the distributional model. 45

3.2 Average sentence cover size: the average number of sentences needed to generate the caseframes in a summary sentence. 49

3.3 Examples of signature caseframes found in Study 2. 50

3.4 Density of signature caseframes (Study 2). 52

3.5 Coverage of summary text caseframes in source text (Study 3). 55

5.1 Basic graphical representation of DSHMM. 91

6.1 An example of the input dependency trees for sentence graph creation and expansion. 112
6.2 Event coreference resolution as a maximum-weight bipartite graph matching problem. 114

Chapter 1

Introduction

Complex natural language processing (NLP) systems can be characterized by their need for semantic inference in order to understand and generate natural language; that is, they require the ability to make explicit some knowledge that is implicit in the text (Blackburn and Bos, 2005). The subject of this dissertation is the complex NLP application of **automatic summarization**, the task of condensing some input source material into a shorter, output summary. In automatic summarization, semantic inference can be used to understand the material in the source in relation to what is expected or known; it can be used to decide what important content should be expressed in the output summary; it can also be used to generate the final form of the output summary.

Automatic summarization has great potential for informing users and helping them make decisions, precisely because it can automate part of the inference that is required for determining the important and relevant information in text. As the amount of textual data that could be relevant increases, so too does the need for summarization as a tool to help make sense of it.

The necessary semantic inference for complex NLP and summarization in particular has proved to be difficult for automatic systems. The reason is that the ideal semantic formalism used to support inference must demonstrate two competing properties. The first is **expressive power**—the formalism must be capable of supporting the type of complex reasoning that

humans perform from the basis of world knowledge. The other is **robustness**—the meaning representations should be easy to construct and apply to any domain without large amounts of manual effort. These properties are in conflict because richer semantic representations necessitate a greater abstraction from the surface form of the text to a form in which inference can be performed, and this process of abstraction or **semantic analysis** can be difficult to learn.

To illustrate this point, consider the following pair of sentence:

(1.1) *The patient gained 10 pounds.*

(1.2) *The patient experienced a 10-pound weight gain.*

A shallow analysis of the meaning of these sentences might rely on their syntactic structures in order to relate the predicates (*gain* and *experience*) with their arguments (*patient*, *10 pounds* and *10-pound weight gain*). Such an analysis would miss the fact that these two sentences are paraphrases of each other that permit the same inferences and have the same truth condition; i.e., whenever one of the statements is true, so is the other. A more expressive semantic formalism would permit both sentences to map to the same meaning representation, which would solve the above problem. However, such an analysis would require greater abstraction over the syntactic structures of sentences; namely, recognizing that the meaning of the verb phrase in the first sentence, *gained 10 pounds*, is identical to that of a noun phrase in the second sentence, *a 10-pound weight gain*.

A caveat is in order at this point. In this dissertation, I take semantics to refer to the sort of literal meaning involving reasoning with objects in the world in the sense found in (Blackburn and Bos, 2005). I do not take it to refer to nuances in suggested meaning (i.e., implicature) that may be conveyed by a particular phrasing. For example, in the above pair of sentences, the second seems to suggest that the patient did not intentionally gain the weight, but that the weight gain might be the result of some course of medication. While the view of semantics that I adopt is incomplete, it is sufficient to account for many of the deductions that occur in automatic summarization, as I will discuss in Section 1.2. It also accords with the Motagovian,

logical accounts of compositional semantics, which I discuss next.

1.1 Two Approaches to Semantics

Logical semantics and **distributional semantics** are traditional approaches to semantics that represent two extremes along the spectrum between expressive power and robustness. Below, I introduce the two approaches by describing the types of meaning representations and inference mechanisms that characterize them. Knowledgeable readers may wish to proceed to Section 1.2, in which I discuss the use of semantics, or the lack thereof, in automatic summarization.

Logical semantics is an approach to computational semantics based on first-order and related logics (Frege, 1892; Montague, 1974)[1]. This approach supports expressive inference through logical rules of inference, chief among them modus ponens. Given a knowledge base of facts about the world, the truth value of a novel statement or query can be judged based on whether it is entailed by the propositions in the knowledge base.

For example, a statement such as *Sebastian is a cat and he likes catnip.* may be presented to the system in the form of a logical formula such as:

$$Cat(\text{Sebastian}) \wedge Likes(\text{Sebastian}, \text{catnip}) \tag{1.3}$$

Determining the truth value of this statement is done with respect to a formal **model** of the world using automated theorem proving techniques. Suppose the knowledge base contains the fact that all cats like catnip. Then, the above statement can be verified by applying generalized modus ponens, as shown in Figure 1.1.

One of the key features of this type of semantics that enable the powerful inference mechanism is that the meaning representation of a sentence is constructed compositionally from

[1]Note that many of the ideas that make up this approach to semantics actually originated before Frege.

Domain knowledge:
$\forall x\ Cat(x) \implies Likes(x, \text{catnip})$
Logical inference:
$Cat(\text{Sebastian})$
$\underline{\forall x\ Cat(x) \implies Likes(x, \text{catnip})}$
$\therefore Likes(\text{Sebastian}, \text{catnip})$

Figure 1.1: An example of inference in proof-theoretic semantics.

$$\overrightarrow{cat} = \begin{bmatrix} 0.29 & 4.63 & -0.39 & 7.77 & -1.11 & 2.03 & 2.32 \end{bmatrix}$$
$$cosine(\overrightarrow{cat}, \overrightarrow{dog}) = \frac{\overrightarrow{cat} \cdot \overrightarrow{dog}}{||\overrightarrow{cat}|| \times ||\overrightarrow{dog}||} = 0.82$$

Figure 1.2: An example of a vector representation of a word in distributional semantics, and of calculating the similarity between the vector representations of two words.

its subparts, being guided by the syntactic structure of the sentence. Compositionality is appealing because it facilitates a view of natural language as a kind of formal language, with a parallelism between syntactic clauses and semantic propositions. Strictly speaking, however, compositionality is broken in natural language by phenomena such as idiomatic expressions.

The main weakness of logical approaches to computational semantics is that they perform well only in the domain and task that they were designed for (see Wong and Mooney (2007) for a representative approach). A great deal of annotation effort is required to train methods to associate each word and sentence with a semantic form, and to build up a knowledge base that is appropriate for a target domain.

The other major approach to semantics is distributional semantics. In this approach, the meanings of words and phrases are modelled by the contexts in which they appear in a training corpus. A common aphorism describing this approach is that "you shall know a word by the company it keeps." (Firth, 1957).

For example, the meaning representation of a target word, say *cat*, would be a vector in which each dimension corresponds to a context word, and the component value represents the strength of the association between the target word (i.e., *cat* in this example) and the corre-

sponding context word (e.g., *purr*). Such vector representations are typically used to compute similarity scores between words. For example, a similarity measure such as cosine similarity can be computed between the vector representations of *cat* and *dog*, as in Figure 1.2. These similarity scores can then be used in some downstream application, such as clustering similar documents for information retrieval, or discovering synonyms.

These models are easy to train and have found successful applications, particularly in lexical semantics and in information retrieval, but they have until recently not been seriously considered as contenders to support semantic inference for complex NLP tasks.

The principal reason is that distributional semantic vectors cannot yet support the type of expressive inference that logical semantics can. In fact, there has not even been any clear evaluation methodology that can demonstrate their potential ability to do so. One reason for this lack is that until recently, distributional semantics has focused on issues of word meaning, and word vectors fall far short in capturing the type of domain knowledge that is built into the inference process of logical semantics. Recent interest in compositional models of distributional semantics aims to expand the domain in which distributional semantics operates with the goal of approximating the type of inferences possible in logical semantics at a phrasal or sentential level (Baroni et al., 2014).

There have also been approaches that can be described as hybrid methods that combine logical and distributional semantics. Unsupervised machine learning methods for semantic parsing (e.g., Poon and Domingos, 2009; Lewis and Steedman, 2013), and for constructing knowledge bases (e.g., Etzioni et al., 2007) essentially rely on distributional information to cluster linguistic expressions that are semantically similar, but produce logical representations.

1.2 Semantics in Automatic Summarization

Perhaps surprisingly, current automatic summarization systems do not make use of rich semantic representations to determine the content and form of output summaries. Instead, they

use simple word-level statistics to determine topic words that appear in the **source text** to be summarized, then select source text sentences that contain these topic words to include in the summary.

This process of sentence **extraction** has been the dominant approach in recent summarization systems. As I will discuss in Chapters 2 and 3, these extractive summarization systems typically rely on the idea of **centrality within the source text** to determine sentence importance. That is, sentences that are representative of other sentences in the source text are preferred for inclusion in the summary.

The alternative to extraction is **abstraction**, in which novel text not found in the source text is composed in order to be included in the summary. Abstraction requires some sort of analysis of the source text, but it has several fundamental advantages over pure extraction, besides the obvious one that humans do not normally write extractive summaries.

First, adjacent sentences in an extractive summary may come from different portions of the source text or even different source documents, which may lead to problems with coherence. For example, there may be a sudden topic shift or contradictory information between the sentences. Such coherence problems may also be due to cohesive devices that no longer function when removed from their original context. For example, pronouns and discourse cues may no longer make sense because their antecedents may not be in the summary text. Consider the following excerpt from an automatically generated extractive summary about the Columbine shootings:

(1.4) *She said the school was allowing students to stay home. Columbine and Chatfield are sports rivals, but junior John Danos said he welcomed the newcomers.*

In the first sentence, it is unclear which entities are referred to by *she* or *the school*, and there are no cohesive links between the two sentences such as coreferent noun phrases or semantically coherent transitions.

Second, there is a fundamental limit to the level of compression and usefulness that can be achieved by extraction. Abstractive summaries are not only able to contain paraphrases

that condense the source text and eliminate unimportant information, they can also contain generalizations, analysis, and aggregation of information from multiple points in the source text. This is especially important in multi-document summarization, where different documents may contain conflicting or divergent information, which would be useful to point out in a summary.

The lack of robust and powerful semantic analysis techniques has caused extraction to in essence be used as a crutch, because source text sentences that are guaranteed to be grammatical and locally coherent are readily available. With recent advances in distributional semantics, however, the time is now ripe to reexamine the role of semantics in automatic summarization to progress towards abstractive summarization. Even within the extractive paradigm, better semantic modelling could be useful to detect salient concepts in the domain that should be included in the summary, as well as to prevent redundancy in the summary.

1.3 Thesis Statement

The thesis of this dissertation is that *distributional semantics can support the semantic inference that is required for robust automatic summarization.* By *robust* summarization, I mean first of all that the summarization system should be wide-coverage and easy to adapt to a new domain without a large amount of manual annotation, similar to the definition of robustness for a semantic representation. By *robustness*, I also mean that the summarization system should exhibit some of the capabilities of human summary writers to precisely convey some content in the most appropriate way, which often involves reformulating and paraphrasing the source text.

What are the ways in which robust automatic summarization depend on semantic inference? One way is paraphrase detection in order to determine when some semantic content is repeated in the source text. Especially in multi-document summarization, repetition is a good indicator of importance.

Entailment relations are useful for avoiding redundancy in summaries (Mehdad et al., 2013). For example, suppose a summarization system has already decided to include the statement:

(1.5) *Search crews determined the source of the fire which damaged five homes.*

Then, there is no reason to include any sentence that is entailed by this statement, such as:

(1.6) *The fire wrecked five homes.*

Consistency checking or its converse of contradiction detection (Condoravdi et al., 2003; de Marneffe et al., 2008) can be important, especially to certain kinds of summarization where differences of opinion may occur. For example, it is important to detect when users have differing opinions in a product review summary, or to detect when two scientific papers disagree on an issue.

Inference is important not just in determining the content of a summary, but also the form of summary sentences. In abstractive summarization, summary sentences are composed by reformulating the source text by some method of semantics-to-text natural language generation. Here, semantics plays an important role in ensuring that the inferences that can be drawn from the source text are preserved in the summary sentence, an issue that will arise in Chapter 6.

1.4 Dissertation Objectives

At a high level, this dissertation is an argument for incorporating more semantic knowledge into automatic summarization systems. I will propose novel techniques for evaluating and using distributional semantics that are inspired by properties of logical semantics, yet are applicable to arbitrary domains, unlike logical semantics. From a practical perspective, this dissertation aims to improve extractive summarization and to progress towards robust abstractive summarization. From a theoretical perspective, it shows that distributional semantics should not be relegated to the domain of lexical semantics in its evaluation and application, but should

instead be considered a full-fledged complement or alternative to logical semantics that can be usefully incorporated into NLP applications.

I enumerate below the specific objectives and contributions of the dissertation.

1. **Novel evaluation framework for distributional semantics.** I propose an evaluation framework for distributional semantics that is based on first principles about the desiderata of semantic representations that were originally proposed for logical semantics, rather than on the types of tasks that distributional semantic models were traditionally expected to perform well on. In two extrinsic evaluation settings, I demonstrate the ability of current distributional semantic approaches to support semantic inference at a phrasal level invariant of the constituent syntactic constructions.

2. **Centrality in extractive summarization systems.** I show that current summarization systems have used centrality within the source text along with sentence extraction as a proxy for the deeper semantic analysis necessary to fully solve the problem.

3. **Domain knowledge in automatic summarization.** I show that domain knowledge is an important factor in automatic summarization which has been largely ignored in recent systems. I investigate reasons that human summary writers might look beyond the source text into in-domain text, and identify a number of features that may be useful for an automatic system with the same goal. I use distributional semantics to produce better domain models which result in better summarization performance.

4. **Embedding distributional semantics into probabilistic models.** To accomplish the above, I present a novel technique to embed distributional semantic vectors into a generative probabilistic model as observed emissions. I show that this model is able to induce a structured representation of a domain better than a state-of-the-art system, and that the improved domain induction translates into better performance in automatic multi-document summarization.

5. **Using greater contexts in abstractive summarization.** I propose **sentence enhancement**, a text-to-text language generation technique that can draw from many points in the source text in order to produce a summary sentence. This contrasts with extraction or sentence compression, which operates at the level of individual sentences, and traditional sentence fusion, which operates on a small number of highly similar sentences.

1.5 Structure of the Dissertation

The following is an outline of the remaining chapters and how they meet the objectives of the dissertation.

Chapter 2 reviews existing work in automatic summarization, focusing on the prevailing assumption that importance in text can be approximated by centrality in an information space using a shallow word- or n-gram-based representation. I also examine work in the psychology of reading and in compositional distributional semantics that challenges these assumptions, providing an avenue of research for development of systems that can make better content selection decisions and, in the long run, perform robust abstractive summarization.

In Chapter 3, I study current summarization systems and examine how they differ from human summary writers. I provide a quantitative measure of centrality to demonstrate the over-reliance of current systems on it, compared to human summary writers. These results suggest that substantial improvements are unlikely to result from better optimizing centrality-based criteria. I also investigate the degree to which human summary writers produce abstractive summaries, and how these summaries may be constructed by automatic systems by considering domain knowledge. Specifically, I identify linguistic factors that are correlated with the use of in-domain text that is external to the source text in human-written summaries.

In Chapter 4, distributional semantics is examined as a potential solution to address the issues raised in the previous chapter. The first challenge to address is to properly evaluate distributional semantic models in terms of the inference decisions that they support. I propose a

novel framework for evaluating distributional semantic phrase representations, invariant to the particular syntactic constructions in the sentence. I propose two evaluation methods in relation classification and QA which reflect these goals, and apply several recent compositional distributional models to the tasks. Experimental results show that the models outperform a simple lemma overlap baseline slightly, demonstrating that distributional approaches can already be useful for tasks requiring deeper inference.

In Chapter 5, I introduce a new technique for using distributional semantics to learn about the characteristic aspects of a domain. Generative probabilistic models have been used for content modelling and automatic summarization, but are typically trained on small corpora in the target domain. Distributional semantic models contain information from the large corpora on which they are trained, and also have the potential to support complex inference decisions as shown in the Chapter 4. I introduce Distributional Semantic Hidden Markov Models, a novel variant of HMMs that integrates these two approaches by incorporating contextualized distributional semantic vectors into a generative model as observed emissions. Experiments in slot induction show that DSHMM yields improvements in learning coherent entity clusters in a domain. A subsequent extrinsic evaluation shows that these improvements are reflected in multi-document summarization.

Further use of distributional semantics and domain knowledge for abstractive summarization is explored in Chapter 6, in which I show that distributional semantics is crucial to the success of a novel sentence revision technique called **sentence enhancement**. Here, distributional semantics forms the basis of an event coreference resolution algorithm that aims to preserve the inferences that can be drawn from the revised output sentence compared to the input source text sentences.

Finally, I conclude in Chapter 7 by describing the limitations of the current work and future research directions.

1.5.1 Peer-Reviewed Publications

Several chapters in this dissertation are based on previous publications at peer-reviewed venues. In particular, Chapter 3 excluding Section 3.4 has been published as (Cheung and Penn, 2013a); Chapter 4 as (Cheung and Penn, 2012); Chapter 5 as (Cheung and Penn, 2013b); and Section 3.4 and Chapter 6 as (Cheung and Penn, 2014). However, Chapter 4 contains new results from models trained on a larger corpus, which supersede the previously published results.

Chapter 2

Centrality in Automatic Summarization

In this chapter, I discuss properties and characteristics that contribute to the definition of importance in text and how this is used to determine the contents and output of automatic summaries. The main assumption in current systems about importance is that it can be approximated using *centrality within the source text*; that is, elements of the source text that are most similar to other elements of the source text should be considered important. I consider potential bases for criticizing this assumption by examining work in the psychology of reading literature on cognitive determinants of interest and memorability. This work suggests that determining importance requires making use of some sort of background knowledge about the domain that the text falls under. In contrast, existing summarization systems largely rely on the concept of centrality to determine summary content, and I survey the variety of techniques that they use to do so.

2.1 The Design and Evaluation of Summarization Systems

The overall goal of automatic summarization is to produce a condensed version of some information source, selecting the most important information in the source to include in the summary. An example of a summary is presented in Figure 2.1.

Figure 2.1: An informative, generic, extractive summary produced by the Columbia Newsblaster system (McKeown et al., 2002).

2.1.1 Classification of Summaries

Automatic summarization can be divided into subtypes according to the specific goal and the method of producing summaries (Mani, 2001). The first distinction that can be made is in the goal of the summary. A summary can be **indicative**, containing pointers to more detailed sources of information (Figure 2.2) much like a search engine; **informative**, aiming to act as a replacement for the source; or **critical**, reviewing the source text and giving a value judgement of it.

Another dimension along which summarization systems differ is in whether they are **generic** or **focused**. A generic summary should appeal to a broad audience without any group-specific goal in mind, whereas a focused summary aims to target particular user groups or needs. Summaries can be focused in several ways. One is by user preferences, such as by placing more emphasis on particular aspects of a product in summarizing product reviews. Another is to focus by some query, such as a query about risks to journalists in articles about civilian casualties in conflicts (Figure 2.3). A third possibility is to focus by the prior information that is assumed to be known by the reader of the summary. This is exemplified by the update summarization

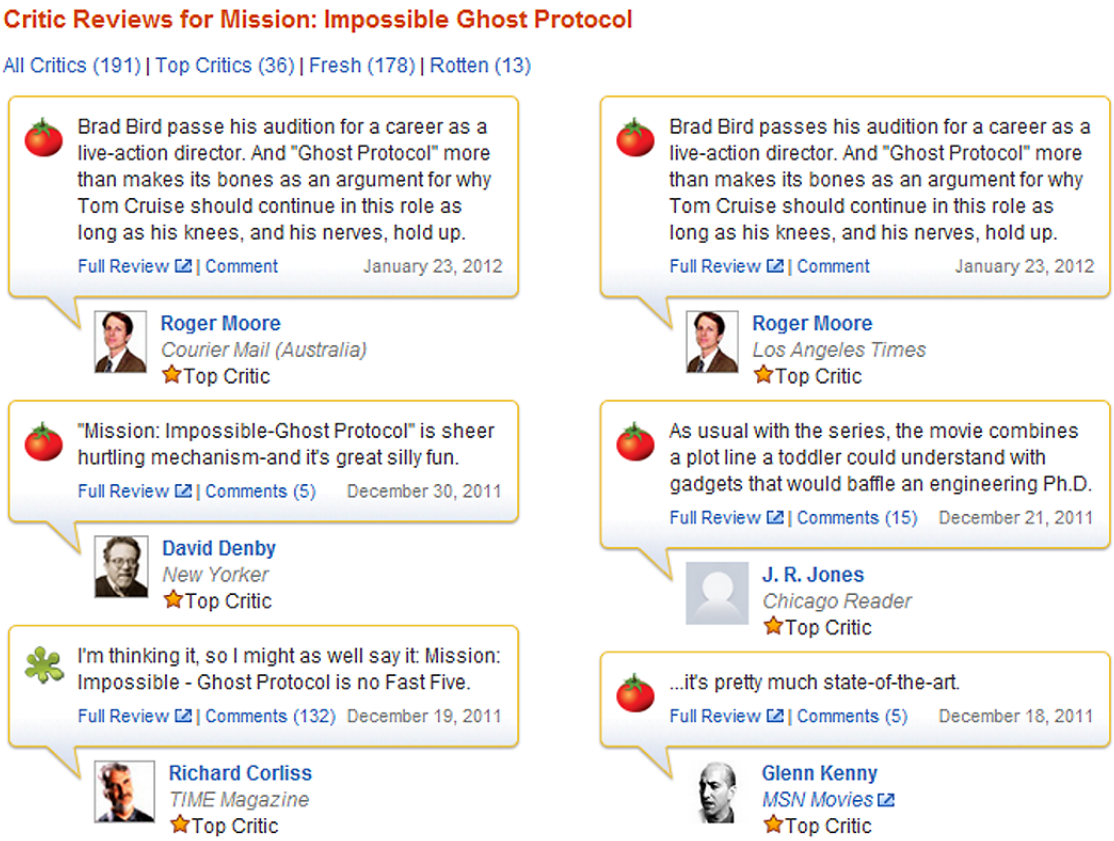


Figure 2.2: An indicative, generic, extractive summary from the movie review website Rotten Tomatoes (`http://www.rottentomatoes.com/m/mission_impossible_ghost_protocol/`).

Journalist risks *What types of dangers do journalists find themselves in and how are these situations related to their work? Are journalists specifically targetted* [sic]*? Are they endangered by the type of situations they are covering or by the content of their reports?*

(a) Summary topic

Journalists may face a variety of dangers in their work. When deployed to an area of active warfare, whether between nations or factions within a nation, they are unlikely to be personally targeted but in reporting on the battle in progress they subject themselves to danger of kidnapping, capture, torture, wounding or death. Examples are media coverage of Romania 1989, Sri Lanka 1989–90, the Gulf War in 1991, Bosnia 1992–93, and Somalia and Haiti in 1993. On the other hand, when the journalist's beat is a nation, state or region governed by autocratic or corrupt government or under the influence of powerful criminal elements, the journalist's truthful reporting puts him in danger of arrest, beating, imprisonment, assassination or execution.

(b) Excerpt from an abstractive summary

Figure 2.3: A summary topic and a human-authored, informative, topic-focused abstractive summary from DUC 2005.

tasks of the Document Understanding Conference (DUC) series[1] since 2007, in which an update summary of an event must be produced, assuming that users have already read a previous cluster of documents on the same event.

Summarization systems can also by classified by the method that is used to produce the summaries. Most current systems are **extractive** (Nenkova et al., 2011b), which means that snippets of source text are extracted and concatenated to form the output summary. **Abstractive** summarization, in contrast, involves the composition of new text not found in the source, and allows more condensed and useful summaries with aggregation and generalization as described above. Abstraction is a prerequisite of many of the desirable properties of an ideal summary like aggregation and generalization, but naturally requires a deeper analysis of the source text and a natural language generation (NLG) component. There are also specialized domains in which it is important to maintain the exact wording of the source text, such as in the legal domain when summarizing legal judgements (Farzindar and Lapalme, 2004), in which case extractive or a combination of extractive and abstractive summaries are necessary.

Lastly, summarization systems can be classified by the modality (text, speech, or multimodal) and domain of interest (for example, news text, product reviews, legal documents, biomedical documents, scientific articles). I will not focus much on the issue of adapting generic summarization methods to particular domains or on issues of summarization modality, except to comment that generic summarization systems have traditionally been tested on news corpora, and a developing research area is to extend the results to other genres, media, and languages other than English (Nenkova et al., 2011a).

Connections can be drawn between types of summaries and related research areas like text mining, information retrieval, and question answering. Generic summaries can be seen as related to text mining, where the goal is to determine what might be interesting or salient a priori without any user input or tailoring. Indicative and focused summarization are rather more like information retrieval and question answering in trying to gather information to address a

[1] DUC was renamed to the Text Analytics Conference (TAC) in 2008.

particular information need. The effectiveness of summarization is very dependent on its final presentation and deployment to the user. Thus, issues in human-computer interaction and information visualization are also pertinent to summarization system builders.

2.1.2 Steps in Summarization

Summarization can be broken down into three broad, interdependent steps: **analysis** (or **content selection**), **transformation** (or **refinement**) and **synthesis** (or **surface realization**) (Mani, 2001). Semantic understanding and inference are important for all three steps, but feature particularly prominently in the first two. During the analysis step, text understanding is required to determine the salient components to include in the summary, which involves sensibly dealing with a wide range of linguistic issues such as paraphrases, relations between entities, coreference resolution, and discourse structure. In transformation, semantic inference is required to do some sort of compaction or analysis of meaning representations, such as by aggregating the opinions and viewpoints of multiple sources, removing redundancies, generalizing conclusions and so forth, which thus profoundly influences the third step of synthesis, in which the final summary string is generated.

In purely extractive systems, the first step is the most important, as there is by definition almost no transformation or synthesis of the selected snippets aside from deciding on how to arrange them. In abstractive systems, transformation and synthesis are much more difficult.

2.1.3 Summarization Evaluation

As in all NLP tasks, proper evaluation of summarization systems is crucial in order to measure progress. Summarization evaluation is a balancing act between the competing issues of **external validity**, **ecological validity**, and cost. An evaluation is said to possess external validity if the results of the evaluation can be generalized from the particular idiosyncratic settings and dataset of the evaluation. This can be accomplished by, for example, testing on

multiple domains on many document clusters. It is ecologically valid if the evaluation closely approximates a real-world application involving summarization. Against both of these, the constraining factor of cost in time and money must be taken into account. The following popular summarization evaluation methods make different choices in balancing these factors.

Responsiveness and Quality judgements

Direct human **responsiveness judgements** are taken to be a useful indication of a summarization system's performance. Annotators give a rating, for example a score between 1 and 5, to describe how well a summary meets the specified information need. The same procedure can be applied to evaluate linguistic quality, using several questions targeting different properties of the summary. A standard set of five questions for linguistic quality have been used in DUC summarization competitions, targeting grammaticality, non-redundancy, referential clarity, focus, and structure and coherence (Figure 2.4).

Three main objections are levied against this type of judgement. The first objection is the cost in hiring and training annotators for evaluating systems. The second is that the scores are difficult to interpret, and it is unclear where exactly a summary is deficient and how to further improve a system. Thirdly, this type of evaluation assumes that people can make this type of judgement reliably and that these judgements correlate well with how well the summarization system can be used in end applications. I examine alternative evaluation methodologies which address each of these objections in turn.

Automatic Evaluation Measures

To address the issue of cost, automatic measures have been devised to compare automatic summaries to human-written ones. Most of these automatic measures focus on summary content rather than linguistic quality. This is a symptom of the current dominance of extractive summarization approaches, where within-sentence grammaticality is assured and where not very much can be done about between-sentence coherence issues. Automatic evaluation measures

1. **Grammaticality**
   The summary should have no datelines, system-internal formatting, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.

2. **Non-redundancy**
   There should be no unnecessary repetition in the summary. Unnecessary repetition might take the form of whole sentences that are repeated, or repeated facts, or the repeated use of a noun or noun phrase (e.g., "Bill Clinton") when a pronoun ("he") would suffice.

3. **Referential clarity**
   It should be easy to identify who or what the pronouns and noun phrases in the summary are referring to. If a person or other entity is mentioned, it should be clear what their role in the story is. So, a reference would be unclear if an entity is referenced but its identity or relation to the story remains unclear.

4. **Focus**
   The summary should have a focus; sentences should only contain information that is related to the rest of the summary.

5. **Structure and Coherence**
   The summary should be well-structured and well-organized. The summary should not just be a heap of related information, but should build from sentence to sentence to a coherent body of information about a topic.

Figure 2.4: Linguistic quality questions used in DUC evaluations. Annotators assign one of five possible ratings to each category, from "Very Poor" to "Very Good".

are evaluated by how well they correlate with manual responsiveness ratings.

In work on extractive systems, performance can be evaluated by comparing their selections against those of a human making the same decisions. This method does not work for comparing extractive to abstractive systems, however. Other methods compare system-generated summaries with human-written summaries directly, being agnostic to the summarization method used. The most popular of these measures is the **ROUGE** suite of evaluation measures (Lin, 2004). ROUGE is similar to the BLEU measure used in machine translation (Papineni et al., 2002) in that it compares n-gram overlaps between system and model summaries. For a set of reference summaries $\mathscr{R}$, the ROUGE-N score of a system summary S measures the n-gram

overlap for $n = \text{N}$, and is defined by

$$\text{ROUGE-N}(S) = \frac{\sum_{S \in \mathscr{R}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \mathscr{R}} \sum_{gram_n \in S} Count(gram_n)}, \tag{2.1}$$

where $gram_n$ is an n-gram in the summary and $Count_{match}(gram_n)$ is the maximum number of n-grams that appear in both the system and reference summary.

ROUGE has been criticized because it does not correlate with human judgements as well outside the domain of news text (Murray et al., 2005b) nor with performance on end tasks (McCallum et al., 2012). Also, it does not consider syntactic relations or linguistic quality.

Basic Elements (Hovy et al., 2006) is an alternative to ROUGE that considers syntactic information. A Basic Element is defined to be the head of a major syntactic constituent (noun, verb, adjective, or adverbial phrase), or a dependency triple of $(head, modifier, relation)$, such as $(throw, ball, \text{OBJ})$ to represent a ball being thrown. Basic Elements are automatically extracted from text using standard parsers. Each Basic Element is then assigned a score equal to the number of reference summaries in which it appears. In practice, Hovy et al. (2006) find that Basic Elements, ROUGE, and responsiveness judgements all correlated highly with each other in text summarization.

There are also automatic measures which do not require model summaries, but compare directly against the source text instead (Louis and Nenkova, 2009; Saggion et al., 2010). These papers use Jensen-Shannon divergence, which is a variant of Kullback-Leibler divergence that measures how different two probability distributions area. Unlike Kullback-Leibler divergence, Jensen-Shannon divergence is symmetric and always produces a finite value. In the automatic evaluations, it is used to compute the divergence between the observed word probability distributions between the source text and the summary text, with the intuition that these distributions should be similar in a good summary. For summarization of news text, these papers find good correlations in ranking the performance of summarization systems between this measure and

other evaluation measures such as responsiveness, ROUGE or the Pyramid method, a more structured evaluation method to be defined shortly.

The above methods focus on automatic evaluation of summary content. There has also been some work on automatic measures of linguistic quality. Pitler et al. (2010) investigate a wide variety of automatic measures of linguistic quality including lexical, syntactic, semantic, and discourse features. Lexical and syntactic features include *n*-gram language models and POS tag and syntactic category frequencies. Semantic features include named entity classes and cosine similarity between adjacent sentences, while discourse features include measuring cohesive devices like pronouns and discourse connectives, coreference chains, and various word and entity coherence measures. Overall, they find that using measures of similarity between adjacent sentences is indicative of linguistic quality of system summaries by correlating with human linguistic quality judgements, but that language models and entity coherence features are also important.

The Pyramid Method

The second objection of the interpretability of results has been addressed by a more structured type of evaluation for content selection called the Pyramid method (Nenkova and Passonneau, 2004). This method compares automatic and human summaries in terms of the **summary content units** (SCUs) that they contain. The main assumption of the method is that many choices of summary content are reasonable, but that the choices that are common to many human summarizers are better than the ones that only a few summarizers make or none. The method is so named because it is expected that relatively few SCUs will be expressed by all of the human-written summaries, which form the top of the pyramid, whereas there will be many SCUs that only one of the human-written summaries contains, corresponding to the bottom of the pyramid. By directly assessing the quality of the SCUs selected by a summarization system, system developers can gain more insight into the deficiencies of the current system over simple responsiveness scores.

To calculate the Pyramid score, annotators divide the content of model and system summaries into SCUs, and annotate which SCUs are expressed in each summary. Then, each SCU is given a weight equal to the number of model summaries that express this SCU, and a system summary is scored by the sum of the weights of the SCUs that they express. This score is divided by the maximum score achievable with the same number of SCUs in the original definition of the Pyramid method. Alternatively, in the modified Pyramid method, the denominator is the optimal score using the average number of SCUs found in the model summaries, which is more consistent with an evaluation setting in which the summary length is limited by the number of words.

Formally, let T_i be the set of the SCUs that occur in i model summaries; that is, it represents the ith tier of the pyramid. The numerator of the Pyramid score of a system summary to be evaluated is then defined as $d = \sum_{i=1}^{n} i \times D_i$, where D_i is the number of SCUs in the system summary that appears in the ith tier of the pyramid. The final Pyramid score is d divided by the optimal score, Max:

$$Max = \sum_{i=j+1}^{n} i \times |T_i| + j \times \left(X - \sum_{i=j+1}^{n} |T_i| \right), \tag{2.2}$$

where $j = \max_i \text{ s.t.} \sum_{t=i}^{n} |T_t| \geq X$, and X is the number of SCUs in the summary. Intuitively, j is the lowest tier in the pyramid from which SCUs are drawn in an optimal selection. In the expression above, the first term corresponds to the contribution of the SCUs drawn from the tiers above the jth tier, and the second term corresponds to the contribution of the SCUs drawn from the jth tier to the optimal score Max.

Extrinsic Evaluation

To address the third objection of ecological validity, an evaluation task should simulates the deployment of summarization technology in the real world. This type of extrinsic, task-based evaluation is more useful though much more expensive and difficult to conduct, and thus rela-

tively uncommon.

One of the largest extrinsic evaluations of an NLG system was by Reiter et al. (2003), who evaluated the effect of user tailoring on an NLG system that sends letters to people trying to stop smoking. They evaluated their system on 2553 subjects in a randomized controlled clinical setting to see if user-tailored letters increase smoking cessation rates, but they found that it did not.

Specific to summarization, there are two main classes of extrinsic task-based evaluation. The first type involves question answering and fact gathering (McKeown et al., 2005; McCallum et al., 2012, for example). In this type of evaluation, users are either given just the source text, the source text with the summary, or just the summary, and are then tested on how well they extract facts from the source text. The test can either be in the form of a quiz, which makes this a question-answering task evaluation, or it can be a timed open fact gathering task. In the latter, the quality of the assembled facts would then be evaluated as a separate step.

The second type of evaluation involves relevance assessment (Mani et al., 2002; Dorr et al., 2005, for example). In this setting, users are asked to classify the source documents either into different categories, or by whether the document is relevant to some topic. The goal of a summarization system is to improve the accuracy of classification, or more often, to reduce the time needed to do the classifications. Dorr et al. (2005) for example find that summaries can improve relevance prediction speed from more than 13 seconds per document to below 5 seconds.

2.2 The Assumption of Centrality

I next examine the assumption of **centrality** used in most current work to identify important sentences to include in an extractive summary. As a working definition, a piece of text can be said to be central within some larger collection if it is "close" to large portions of the collection. Usually, "closeness" is defined according to some proxy of information content overlap, but

it can also be defined structurally, such as by using discourse structure. I first discuss various properties of text related to centrality and the determination of text importance, then consider theories in psychology of reading which emphasize the importance of prior knowledge and context in determining cognitive interest and memorability. This stands in contrast to methods for content selection in current summarization systems, which I then review.

2.2.1 Text Properties Important in Summarization

In content selection, it is important to consider a number of properties of the source text, depending on the type of summarization. I will discuss these properties starting from the most widely applicable to the least.

Relevance is the property of being pertinent to the topic or query at hand. In generic summarization, relevance is usually quantified by how similar a piece of source text is to other pieces of source text; that is, relevance reduces to centrality. In query-focused summarization, relevance can in addition be determined relative to the query topic statement.

A number of properties pertain to how prominent a piece of source text is within the context in which it is found. **Salience** is the property of being noticeable or prominent and is taken to be the goal of generic summarization. It is often taken to be synonymous to relevance (Erkan and Radev, 2004), though the two can in fact be distinguished, as I will further clarify. A piece of information can be inherently salient because it is expected to be in the domain, such as the number of deaths in a natural disaster, or paradoxically because it is somehow unusual for that domain. Salience can also be predicted in some genres of text based on positional cues. For example, the first paragraphs of a news article tend to contain salient information and indeed, this is used as a baseline in news text summarization, but this does not hold in, say, novels. Related terms that have been used are **interestingness** or **memorability**, which seem to focus more on the effect that the source text has on the reader, such as the emotional response the text evokes or how well the information can be recalled. I will expand upon these definitions in the next section.

Other properties that have been less explored in automatic summarization focus on the expectation of the reader about the topic. **Surprise** or **counterintuitiveness** and their antonym **predictability** measures how unexpected some text is upon encountering it during reading. **Postdictability** in contrast focuses on how well some text fits into the mental picture of the reader after being read; that is, it is "the ease with which a concept's inclusion in a piece of text can be justified after the textual unit containing that concept has been read" (Upal, 2005).

While these notions are correlated, there is a subtle distinction between them. As mentioned, salience and relevance have been taken to be synonymous in previous summarization literature, but there are cases where the two diverge, particularly in the case of query-focused summarization. The divergence comes from the fact that relevance here is defined by the query and the grouping of the documents, and is external to any particular document, whereas salience often depends on document-internal properties such as the position of the text in the document.

Take the topic of "risks to journalists" (Figure 2.5) found in the Document Understanding Conference (DUC) 2005 summarization competition (Dang, 2005). The source document cluster contains instances of journalists or civilians being taken hostage or killed in various incidents around the world. The first two sentences are examples of salient text, as they are the leading sentence in their respective articles. However, only the first sentence is relevant to the topic of the summary, describing a specific instance of journalist deaths. The second sentence serves as background context to an event involving journalists, and so is not by itself very relevant. One can also find text that is not salient but relevant, as in the third sentence. The article from which this sentence is drawn is actually about another instance of hostage taking in Columbia, but this sentence is included as additional information on other hostage taking events by the same group. Lastly, there is text that is not salient or relevant, such as a description of what exactly occurred when a journalist is released from custody.

In addition to salience and relevance, other less often discussed properties can also be important, depending on the particular summarization task and evaluation method. Current

Salience	*Relevance*	*Text*
Yes	Yes	*Two Financial Times journalists, David Thomas, Natural Resources Editor, and Alan Harper, staff photographer, died on Wednesday when their car was engulfed by flames in the southern oilfields of Kuwait.* (FT911-2977)
Yes	No	*A few years ago, when I used to cover Opec conferences for the Financial Times, a bizarre event took place at one of the Geneva price-fixing meetings of the cartel.* (FT911-2786)
No	Yes	*Another Bogota newspaper, La Prensa, reported Friday that the cartel is planning to release the other five journalists, including a West German.* (LA092290-0094)
No	No	*They walked to a government visa office and were driven away in a government car.* (LA060589-0077)

Figure 2.5: Sentences exhibiting different levels of salience and relevance in the DUC 2005 topic on risks to journalists. See Figure 2.3a for the topic statement. The document number from which the sentence is drawn is given in parentheses.

approaches focus mostly on relevance and salience, because nothing in the evaluation protocol requires anything further, as discussed in Section 2.1.3. There is, however, a movement towards extrinsic evaluation where the goal of summarization is seen to be supporting decision-making or fact-recall. This question-answering approach can directly validate the utility and value of summarization, but issues not currently considered by system developers like the memorability of the selected text becomes important. In the next section, I review results in the psychology of reading that provide insights on the determinants of relevant factors like interest, memorability, and surprise.

2.2.2 Cognitive Determinants of Interest, Surprise, and Memorability

Research in the psychology of reading literature informs us that humans rely on their existing knowledge of a domain to determine what information in a new document in that domain is interesting or salient (Kintsch, 1980). Maximally interesting information tends not to be too similar to existing knowledge (and thus redundant), yet also not so dissimilar as to cause dissonance with existing knowledge about that domain. In particular, Kintsch (1980) speculates that

cognitive interest is maximized at intermediate levels of reader knowledge about the subject, as well as at intermediate levels of predictability and postdictability, as defined in the previous section.

He also defines three different types of importance of a piece of text. The first is importance for a macrostructure or schema of the domain. For example, a sentence about the location of an earthquake would be important in an article about the earthquake, as the location is one of the main slots in the schema. The second type of importance is how useful some text is for an external task such as answering questions on an exam. Kintsch sees this as an ad hoc schema, rather like the first type of importance. The third type of importance is rather different; something might be important as perceived by a reader due to the reader's internal state. For example, a reader might be interested in a particular country because of a recent visit there.

The above speculations received experimental support in later work. For example, Iran-Nejad (1987) showed that interest in a story is evoked not simply when a story is surprising, but rather when a surprise occurs, and then is resolved and explained. He tested this by constructing different versions of two stories with varying degrees of surprise and resolution of the surprise. For example, in the first scenario, a stranger is introduced in a story and either implied to be a maniac (in the high surprise setting) or not. He is then revealed to be a Good Samaritan who intervened when the actual maniac in the story attempted to kill the protagonist, thereby preventing the protagonist's death. Alternate versions with tragic endings are also constructed, with the roles of the maniac and the Good Samaritan and the implications reversed. Information about the manner of survival or death of the protagonist and at whose hands is varied depending on the degree of resolution. Judgements by annotators reading the story show that the ratings for "interestingness" in the low surprise setting is significantly lower than for medium and high surprise settings only when the incongruity is resolved. Stories with incongruity resolved were also significantly more interesting and well liked overall, regardless of the other variables.

Hidi and Baird (1988) conducted an experiment on creating interest in text to improve student learning. They composed three versions of texts about inventors which employed different

strategies to create interest, such as to add anecdotes or insert questions about the inventors that were answered after an intervening paragraph. In a fact recall task, the results were inconclusive, as the texts did not seem to increase recall of facts over a baseline method. It is possible that the added anecdotes or questions are not relevant or salient enough to aid learning, and thus are disruptive rather than helpful.

Upal (2005) proposed that the memorability of a concept is proportional to its postdictability minus its predictability, which he called "minimal counterintuitiveness". This formulation of memorability explains why intuitive concepts are not memorable (high predictability), as well as why very divergent concepts are not memorable (low postdictability). Since postdictability involves a sentence in context, much of the discussion is in the type of contexts that surround a concept and how this affects its recall. In the experiments, different versions of a story were constructed in which the prior context of certain concepts was changed in order to control the levels of predictability and postdictability. For example, one story describes a boy and a girl returning home from school who encounter strange events and concepts such as a dog composing a symphony or a talking carrot. Predictability can be increased by adding a prior context describing how dreams often violate the laws of nature. They found that participants who read the version of the story with high predictability containing more prior context recalled fewer of the target concepts. This study has ramifications for summarization. In particular, if the goal of summarization is to inform users, then the current extractive methods and the evaluation measures which compare overlap to human written summaries without considering context are not adequate. Rather, a more nuanced view based on the relation of some information to the previous context or to the background domain is needed.

Most recently, Danescu-Niculescu-Mizil et al. (2012) provide some concrete statistical measures that partially capture these insights about memorability. Using a collection of memorable quotes from an online movie database, they find that memorable quotes tend to be lexically more distinctive (i.e., they have lower lexical *n*-gram language model probability) yet syntactically less so (higher part-of-speech tag language model probability) when compared to

similar non-memorable quotes. Memorable quotes also tend to have fewer third person pronouns, more indefinite articles, and more present tense verbs. In the terminology of Upal's theory, postdictability can be seen as having a common syntactic pattern, while lack of predictability can be seen as using novel word choices and introducing new entities into the discourse (fewer pronouns and more indefinite articles).

2.2.3 Centrality in Content Selection

I now provide a survey of past, influential summarization systems as well as several more recent systems to illustrate current trends. Broadly speaking, I divide the systems into extractive systems where the primary focus is on content selection, and text simplification-based pseudo-abstractive systems where determining linguistic well-formedness is also an issue.

I first focus on the use of salience in content selection in extractive summarization. Whether or not this is explicitly acknowledged, most systems make use of the *centrality assumption* that a summary of the source text should contain the parts of it that are representative of it. In other words, an information space is defined and the units of the source text that are centrally located within this space are selected for inclusion in the summary. An additional concern is that the selected units should not be too similar to each other, so some mechanism for avoiding redundancy must also be proposed.

Maximal Marginal Relevance

The canonical system which first proposed this view of summarization is the Maximal Marginal Relevance (MMR) system of Carbonell and Goldstein (1998). Sentences with the highest MMR score are greedily selected from the source text, where the MMR score is defined to be a linear combination of a centrality-based relevance term and a redundancy penalty term.

$$MMR = \arg\max_{D_i \in R \setminus S} \lambda Sim_1(D_i, Q) - (1-\lambda) \max_{D_j \in S} Sim_2(D_i, D_j), \tag{2.3}$$

where D_i is a candidate summary segment (in practice usually a sentence), Q is a query vector, or a vector of the source text in generic summarization, Sim_1, Sim_2 are similarity measures, usually the cosine similarity. The MMR score for each sentence is recalculated after a sentence is selected in the summary.

SumBasic

Another influential word-based approach is the `SumBasic` system of Nenkova and Vanderwende (2005), where each word-type is assigned a score based on its frequency in the source text, $p(w_i) = n_i/N$, where n_i is the number of times w_i appears in the source text, and N is the total number of words in the source text. Sentences with high average word scores are selected in the summary. After a sentence is selected, the score of all the words in the sentence is updated by squaring them. Since these scores are originally probabilities that are less than one, squaring them has the effect of lowering their chance of being subsequently selected again, thereby acting to avoid redundancy.

This word-based approach can be enhanced in several ways, such as by using tf-idf weights, which nearly all current summarizers do, or using lexical chains to deal with words with related meaning (Barzilay and Elhadad, 1997).

Topic Word and Content Models

Rather than assigning a weight to every word-type as in SumBasic, many models treat the detection of topic words that are considered important as a separate step. Lin and Hovy (2000) use a log-likelihood ratio to calculate how indicative a term t_i is of a document being relevant to a topic being summarized. In particular, the probability of t_i is first estimated in three ways: using just relevant documents in the topic, p_1; using just irrelevant documents, p_2; and using all of the documents p. Then, two hypotheses are contrasted. The first, H_1, states that the probability of t_i is independent of whether the document is relevant to the topic or not; that is, only the one parameter p is needed. The second hypothesis, H_2, states that the probability

of t_i is different depending on whether the document is relevant; that is, $p_1 \gg p_2$. The log likelihood ratio of the documents given the two hypotheses is then calculated as follows:

$$\lambda = -2\log\frac{L(H_1)}{L(H_2)}. \tag{2.4}$$

The coefficient of -2 ensures that λ approximates a chi-square distribution. Terms which have a high λ value are considered to be topic words (or as the authors call them, **signature terms**), the presence of which is then used to indicate important sentences to include in a summary.

Conroy et al. (2006) builds on top of this method by including topic words from the topic description in the topic-focused DUC tasks. In particular, the probability of a term t being included in a summary given the topic τ is

$$P(t|\tau) = \frac{1}{2}q_t(\tau) + \frac{1}{2}s_t(\tau), \tag{2.5}$$

where $q_t(\tau)$ is an indicator function that is 1 iff t appears in the topic statement of τ, and $s_t(\tau)$ is another indicator function that is 1 iff t is a signature term. Sentences with high average probabilities are selected to be in a summary, after several linguistic preprocessing and redundancy removal steps. This system produces state-of-the-art ROUGE results on DUC evaluation data.

Interestingly, Conroy et al. (2006) also conducted a study on the upper bound of extractive systems by showing that an extractive system can score as well according to the ROUGE measure as human abstractors. They define an oracle score for each word equal to the probability that the word appears in a human-written model summary, and select sentences with high average oracle scores. While such a procedure does result in very high ROUGE scores, a human evaluation is needed to investigate whether these summaries are actually on par with human abstractors' summaries in quality. A similar study on the limits of extractive methods in the speech presentation domain has been done by He et al. (2000), in which various extractive summarization methods of presentations are compared. It was found that highlighting portions

of text transcripts and extracts of the audio-visual presentation as determined by the presenter were the most successful in improving quiz scores of study participants, more so than simply presenting the slides or transcripts alone.

In a similar vein, Bayesian topic models have been used to learn a distribution of word probabilities that are relevant to the topic (Daumé and Marcu, 2006; Haghighi and Vanderwende, 2009; Celikyilmaz and Hakkani-Tur, 2010). For example, Haghighi and Vanderwende (2009) propose a hierarchical generative probability model based on latent Dirichlet allocation (LDA) for this purpose. In LDA, each word is generated from one of multiple categorical distributions, which is selected based on the value of a hidden topic[2] state. The value of this topic state as well as the categorical distributions associated with each topic state are drawn for each word, sentence, or larger unit of text, depending on the specific details of the model, based on hyperparameters.

The TOPICSUM model of Haghighi and Vanderwende (2009) contains three such distributions: a background categorical distribution ϕ_B, a document-set-specific distribution ϕ_C, and a document-specific distribution ϕ_D. For each sentence in the document set, a distribution is drawn over the three topics to determine how likely each of ϕ_B, ϕ_C, and ϕ_D are used in this sentence. Training consists of learning the parameters of these distributions, and after this, the document-set-specific distribution ϕ_C is used to produce a summary. They also present an enhancement to this model called HIERSUM, which decomposes the ϕ_C distribution into multiple distributions, each representing either the general word frequencies of the document cluster or the word frequencies of subtopics within the document cluster. ROUGE and user judgement evaluations show that HIERSUM outperforms the best performing system in DUC 2007.

[2]This use of the word "topic" is specific to topic models and unrelated to the notion of the summarization topic.

Graph Centrality

Some of the above models instantiate centrality implicitly by preferring sentences with words that are frequent in the source text, but another approach makes this more explicit by treating content selection as selecting nodes from a graph-based representation of the source text. Nodes in the graph are typically text spans, and edges are weighted according to the similarity between the connected text spans.

One such system is LexRank (Erkan and Radev, 2004), which is inspired by the PageRank method of returning relevant documents in information retrieval (Page et al., 1998). In this method, similarity scores are calculated between each pair of sentences in the input text using cosine similarity of tf-idf scores.

$$sim(s_1, s_2) = \sum_{w \in s_1, s_2} \frac{tf_{w,s_1} tf_{w,s_2} idf_w^2}{||\vec{s_1}|| \times ||\vec{s_2}||} \tag{2.6}$$

Then, a threshold is set to filter out low similarity scores, resulting in an undirected graph with weighted connections between nodes. Then, the centrality of a node is determined by

$$c(s) = \sum_{t \in adj(s)} = \frac{c(t)}{deg(t)} \tag{2.7}$$

Since, this definition of centrality is recursive, PageRank's random walk algorithm is used to determine the final centrality scores. Sentences are then selected for a summary according to the centrality score, with a reranking step after each selection to avoid redundancy as in SumBasic.

Lin and Bilmes (2011) quantify centrality as coverage. In this framework, every sentence in the source text must be "covered" by a sentence in the summary, and an optimal summary is one that maximizes the coverage score. In this way, redundancy is handled automatically without requiring any extra steps, as a redundant sentence would not increase the coverage as much as a more diverse sentence would, though the authors actually found a slight improvement if

a separate term rewarding diversity in lexical choice is explicitly included. In particular, the coverage of a summary S of source text V is

$$\mathscr{L}(S) = \sum_{i \in V} \min\{\mathscr{C}_i(S), \alpha\mathscr{C}_i(V)\}, \tag{2.8}$$

where $\mathscr{C}_i$ is a coverage function defining how much the summary covers sentence i in the source text, and $\alpha\mathscr{C}_i(V)$ is a constant limit of how much coverage score each sentence in the source text can contribute. The coverage term $\mathscr{C}_i$ is defined as the sum of the tf-idf cosine similarity scores between sentences in the summary and sentence i. Because the objective function satisfies a certain monotonicity requirement (it is monotone submodular), there is a theoretical guarantee of how well a greedy selection algorithm would perform, and the authors achieve state-of-the-art ROUGE results on DUC data with this method.

While the above work represents each sentence as a node in the graph, other choices can be made, depending on the particular domain of summarization. For example, nodes can represents fragments of an e-mail conversation or dialogue acts in a meeting discussion (Carenini et al., 2008; Murray et al., 2005a). Weighted edges between nodes can be determined in various ways as above. More abstractly, nodes can represent aspects of a product, such as the zoom feature of a digital camera (Carenini et al., 2006) and the graph structure may be specified manually.

Discourse Centrality

A rather different realization of the centrality assumption is to use discourse structure. Discourse theories such as Rhetorical Structure Theory (Mann and Thompson, 1987) often assign asymmetrical relations between clauses, such that one is the *nucleus*, or more central, and others are *satellites*, or more peripheral. For example, in the passage *Jane did not want to go to the circus. She was afraid of clowns.*, the first sentence would be the nucleus, and the second sentence would be the satellite. Discourse-based summarization approaches make use of this

asymmetry to define the central portions of the source text that are assumed to be more important. For example, the algorithm of Marcu (2000) assigns scores to nodes in a discourse parse tree based on whether the node is the nucleus and the depth of the node in the tree. These nodes correspond to text spans that are ordered by this score to compose a summary.

Using Domain Knowledge

The centrality assumption means that current systems do not make much use of background knowledge or a corpus to inform content selection, as shown above. To the extent that they do so, it is usually in the form of aggregate statistics such as tf-idf scores or word probabilities.

More use of the background corpus can be found in the work of Barzilay and Lee (2004), whose system clusters sentences in related documents into coarse topics using a hidden Markov model-based content model. In this work, a sequence of hidden states takes on values representing topics in the domain. Each hidden topic state emits an observation which represents a sentence. The hidden state transitions are modelled by a categorical distribution, while the emissions are modelled by a smoothed bigram language model.

The model is trained as follows. First, a collection of articles in the same domain with associated summaries is gathered for training data. Then, a content model is trained on the articles by an EM-like iterative method. The learned content model is then applied to the summaries to determine the topics (i.e., the hidden states) that are important and likely to be in a summary in this domain. To summarize a new article, the Viterbi algorithm is first applied to determine the topics in the new article, then the sentences that are generated by the most important topics are selected for the summary.

Instead of being learned as in this work, earlier work directly encodes domain knowledge in the form of templates or schemata (Radev and McKeown, 1998; White et al., 2001). This type of system requires detailed specification and information extraction techniques to determine the major slots and slot fillers in a particular frame or scenario, such as to determine that *Afghanistan* is the slot filler for the *Location* slot in an article about earthquakes. On the other

Accidents and Natural Disasters:

1. WHAT: what happened
2. WHEN: date, time, other temporal placement markers
3. WHERE: physical location
4. WHY: reasons for accident/disaster
5. WHO_AFFECTED: casualties (death, injury), or individuals otherwise negatively affected by the accident/disaster
6. DAMAGES: damages caused by the accident/disaster
7. COUNTERMEASURES: countermeasures, rescue efforts, prevention efforts, other reactions to the accident/disaster

Figure 2.6: A sample template give in the TAC 2011 summarization task.

hand, the detailed structure of the extracted information could be passed onto a generation component to create an abstractive summary. There is recent work in automatically extracting the template for information extraction, which would eliminate the first obstacle (Chambers and Jurafsky, 2011; Cheung et al., 2013). Recent TAC summarization tasks have returned to this view of template-based summarization in an effort to encourage more linguistic analysis in summarization (Figure 2.6).

2.2.4 Abstractive Summarization

Abstractive summarization is a comparatively less researched area due to its greater difficulty. Despite the benefits of extractive summarization, there are several issues with pure extraction that must be addressed. First, taking snippets of text from multiple documents without regard for their context can be problematic, because a sentence may refer to or be connected to elements in previous sentences. For example, discourse markers like *therefore* or *because* become nonsensical out of context, and antecedents of anaphora may be lost. Second, extraction does not achieve as high a compression ratio as abstraction potentially can, simply because the pos-

sibilities for sentence realization are much greater with abstraction; for example, a sentence in the source document may contain important information, but be saddled by unnecessary detail which should not be included in an ideal summary. Third, the crucial goals of generalization and aggregation of information are not possible with extraction, because they necessarily require reasoning over multiple source sentences.

Even by current summarization evaluation methodology, abstraction offers potential advantages. Genest et al. (2009) find that human-written abstracts outperform human-created extracts as well as current automatic systems on responsiveness, linguistic quality, and Pyramid score.

Most abstractive systems focus on rewriting or simplifying source document sentences to solve the problems of dangling discourse markers and to reduce unnecessary detail and improve compression ratio. For example, work on sentence compression uses a noisy channel model and a syntactic parse tree to prune unimportant parts of the sentences while maintaining grammaticality (Knight and Marcu, 2000; Daumé and Marcu, 2002).

Sentences can also be made more complex by combining multiple sentences that contain overlapping information (**sentence fusion**). Jing and McKeown (2000) define manual rules to combine multiple source text sentences. Another option is to identify overlaps in the syntax trees followed by a linearization component to turn the tree into a summary sentence (Barzilay and McKeown, 2005; Filippova and Strube, 2008).

Besides these rewriting approaches, I have already mentioned domain-dependent template-based summarization systems as an alternative to extractive systems, but they require rich knowledge about a domain and information extraction techniques to generate a summary, possibly using a natural language generation system (Radev and McKeown, 1998; White et al., 2001; McKeown et al., 2002). There have also been limited aggregation in certain domains such as opinion summarization of products or reviews (Carenini et al., 2006). Robust abstractive summarization outside of specific domains using deep language understanding remains a distant goal for the field.

2.3 Summarizing Remarks on Summarization

This chapter reviewed current work in automatic summarization, discussing in particular the reliance of current models on the assumption of centrality to determine importance, and on word or *n*-gram-level representations to produce extractive summaries. I also discussed how such summary output is evaluated using automatic and manual means. I examined work in the psychology of reading that challenge the current paradigm of summarization. This work suggests that centrality may not be the most appropriate assumption about importance; rather, a more sophisticated connection between the source text and background knowledge must be drawn to determine salience and importance.

As discussed above, the most successful extant extractive summarization systems operate at the word or *n*-gram level, such as the topic-word model of Conroy et al. (2006). The most important reason that word-level models have been so dominant in DUC and TAC evaluations is likely the behaviour of human abstractors in generating the model summaries. As noted by Mani (2001), professional abstractors typically copy snippets of text verbatim from the source text, especially in single document summarization. Given that the ROUGE evaluation measure itself also operates at the bigram level, these two factors together contribute to the success of word-level models.

Thus, summarization systems typically ignore or do not explicitly consider the issue of how to represent the meaning of the sentence in source text above the word or bigram level, but there is in fact a large amount of work in semantics on how to do so. Ignoring the work in semantics is a missed opportunity for the summarization field for several reasons.

First, semantics work has had to grapple with the same issue of robustness versus preciseness of inference that is found in the tension between frame-based and shallow statistical approaches to summarization. Several approaches in semantics have been developed recently which try to combine the benefits of the two extremes which may be important to summarization. Second, semantic models provide tools for implementing assumptions about importance such as centrality or predictability with more sophisticated meaning representations and sim-

ilarity measures. Third, abstractive summarization will necessarily require deeper meaning representations than scores over *n*-grams to be able to fulfill its potential for generalizing and aggregating information. I return to the issue of semantic representations for summarization beginning in Chapter 4.

Chapter 3

A Case for Domain Knowledge in Automatic Summarization

Existing extractive summarization systems rely on the concept of centrality to inform their content selection decisions, as discussed in the previous chapter, and these systems have been considered state-of-the-art in recent evaluations of summarization systems. While extractive methods based on centrality have thus achieved success, abstractive methods are ultimately more desirable for reasons such as better compression ratios and the ability to aggregate and synthesize information.

In this chapter, I provide experimental support for the position that centrality is not enough to make substantial progress towards abstractive summarization that is capable of this type of semantic inference; instead, summarization systems need to make better use of domain knowledge. I present two sets of studies on the TAC 2010 guided summarization data set. In the first (Studies 1 to 3), I compare the behaviours of automatic summarizers to human summarizers by examining how the contents of the summaries relate to the source text and to in-domain articles.

Study 1 confirms that human-written **model**[1] summaries are indeed more abstractive than automatic **peer** summaries according to a quantitative measure of the degree of sentence aggregation in a summarization system. Study 2 shows that centrality-based measures are unlikely to lead to substantial progress towards abstractive summarization, because current top-performing systems already produce summaries that are more "central" than humans do. Finally, Study 3 considers how domain knowledge may be useful as a resource for an abstractive system, by showing that key parts of model summaries can be reconstructed from the source plus related in-domain documents.

In the second set of studies (Studies 4 and 5), I examine in more detail some possible reasons that human summary writers look beyond the source text when composing the summary text. Study 4 considers how elements of human written summaries that are found in the source text differ from those that are not. Study 5 identifies features that might be useful for an automatic system that mines in-domain articles for elements to incorporate into an automatic summary.

These contributions are novel in the following respects. First, previous studies have operated at the level of words or syntactic dependencies. By contrast, the present analyses are performed at the level of **caseframes**, which are shallow approximations of semantic roles that are well suited to characterizing a domain by its slots. Furthermore, this work will take a *developmental* rather than *evaluative* perspective—the goal here is not to develop a new evaluation measure as defined by correlation with human responsiveness judgements. Instead, these studies reveal useful criteria with which to distinguish (1) model from peer summaries, (2) model summary components according to whether they are found in the source text, and (3) in-domain article components according to whether they are used in the summary of the target domain instance. These findings can thus guide the development of future abstractive systems and frameworks for summarization.

[1] The summarization community uses "model" and "peer" to refer to gold-standard and automatic summarizers respectively.

3.1 Related Work

Domain-dependent template-based summarization systems have been an alternative to extractive systems which make use of rich knowledge about a domain and information extraction techniques to generate a summary, possibly using a natural language generation system (Radev and McKeown, 1998; White et al., 2001; McKeown et al., 2002). This work can be seen as a first step towards reconciling the advantages of domain knowledge with the resource-lean extraction approaches popular today.

Lin and Hovy's (2000) method discovers signature terms that appear in the source text with unusual frequency, indicating that these terms are likely important to the text. These terms are identified by a log-likelihood ratio test based on their relative frequencies in relevant and irrelevant documents. They were originally proposed in the context of single-document summarization, where they were calculated using in-domain (relevant) vs. out-of-domain (irrelevant) text. In multi-document summarization, the in-domain text has been replaced by the source text cluster (Conroy et al., 2006), thus they are now used as a form of centrality-based features. In this chapter, I use guided summarization data as an opportunity to reopen the investigation into the effect of domain, because multiple document clusters from the same domain are available.

Several studies complement the present work by examining the best possible extractive system using current evaluation measures, such as ROUGE (Lin and Hovy, 2003; Conroy et al., 2006). They found that the best possible extractive systems score higher or as highly than human summarizers, but it is unclear whether this means the oracle summaries are actually as useful as human ones in an extrinsic setting. Genest et al. (2009) asked humans to create extractive summaries, and found that they scored in between current automatic systems and human-written abstracts on responsiveness, linguistic quality, and Pyramid score. In the lecture domain, He et al. (1999; 2000) found that lecture transcripts that have been manually highlighted with key points improved students' quiz scores more than when using automated summarization techniques or when providing only the lecture transcript or slides.

Jing and McKeown (2000) manually analyzed 30 human-written summaries, and found that 19% of sentences cannot be explained by **cut-and-paste** operations from the source text. Saggion and Lapalme (2002) similarly defined a list of transformations necessary to convert source text to summary text, and manually analyzed their frequencies. Copeck and Szpakowicz (2004) found that at most 55% of vocabulary items found in model summaries occur in the source text, but they did not investigate where the other vocabulary items might be found.

3.2 Theoretical basis of the analysis

Many existing summarization evaluation methods rely on word or *n*-gram overlap measures, but these measures are not appropriate for the present analysis. Word overlap can occur due to shared proper nouns or entity mentions. Good summaries should certainly contain the salient entities in the source text, but when assessing the effect of the domain, different domain instances (i.e., different document clusters in the same domain) would be expected to contain different salient entities. Also, the realization of entities as noun phrases depends strongly on context, which would confound the analysis if coreference is not also correctly resolved, a difficult problem in its own right. Such issues are best left to other work (e.g., Nenkova and McKeown, 2003).

Domains would rather be expected to share **slots** (a.k.a. **aspects**), which require a more semantic level of analysis that can account for the various ways in which a particular slot can be expressed. Another consideration is that the structures to be analyzed should be extracted automatically. Based on these criteria, I selected **caseframes** to be the appropriate unit of analysis. A caseframe is a shallow approximation of the semantic role structure of a proposition-bearing unit like a verb, and is derived from the dependency parse of a sentence. In particular, they are $(pred, role)$ pairs, where *pred* is a proposition-bearing element, and *role* is an approximation of a semantic role with *pred* as its head (see Table 3.1 for examples).

The use of caseframes is well grounded in a variety of NLP tasks relevant to summarization

Sentence:	
At one point, two bomb squad trucks sped to the school after a backpack scare.	
Dependencies:	
$num(point, one)$	$prep\_at(sped, point)$
$num(trucks, two)$	$nn(trucks, bomb)$
$nn(trucks, squad)$	$nsubj(sped, trucks)$
$root(ROOT, sped)$	$det(school, the)$
$prep\_to(sped, school)$	$det(scare, a)$
$nn(scare, backpack)$	$prep\_after(sped, scare)$
Caseframes:	
$(speed, prep\_at)$	$(speed, nsubj)$
$(speed, prep\_to)$	$(speed, prep\_after)$

Table 3.1: A sentence decomposed into its dependency edges, and the caseframes derived from those edges that are considered (in black).

such as coreference resolution (Bean and Riloff, 2004), and information extraction (Chambers and Jurafsky, 2011), where they serve the central unit of semantic analysis. I adopt the term for terminological consistency with previous work, but note that caseframes are distinct from (though directly inspired by) the similarly named **case frames** of Case Grammar (Fillmore, 1968) and derivative formalisms such as frame semantics (Fillmore, 1982). Thus, the predicates and roles found in caseframes operate at a level close to the surface form, such that $(speed, prep\_to)$ in Table 3.1, for example, would not be further analyzed and decomposed into a form that indicates a reference to the destination of a travel action that is proceeding *quickly*. Caseframes also do not explicitly take into account the dependents or fillers of the semantic role approximations.

The following algorithm extracts caseframes from dependency parses. First, those dependency edges with a relation type of subject, direct object, indirect object, or prepositional object (with the preposition indicated) are extracted, along with their governing predicates. The governor must be a verb, event noun (as defined by the hyponyms of the WordNet EVENT synset), or nominal or adjectival predicate. Then, a series of deterministic transformations are applied

Relation	Caseframe Pair	Sim.
Degree	$(kill, dobj)$ $(wound, dobj)$	0.82
Causal	$(kill, dobj)$ $(die, nsubj)$	0.80
Type	$(rise, dobj)$ $(drop, prep\_to)$	0.81

Figure 3.1: Sample pairs of similar caseframes by relation type, and the similarity score assigned to them by the distributional model.

to the syntactic relations to account for voicing alternations, control, raising, and copular constructions.

3.2.1 Caseframe Similarity

Direct caseframe matches account for some variation in the expression of slots, such as voicing alternations, but there are other reasons different caseframes may indicate the same slot (Figure 3.1). For example, $(kill, dobj)$ and $(wound, dobj)$ both indicate the victim of an attack, but differ by the degree of injury to the victim. $(kill, dobj)$ and $(die, nsubj)$ also refer to a victim, but are linked by a causal relation. $(rise, dobj)$ and $(drop, prep\_to)$ on the other hand simply share a named entity type (in this case, numbers). To account for these issues, I measure caseframe similarity based on the distributional similarity between a pair of caseframes in a large training corpus.

First, a vector representation of each caseframe is constructed, where the dimensions of the vector correspond to the lemma of the head word that fills the caseframe in the training corpus. For example, *kicked the ball* would result in a count of 1 added to the caseframe $(kick, dobj)$ for the context word *ball*. Then, the counts are rescaled into pointwise mutual information values, which has been shown to be more effective than raw counts at detecting semantic relatedness (Turney, 2001). Similarity between caseframes is then defined by the cosine similarity between their vector representations.

For training, I used the AFP portion of the Gigaword corpus (Graff et al., 2005), which was parsed using the Stanford parser's typed dependency tree representation with collapsed

conjunctions (de Marneffe et al., 2006). For reasons of sparsity, only caseframes that appear at least five times in the guided summarization corpus are considered, and only the 3000 most common lemmata in Gigaword are used as context words.

3.2.2 An Example

The following fragment of a model summary from TAC about the **Unabomber trial** illustrates how caseframes indicate the slots in a summary:

(3.1) *In Sacramento, Theodore Kaczynski faces a 10-count federal indictment for 4 of the 16 mail bomb attacks attributed to the Unabomber in which two people were killed. If found guilty, he faces a death penalty. ... He has pleaded innocent to all charges. U.S. District Judge Garland Burrell Jr. presides in Sacramento.*

All of the slots provided by TAC for the **Investigations and Trials** domain can be identified by one or more caseframes. The DEFENDANT can be identified by $(face, nsubj)$, and $(plead, nsubj)$; the CHARGES by $(face, dobj)$; the REASON by $(indictment, prep\_for)$; the SENTENCE by $(face, dobj)$; the PLEAD by $(plead, dobj)$; and the INVESTIGATOR by $(preside, nsubj)$.

3.3 Experiments

The experiments are conducted on the data and results of the TAC 2010 summarization workshop. This data set contains 920 newspaper articles in 46 topics of 20 documents each. Ten are used in an initial guided summarization task, and ten are used in an update summarization task, in which a summary must be produced assuming that the original ten documents had already been read. All summaries have a word length limit of 100 words. I analyzed the results of the two summarization tasks separately in the experiments.

The 46 topics belong to five different categories or domains: **Accidents and natural disasters**, **Criminal or terrorist attacks**, **Health and safety**, **Endangered resources**, and **Investigations and trials**. Each domain is associated with a template specifying the type of

information that is expected in the domain, such as the participants in the event or the time that the event occurred.

This study compares the characteristics of summaries generated by the eight human summarizers with those generated by the automatic peer summaries, which are basically extractive systems. There are 43 peer summarization systems, including two baselines defined by NIST. The systems will be referred to by their ID given by NIST, which are alphabetical for the human summarizers (A to H), and numeric for the peer summarizers (1 to 43). Two peer systems (systems 29 and 43) were removed because they did not generate any summary text in the workshop, presumably due to software problems. For each measure to be considered, I compare the average among the human-written summaries (the **model average**) to the average among the 41 peer summarizers (the **peer average**. In addition, I also compare against three individual peer systems, which represent the state of the art in automatic summarization according to current evaluation methods. These systems are all primarily extractive, like most of the systems in the workshop:

Peer 16 This system scored the highest in responsiveness scores on the original summarization task and in ROUGE-2, responsiveness, and Pyramid score in the update task.

Peer 22 This system scored the highest in ROUGE-2 and Pyramid score in the original summarization task.

Peer 1 The NIST-defined baseline, which is the leading sentence baseline from the most recent document in the source text cluster. This system scored the highest on linguistic quality in both tasks.

3.3.1 Study 1: Sentence Aggregation

I first confirm that human summarizers are more prone to sentence aggregation than system summarizers, showing that abstraction is indeed a desirable goal. To do so, I propose a measure

Condition	Initial	Update
Model average	1.58	1.57
Peer average	1.06	1.06
Peer 1	1.00	1.00
Peer 16	1.04	1.04
Peer 22	1.08	1.09

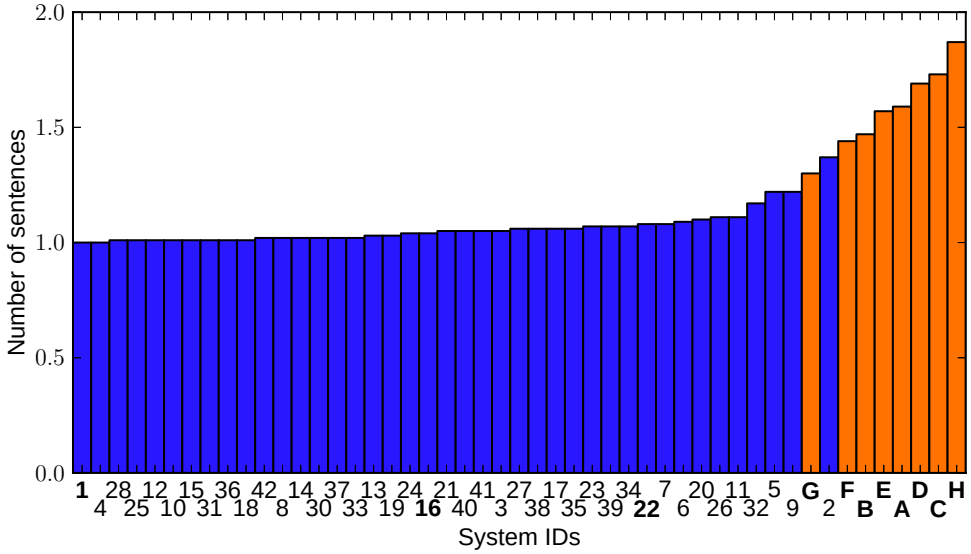
Table 3.2: The average number of source text sentences needed to cover a summary sentence. The model average is statistically significantly different from all the other conditions, $p < 10^{-7}$ (Study 1).

to quantify the degree of sentence aggregation exhibited by a summarizer, which I call **average sentence cover size**. This is defined to be the minimum number of sentences from the source text needed to cover all of the caseframes found in a summary sentence (for those caseframes that can be found in the source text at all), averaged over all of the summary sentences. Purely extractive systems would thus be expected to score 1.0, as would systems that perform text compression by removing constituents of a source text sentence. Human summarizers would be expected to score higher, if they actually aggregate information from multiple points in the source text.

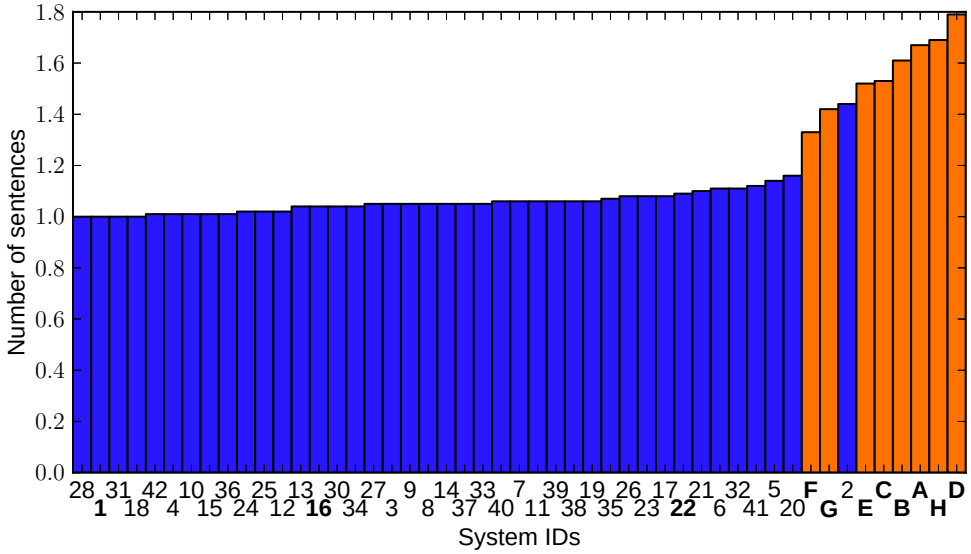
To illustrate, suppose I assign arbitrary indices to caseframes, a summary sentence contains caseframes $\{1,2,3,4,5\}$, and the source text contains three sentences with caseframes, which can be represented as a nested set $\{\{1,3,4\},\{2,5,6\},\{1,4,7\}\}$. Then, the summary sentence can be covered by two sentences from the source text, namely $\{\{1,3,4\},\{2,5,6\}\}$.

This problem is actually an instance of the minimum set cover problem, in which sentences are sets, and caseframes are set elements. Minimum set cover is NP-hard in general, but the standard integer programming formulation of set cover sufficed for this data set; I used ILOG CPLEX 12.4's mixed integer programming mode to solve all the set cover problems optimally.

Results Figure 3.2 shows the ranking of the summarizers by this measure. Most peer systems have a low average sentence cover size of close to 1, which reflects the fact that they are



(a) Initial guided summarization task



(b) Update summarization task

Figure 3.2: Average sentence cover size: the average number of sentences needed to generate the caseframes in a summary sentence (Study 1). Model summaries are shown in darker bars. Peer system numbers that I focus on are in bold.

Topic: Unabomber trial
(*charge*, *dobj*), (*kill*, *dobj*),
(*trial*, *prep_of*), (*bombing*, *prep_in*)
Topic: Mangrove forests
(*beach*, *prep_of*), (*save*, *dobj*)
(*development*, *prep_of*), (*recover*, *nsubj*)
Topic: Bird Flu
(*infect*, *prep_with*), (*die*, *nsubj*)
(*contact*, *dobj*), (*import*, *prep_from*)

Figure 3.3: Examples of signature caseframes found in Study 2.

purely or almost purely extractive. Human model summarizers show a higher degree of aggregation in their summaries. The averages of the tested conditions are shown in Table 3.2, and the differences between the model average and the other conditions are statistically significant. Peer 2 shows a relatively high level of aggregation despite being an extractive system. Upon inspection of its summaries, it appears that Peer 2 tends to select many datelines, and without punctuation to separate them from the rest of the summary, the automatic analysis tools incorrectly merged many sentences together, resulting in incorrect parses and novel caseframes not found in the source text.

3.3.2 Study 2: Signature Caseframe Density

Study 1 shows that human summarizers are more abstractive in that they aggregate information from multiple sentences in the source text, but how is this aggregation performed? One possibility is that human summary writers are able to pack a greater number of salient caseframes into their summaries. That is, humans are fundamentally relying on centrality just as automatic summarizers do, and are simply able to achieve higher compression ratios by being more succinct. If this is true, then sentence fusion methods over the source text alone might be able to solve the problem. Unfortunately, I show that this is false and that system summaries are actually more central than model ones.

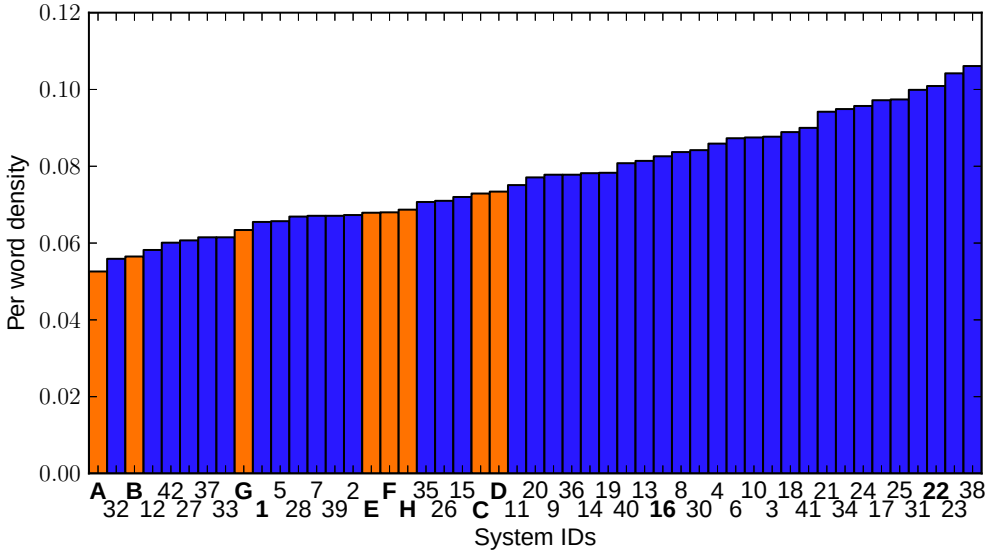
Condition	Initial	Update
Model average	0.065	0.052
Peer average	0.080*	0.072*
Peer 1	0.066	0.050
Peer 16	0.083*	0.085*
Peer 22	0.101*	0.084*

Table 3.3: Signature caseframe densities for different sets of summarizers, for the initial and update guided summarization tasks (Study 2). *: Statistically significant difference against the model average at $p < 0.005$.

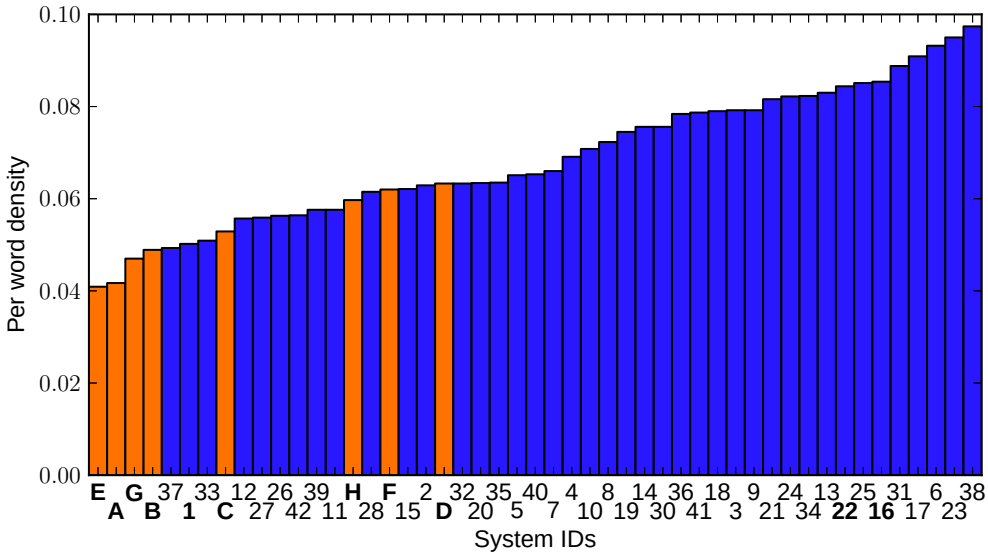
To extract topical caseframes, I use Lin and Hovy's (2000) method of calculating signature terms, but extend the method to apply it at the caseframe rather than the word level. I follow Lin and Hovy (2000) in using a significance threshold of 0.001 to determine signature caseframes[2]. Figure 3.3 shows examples of signature caseframes for several topics. Then, I calculate the **signature caseframe density** of each of the summarization systems. This is defined to be the number of signature caseframes in the set of summaries divided by the number of words in that set of summaries.

Results Figure 3.4 shows the density for all of the summarizers, in ascending order of density. As can be seen, the human abstractors actually tend to use fewer signature caseframes in their summaries than automatic systems. Only the leading baseline is indistinguishable from the model average. Table 3.3 shows the densities for the conditions that I described earlier. The differences in density between the human average and the non-baseline conditions are highly statistically significant, according to paired two-tailed Wilcoxon signed-rank tests for the statistic calculated for each topic cluster.

These results show that human abstractors do not merely repeat the caseframes that are indicative of a topic cluster or use minor grammatical alternations in their summaries. Rather, a genuine sort of abstraction or distillation has taken place, either through paraphrasing or



(a) Initial guided summarization task



(b) Update summarization task

Figure 3.4: Density of signature caseframes (Study 2).

Threshold	**0.9**		**0.8**	
Condition	**Init.**	**Up.**	**Init.**	**Up.**
Model average	0.066	0.052	0.062	0.047
Peer average	0.080	0.071	0.071	0.063
Peer 1	0.068	0.050	0.060	0.044
Peer 16	0.083	0.086	0.072	0.077
Peer 22	0.100	0.086	0.084	0.075

Table 3.4: Density of signature caseframes after merging to various thresholds for the initial (**Init.**) and update (**Up.**) summarization tasks (Study 2).

semantic inference, to transform the source text into the final informative summary.

Merging Caseframes A natural question to ask is if simple paraphrasing could account for the above results; it may be the case that human summarizers simply replace words in the source text with synonyms. To account for this, I merged similar caseframes into clusters according to the distributional semantic similarity defined in Section 3.2.1, and then repeated the previous experiment. I chose two relatively high levels of similarity (0.8 and 0.9), and used complete-link agglomerative (i.e., bottom-up) clustering to merge similar caseframes. That is, each caseframe begins as a separate cluster, and the two most similar clusters are merged at each step until the desired similarity threshold is reached. Cluster similarity is defined to be the minimum similarity (or equivalently, maximum distance) between elements in the two clusters; that is, $\max_{c \in C_1, c' \in C_2} -sim(c, c')$. Complete-link agglomerative clustering tends to form coherent clusters where the similarity between any pair within a cluster is high (Manning et al., 2008).

Cluster Results Table 3.4 shows the results after the clustering step, with similarity thresholds of 0.9 and 0.8. Once again, model summaries contain a lower density of signature caseframes. The statistical significance results are unchanged. This indicates that simple paraphras-

[2]Other thresholds did not produce substantially different results.

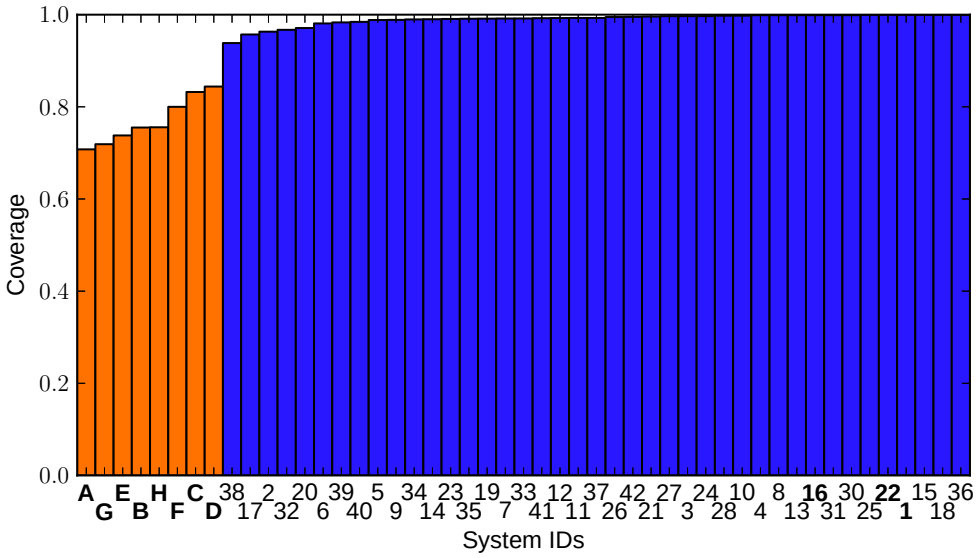
ing alone cannot account for the difference in the signature caseframe densities, and that some deeper abstraction or semantic inference has occurred.

Note that a lower density of signature caseframes does not necessarily correlate with a more informative summary. For example, some automatic summarizers are comparable to the human abstractors in their relatively low density of signature caseframes, but these are in fact the worst performing summarization systems by all measures in the workshop, and they are unlikely to rival human abstractors in any reasonable evaluation of summary informativeness. It does, however, appear that further optimizing centrality-based measures alone is unlikely to produce better informative summaries, even if the summary is analyzed at a syntactic/semantic rather than lexical level.

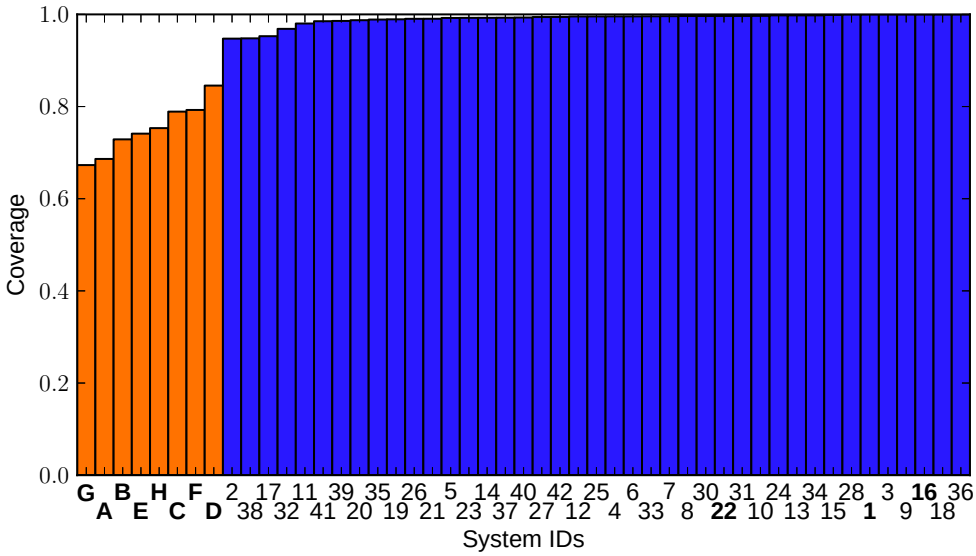
3.3.3 Study 3: Summary Reconstruction

The above studies show that the higher degree of abstraction in model summaries cannot be explained by better compression of topically salient caseframes alone. I now switch perspectives to ask how model summaries might be automatically generated at all. I will show that they cannot be reconstructed solely from the source text, extending Copeck and Szpakowicz (2004)'s result to caseframes. However, I also show that if articles from the same domain are added, reconstruction then becomes possible. I define **caseframe coverage** to measure the degree to which a model summary can be reconstructed from some reference set. Specifically, this is the proportion of caseframes in a summary that is contained by the reference set. This is thus a score between 0 and 1. Unlike in the previous study, it is necessary to consider the full set of caseframes, not just signature caseframes, because the goal now is to create a hypothesis space from which it is in principle possible to generate the model summaries.

Results The results of calculating caseframe coverage with respect to the source text alone are shown in Figure 3.5. As expected, automatic systems show close to perfect coverage, because of their basically extractive nature, while model summaries show much lower coverage.



(a) Initial guided summarization task



(b) Update summarization task

Figure 3.5: Coverage of summary text caseframes in source text (Study 3).

Condition	Initial	Update
Model average	0.77	0.75
Peer average	0.99	0.99
Peer 1	1.00	1.00
Peer 16	1.00	1.00
Peer 22	1.00	1.00

Table 3.5: Coverage of caseframes in summaries with respect to the source text. The model average is statistically significantly different from all the other conditions, $p < 10^{-8}$ (Study 3).

These statistics are summarized by Table 3.5. These results present a fundamental limit to extractive systems, and also text simplification and sentence fusion methods based solely on the source text.

The Impact of Domain Knowledge How might automatic summarizers be able to acquire these caseframes from other sources? Traditional systems that perform semantic inference do so from a set of known facts about the domain in the form of a knowledge base, but as I have shown, most extractive summarization systems do not make much use of in-domain corpora. As a first approximation to having an in-domain knowledge base, I examined whether adding in-domain text to the source text could improve coverage.

Recall that the 46 topics in TAC 2010 are categorized into five domains. To calculate the impact of domain knowledge, I now add all the documents that belong in the same domain as the source text to the reference set when calculating coverage. To ensure that coverage does not increase simply due to increasing the size of the reference set, I compare to the baseline of adding the same number of documents that belong to another domain. As shown in Table 3.6, the effect of adding more in-domain text on caseframe coverage is substantial, and noticeably more than using out-of-domain text. In fact, nearly all caseframes can be found in the expanded set of articles. The implication of this result is that it may be possible to generate better summaries by mining in-domain text for relevant caseframes.

Reference corpus	Initial	Update
Source text only	0.77	0.75
+out-of-domain	0.91	0.91
+in-domain	0.98	0.97

Table 3.6: The effect on caseframe coverage of adding in-domain and out-of-domain documents. The difference between adding in-domain and out-of-domain text is significant $p < 10^{-3}$ (Study 3).

3.4 Why Source-External Elements?

I argued above that current extractive state-of-the-art summarization systems rely too heavily on notions of information centrality and do not make enough use of domain knowledge. As shown by Study 3, one possible direction is to incorporate elements from in-domain articles into the summary. Study 3 and other previous studies on cut-and-paste summarization thus (Jing and McKeown, 2000; Saggion and Lapalme, 2002) investigate the operations that human summarizers perform on the source text in order to produce the summary text. While such studies elucidate the *mechanisms* by which such source text modification occurs, they leave unresolved the *reasons* why such techniques are required in the first place.

What previous studies lack is a detailed analysis of the factors surrounding why human summary writers use non-source-text elements in their summaries, and how these may be automatically identified in the in-domain text. In this section, I supply such an analysis and provide evidence that human summary writers actually do incorporate elements external to the source text for a reason; namely, that these elements are more specific to the semantic content that they wish to convey. I also identify a number of features that may be useful to future systems for automatically identifying which of these elements in in-domain text may be used in a summary.

Because the focus in this section has shifted from characterizing the relationship between a summary's content and its domain towards how an automatic system might identify elements that are external to the source text, I expand the definition of caseframes in the following studies to include all relation types, not just verb complements and prepositional objects. So,

constructions such as attributive adjectives (e.g. $(computer, amod)$) would be captured.

I divide my analyses into two studies. In the provenance study (Study 4), I divide the caseframes in human-written summaries according to whether they are found in the source text (**source-internal**) or not (**source-external**). In the domain study (Study 5), I divide in-domain caseframes according to whether they are used in a human-written summary (**gold-standard**) or not (**not gold-standard**).

3.4.1 Study 4: Provenance Study

I compare the characteristics of gold-standard caseframes according to their provenance; that is, are they found in the source text itself? The question of interest here is why human summarizers need to look beyond the source text at all for caseframes when writing their summaries. I will provide evidence that they do so because they can find predicates that are more appropriate to the content that is being expressed according to two quantitative measures.

Predicate Provenance

Source-external caseframes may be external to the source text for two reasons. Either the predicate is found in the source text, but the relation is not found with that particular predicate, or the predicate itself may be external to the source text altogether. If the former is true, then perhaps there is little need to look beyond the source text after all. I thus compute the proportion of source-external caseframes where the predicate already exists in the source text.

I find that in 2413 of the 4745 source-external caseframes (or 51%), the predicate can be found in the source text. This indicates that an abstractive summarization method based on extending the source text by expanding predicates with relations not necessarily found in the source text could already capture some of the source-external caseframes in its hypothesis space.

	Average freq (millions)
Source-internal	1.77 (1.57, 2.08)
Source-external	1.15 (0.99, 1.50)

(a) The average predicate frequency of source-internal vs. source-external gold-standard predicates in an external corpus.

	Arg entropy
Source-internal	7.94 (7.90, 7.97)
Source-external	7.42 (7.37, 7.48)

(b) The average argument entropy of source-internal vs. source-external PR pairs in bits.

Table 3.7: Results of the provenance study. 95% confidence intervals are estimated by the bootstrap method and indicated in parentheses.

Predicate Frequency

What factors then can account for the remaining predicates that are not found in the source text at all? The first such factor I identify is the frequency of the predicates. Here, I take frequency to be the number of occurrences of the predicate in an external corpus; namely the Annotated Gigaword, which gives us a proxy for the specificity or informativeness of a word. In this comparison, I take the set of predicates in human-written summaries, divide them according to whether they are found in the source text or not, and then look up their frequency of appearance in the Annotated Gigaword corpus.

As Table 3.7a shows, the predicates that are not found in the source text consist of significantly less frequent words on average (Wilcoxon rank-sums test, $p < 10^{-17}$). This suggests that human summary writers are motivated to use source-external predicates, because they are able to find a more informative or apposite predicate than the ones that are available in the source text.

Entropy of Argument Distribution

Another measure of the informativeness or appropriateness of a predicate is to examine the range of arguments that it tends to take. A more generic word would be expected to take a

wider range of arguments, whereas a more particular word would take a narrower range of arguments, for example those of a specific entity type.

I formalize this notion by measuring the entropy of the distribution of arguments that a caseframe takes as observed in Annotated Gigaword. Given frequency statistics $f(p,r,a)$ of predicate p taking an argument word a in relation r, I define the argument distribution of caseframe (p,r) as:

$$P(a|p,r) = f(p,r,a)/\sum_{a'} f(p,r,a') \tag{3.2}$$

I then compute the entropy of $P(a|p,r)$ for the gold-standard caseframes, and compare the average argument entropies of the source-internal and the source-external subsets.

Table 3.7b shows the result of this comparison. Source-external caseframes exhibit a lower average argument entropy, taking a narrower range of possible arguments. Together these two findings indicate that human summary writers look beyond the source text not just for the sake of diversity or to avoid copying the source text; they do so because they can find predicates that are more specifically convey some desired semantic content.

3.4.2 Study 5: Domain Study

The final study that I perform is to examine how to distinguish those source-external caseframes in in-domain articles that are used in a summary from those that are not. For this study, I rely on the topic category divisions in the TAC 2010 data set, using all of the documents of the same topic category as the target document cluster as the in-domain text. The contribution of this study is to show the importance of better semantic understanding for developing a text-to-text generation system that uses in-domain text, and to identify a potentially useful feature for training such a system.

	N	NN sim
GS	2202	0.493 (0.486, 0.501)
Non-GS	789K	0.443 (0.442, 0.443)

(a) Average similarity of gold-standard (GS) and non-gold-standard (non-GS) caseframes to the nearest neighbour in the source text.

	N	Freq. (millions)	Fecundity
GS	1568	2.44 (2.05, 2.94)	21.6 (20.8, 22.5)
non-GS	268K	0.85 (0.83, 0.87)	6.43 (6.41, 6.47)

(b) Average frequency and fecundity of GS and non-GS predicates in an external corpus. The differences are statistically significant.

Table 3.8: Results of the domain study. 95% confidence intervals are given in parentheses.

Similarity to Nearest Source-text Neighbour

I examine whether distributional similarity may be used to determine whether a source-external caseframe may be used in a summary by measuring its similarity to the nearest caseframe in the source text. To determine the similarity between two caseframes, I compute the cosine similarity between their vector representations. The vector representation of a caseframe is the concatenation of a context vector for the predicate itself and a selectional preferences vector for the caseframe; that is, the vector of counts with elements $f(p,r,a)$ for fixed p and r. As before, these vectors are trained from the Annotated Gigaword corpus.

The average nearest-neighbour similarities of PR pairs are shown in Table 3.8a. While the difference between the gold-standard and non-gold-standard caseframes is indeed statistically significant, the magnitude of the difference is not large. This illustrates the challenge of mining source-external text for abstractive summarization, and demonstrates the need for a more structured or detailed semantic representation in order to determine the caseframes that would be appropriate.

Frequency and fecundity

We also explore several features that would be relevant to identifying predicates in in-domain text that are used in the automatic summary. This is a difficult problem, as less than 0.6% of such predicates are actually used in the source text. As a first step, we consider several simple measures of the frequency and characteristics of the predicates.

The first measure that we compute is the average predicate frequency of the gold-standard and non-gold-standard predicates in an external corpus, as in Section 3.4.1. A second, related measure is to compute the number of possible relations that may occur with a given predicate. We call this measure the **fecundity** of a predicate. Both of these are computed with respect to the external Annotated Gigaword corpus, as before.

As shown in Table 3.8b, there is a dramatic difference in both measures between gold-standard and non-gold-standard predicates in in-domain articles. Gold-standard predicates tend to be more common words compared to non-gold-standard ones. This result is not in conflict with the result in the provenance study that source-external predicates are less common words. Rather, it is a reminder that the background frequencies of the predicates matter, and must be considered together with the semantic appropriateness of the candidate word.

3.5 Summary and Discussion

In this chapter, I have argued for the use of domain knowledge in summarization in two series of studies. In the first, I distinguish human-written informative summaries from the summaries produced by current systems. The studies are performed at the level of caseframes, which are able to characterize a domain in terms of its slots. First, I confirm that model summaries are more abstractive and aggregate information from multiple source text sentences. Then, I show that this is not simply due to summary writers packing together source text sentences containing topical caseframes to achieve a higher compression ratio, even if paraphrasing is taken into account. Indeed, model summaries cannot be reconstructed from the source text

alone. However, the results are also positive in that nearly all model summary caseframes can be found in the source text together with some in-domain documents.

Then, in the second series of studies, I investigate the reasons that human summary writers look beyond the source text, and show that they do so in order to find predicates that are better able to convey some intended content. I also identify several features that might be useful for determining which caseframes from in-domain text outside of the source text might be used in a summary of some target domain instance.

Current summarization systems have been heavily optimized towards centrality and lexical-semantical reasoning, but the field is nearing the bottom of the barrel. Domain inference, on the other hand, and a greater use of in-domain documents as a knowledge source for domain inference, are very promising indeed. Mining useful caseframes for a sentence fusion-based approach has the potential, as these experiments have shown, to deliver results in just the areas where current approaches are weakest.

Chapter 4

Compositional Distributional Semantics

If neither logical semantics nor n-grams are ideal for automatic summarization systems, then what semantic representation is? This chapter examines distributional semantic models as a potential tool for complex NLP tasks. I first review a number of recent distributional semantic models which attempt to construct representations for linguistic units larger than that of single words. Then, I discuss problems with current evaluations of such semantic models, and propose a novel evaluation framework for distributional semantics based on first principles about the function of semantic models in general. I describe experiments using this framework which demonstrate the potential of current distributional semantic models, and serve as the basis for further experiments in Chapter 5.

4.1 Compositionality and Co-Compositionality in Distributional Semantics

Distributional semantics takes the view that a word's meaning can be characterized by the contexts in which it appears, which is known as the **distributional hypothesis** (Harris, 1954). Such models represent word meaning as one or more high-dimensional vectors which capture the lexical and syntactic contexts of the word's occurrences in a training corpus. For example,

the vector representation of a word like *cat* might have high values in the dimensions corresponding to *purr* or *rat*, but low values in dimensions corresponding to unrelated words like *democracy* or *insightful*.

Both of these views of semantics have influenced and inspired a recent line of work called **compositional distributional semantics** (CDS). These models of semantics take the distributional hypothesis as a starting point to construct vector representations of words, but attempt to compositionally build representations of phrases and sentences from them.

The idea of compositionality has been central to understanding contemporary natural language semantics from an historiographic perspective. Compositionality is a natural way to construct representations of linguistic units larger than a word, and it has a long history in logical semantics for dealing with argument structure and assembling rich semantical expressions of the kind found in predicate logic. The idea is often credited to Frege, although in fact Frege had very little to say about compositionality that had not already been repeated since the time of Aristotle (Hodges, 2005). The modern notion of compositionality took shape primarily with the work of Tarski (1956), who was actually arguing that a central difference between formal languages and natural languages is that natural language is not compositional. This in turn was the *"the contention that an important theoretical difference exists between formal and natural languages,"* that Richard Montague so famously rejected (Montague, 1974). Compositionality also features prominently in Fodor and Pylyshyn's (1988) rejection of early connectionist representations of natural language semantics.

A related idea is **co-compositionality**, which is the idea that the meaning of words that are part of the same phrase mutually influence each other (Pustejovsky, 1991, 2000). This is canonically illustrated by showing that a direct object "selects" the sense of the verb that is the complement to, as in the following example by Pustejovsky:

(4.1) *John cut the bread.*

(4.2) *John cut the string.*

(4.3) *John cut his hair.*

(4.4) *John cut his finger.*

Here, the verb *cut* takes on distinct senses meaning to separate, to shorten, to open, or to slice, depending on the direct object. In distributional semantics, this effect is modelled by contextualizing the vector representations of words by their neighbours.

4.1.1 Several Distributional Semantic Models

The standard method of training a distributional semantic model is to first create a term-context matrix, in which rows correspond to target words and columns correspond to context words. Training begins by counting context words that appear within a context window and updating the corresponding cells in the matrix. These counts may then be rescaled into general correlation measures of the association between the target and context word, such as by using pointwise mutual information scaling. The rows in the resulting matrix are then the vector representations of the target words. From this basis, various compositional and co-compositional models can be derived. I describe below several that will be used in later experiments.

The Simple Vector Space Model

Mitchell and Lapata (2008) (M&L) propose a framework for compositional distributional semantics using the standard term-context vector space word representation. A phrase is represented as a vector of context-word counts (actually, values scaled by computing pointwise mutual information), which is derived compositionally by a function over constituent vectors, as described by the following equation:

$$\mathbf{p} = f(\mathbf{u}, \mathbf{v}, R, K) \tag{4.5}$$

u and **v** are vector representations of the constituents which compose into vector **p** according to relation R with additional background knowledge K, and f is the composition function. While

the framework is quite general, the instantiations of f that they test assume component-wise independence, with the best performing models being component-wise multiplication and a combined model of multiplication and addition. In the experiments, I use M&L to refer to the model instances that use component-wise operators.

Syntax-Modulated Models

M&L use a bag-of-words context representation which ignores syntactic relations and is insensitive to word-order and hence voicing alternations. Erk and Padó (2008) (E&P) introduce a structured vector space model for co-compositional effects which uses syntactic dependencies by modelling words' selectional preferences. The vector representation of a word in context is modulated by the **inverse selectional preferences** of its dependents, and the selectional preferences of its head. For example, suppose *catch* occurs with a dependent *ball* in a direct object relation. The vector for *catch* would then be influenced by the inverse direct object preferences of *ball* (e.g. *throw*, *organize*), and the vector for *ball* would be influenced by the selectional preferences of *catch* (e.g. *cold*, *drift*). More formally, given a dependency between words a and b in a relation r, a distributional representation of a, v_a the representation of a in context, a', is given by

$$a' = v_a \odot R_b^{-1}(r) \tag{4.6}$$

$$R_b^{-1}(r) = \sum_{c:f(b,r,c)>\theta} f(b,r,c) \cdot v_c, \tag{4.7}$$

where $R_b^{-1}(r)$ is the vector describing the inverse selectional preference of word b in relation r, $f(b,r,c)$ is the frequency of the dependency triple headed by c with dependent b in relation r, θ is a frequency threshold to weed out low-frequency dependency triples, and $\odot$ is a vector combination operator for contextualization, which is component-wise multiplication in their work.

Dinu and Lapata (2010) (D&L) assume that there exists a global, abstract set of senses or semantic primitives, and that the meaning of a word can be modelled as a mixture of these latent senses. In this model, the vector for a word t_i in the context of a word c_j is modelled by

$$v(t_i, c_j) = P(z_1|t_i, c_j), ..., P(z_K|t_i, c_j) \tag{4.8}$$

where $z_{1...K}$ are the latent senses. By making independence assumptions and decomposing probabilities, training becomes a matter of estimating the probability distributions $P(z_k|t_i)$ and $P(c_j|z_k)$ from data. While Dinu and Lapata (2010) describe two learning algorithms to do so, based on non-negative matrix factorization and latent Dirichlet allocation, the performances are similar.

4.1.2 Other Distributional Models

Turney and Pantel (2010) survey various types of vector space models and applications thereof in computational linguistics. I describe below a number of other word- or phrase-level distributional models.

Thater et al. (2010) (TFP) propose an alternative model that is sensitive to selectional preferences, but whereas E&P represent each (inverse) selectional preference with a separate vector, TFP's model encodes the selectional preferences in a single vector directly using frequency counts. Furthermore, TFP consider selectional preferences to two degrees. For example, the vector for *catch* might contain a dimension labelled (OBJ, OBJ^{-1}, *throw*), which indicates the strength of connection between the two verbs through all of the co-occurring direct objects which they share. Baroni and Lenci (2010) define **Distributional Memory** to be a third-order tensor of dependency path triples consisting of two arguments and a linking word, such as (*marine, own, gun*), each associated with a score derived from a training corpus. They project these tensors down to matrix subspaces in various ways and test them on a variety of semantic tasks.

The syntax-modulated approaches deal with polysemy and homonymy implicitly by using a dependency context. Another approach is to explicitly model the multiple senses. The **multi-prototype** approach determines top-down a number of senses for each word, and then clusters the occurrences of the word (Reisinger and Mooney, 2010) into these senses. A prototype vector is created for each of these sense clusters. When a new occurrence of a word is encountered, it is represented as a combination of the prototype vectors, with the degree of influence from each prototype determined by the similarity of the new context to the existing sense contexts. In contrast, the bottom-up **exemplar**-based approach assumes that each occurrence of a word expresses a different "sense" of the word. The most similar senses of the word are activated when a new occurrence of it is encountered and combined, for example with a kNN algorithm (Erk and Padó, 2010).

The above work assumes each dimension in the feature vector corresponds to a context word. In contrast, Washtell (2011) uses potential paraphrases directly as dimensions in his *expectation vectors*. Unfortunately, this approach does not outperform various context word-based approaches in two phrase similarity tasks.

Dimensionality reduction methods can also produce word vector representations with a low number of dimensions amenable to fast processing. Singular value decomposition is a popular approach to produce condensed versions of term-context matrices with minimal squared error loss in reconstruction error. Dhillon et al. (2011) apply canonical correlation analysis to produce low-dimensional contextualized word representations which are successful when used as features for named entity recognition and chunking.

Other Composition Operators In terms of the vector composition function, component-wise addition and multiplication are the most popular in recent work, but there also exist a number of other vector space-based composition operators, such as tensor product and convolution product, which are reviewed by Widdows (2008). Similarly, instead of vector space representations, one could also use a matrix space representation with its much more expres-

sive matrix operators (Rudolph and Giesbrecht, 2010). So far, however, this has only been applied to specific syntactic contexts like adjective-noun compositions (Baroni and Zamparelli, 2010; Guevara, 2010) and verb-noun compositions (Grefenstette and Sadrzadeh, 2011), or tasks (Yessenalina and Cardie, 2011).

Neural networks have been used to learn both the representation and the composition function. In these models, one neural network is used in a pre-training step to learn word representations (Bengio et al., 2006; Collobert and Weston, 2008). Then, the learned representations are fed as input into a subsequent network that learns and constructs representations for phrases. This second model is typically a recursive neural network, in which a set of nodes in a neural network represents one constituent, whose outputs are connected to another set of nodes which represents the parent in a syntactic tree. The syntactic tree can be learned (Socher et al., 2010), or given (Socher et al., 2011a,b). In the latter papers, a recursive autoencoder model is used, where the learning objective is to minimize reconstruction error of the training text at every point of the parse tree. Socher et al. (2012) further develop this approach by learning matrix representations for words to model their co-compositionality effects, in addition to the regular vector representations. Huang et al. (2012) introduce a neural network architecture that combines information from both the local context window as in standard models, as well as a global context from the document.

Blacoe and Lapata (2012) compare the simple vector space model of Mitchell and Lapata (2008) to the Distributional Memory approach and Socher et al. (2011a)'s model on several semantic tasks involving lexical semantics and paraphrase detection. Surprisingly, they find that the simple method performs about as well as the other two more sophisticated models, despite the simple model not requiring syntactic information or learning.

4.2 Evaluating Distributional Semantics for Inference

One obstacle to using these semantic models is the lack of guidance of which model to select or the strengths and weaknesses of each. This is primarily due to ad-hoc evaluation methods of models designed to showcase the specific strengths of a system, rather than to evaluate the potential of using the semantic model in an applied setting such as summarization. The first step to applying these models would be a better evaluation and comparison of these models to determine their potential for semantic inference.

In addition, the above work focused on the notion of compositionality as the litmus test of a truly semantic model. While compositionality may provide a convenient recipe for producing representations of propositionally typed phrases, it is not a necessary condition for a semantic representation. That distinction still belongs to the crucial ability to support inference.

As Richard Montague put it, *"The basic aim of semantics is to characterize the notion of a true sentence (under a given interpretation) and of entailment"* (1970). In other words, a model that is not capable of natural language inference *does not even deserve to be called a semantics*.

A desirable and arguably necessary for a compositional semantic representation to support inference *invariantly*, in the sense that the particular syntactic construction that guided the composition should not matter relative to the representations of syntactically different phrases with the same meanings. For example, one can assert that *John threw the ball* and *The ball was thrown by John* have the same meaning for the purposes of inference, even though they differ syntactically.

An analogy can be drawn to research in image processing, in which it is widely regarded as important for the representations of images to be invariant to rotation and scaling. What should be desired of a representation of sentence meaning that it should be invariant to diathesis, other regular syntactic alternations in the assignment of argument structure, and, ideally, even invariant to other meaning-preserving or near-preserving paraphrases.

Existing evaluations of distributional semantic models fall short of measuring this. One

evaluation approach consists of lexical-level word substitution tasks which primarily evaluate a system's ability to disambiguate word senses within a controlled syntactic environment (McCarthy and Navigli, 2009, for example). Another approach is to evaluate parsing accuracy (Socher et al., 2010, for example), which is really a formalism-specific approximation to argument structure analysis. These evaluations may certainly be relevant to specific components of, for example, machine translation or natural language generation systems, but they tell us little about a semantic model's ability to support inference.

Below, I propose a general framework for evaluating distributional semantic models that build sentence representations, and suggest two evaluation methods that test the notion of structurally invariant inference directly. Both rely on determining whether sentences express the same semantic relation between entities, a crucial step in solving a wide variety of inference tasks like recognizing textual entailment, information retrieval, question answering, and summarization.

The first evaluation is a relation classification task, where a semantic model is tested on its ability to recognize whether a pair of sentences both contain a particular semantic relation, such as *Company X acquires Company Y*. The second task is a question answering task, the goal of which is to locate the sentence in a document that contains the answer. Here, the semantic model must match the question, which expresses a proposition with a missing argument, to the answer-bearing sentence which contains the full proposition.

I apply these new evaluation protocols to several recent distributional models, extending several of them to build sentence representations. I find that the models outperform a simple lemma overlap model only slightly, but that combining these models with the lemma overlap model can improve performance. This result is likely due to weaknesses in current models' ability to deal with issues such as named entities, coreference, and negation, which are not emphasized by existing evaluation methods, but it does suggest that distributional models of semantics can play a more central role in systems that require deep, precise inference.

4.2.1 Existing Evaluations

Logic-based forms of compositional semantics have long striven for syntactic invariance in meaning representations, which is known as the doctrine of the canonical form. The traditional justification for canonical forms is that they allow easy access to a knowledge base to retrieve some desired information, which amounts to a form of inference. This work can be seen as an extension of this notion to distributional semantic models with a more general notion of representational similarity and inference.

There are many regular alternations that semantics models have tried to account for such as passive or dative alternations. There are also many lexical paraphrases which can take drastically different syntactic forms. Take the following example from Poon and Domingos (2009), in which the same semantic relation can be expressed by a transitive verb or an attributive prepositional phrase:

(4.9) *Utah borders Idaho.*

Utah is next to Idaho.

In distributional semantics, the original sentence similarity test proposed by Kintsch (2001) served as the inspiration for the evaluation performed by Mitchell and Lapata (2008) and most later work in the area. Intransitive verbs are given in the context of their syntactic subject, and candidate synonyms are ranked for their appropriateness. This method targets the fact that a synonym is appropriate for only some of the verb's senses, and the intended verb sense depends on the surrounding context. For example, *burn* and *beam* are both synonyms of *glow*, but given a particular subject, one of the synonyms (called the High similarity landmark) may be a more appropriate substitution than the other (the Low similarity landmark). So, if *the fire* is the subject, *glowed* is the High similarity landmark, and *beamed* the Low similarity landmark.

Fundamentally, this method was designed as a demonstration that compositionality in computing phrasal semantic representations does not interfere with the ability of a representation to synthesize non-compositional collocation effects that contribute to the disambiguation of

homographs. Here, word-sense disambiguation is implicitly viewed as a very restricted, highly lexicalized case of inference for selecting the appropriate disjunct in the representation of a word's meaning.

Kintsch (2001) was interested in sentence similarity, but he only conducted his evaluation on a few hand-selected examples. Mitchell and Lapata (2008) conducted theirs on a much larger scale, but chose to focus only on this single case of syntactic combination, intransitive verbs and their subjects, in order to "factor out inessential degrees of freedom" to compare their various alternative models more equitably. This was not necessary—using the same, sufficiently large, unbiased but syntactically heterogeneous sample of evaluation sentences would have served as an adequate control—and this decision furthermore prevents the evaluation from testing the desired invariance of the semantic representation.

Other lexical evaluations suffer from the same problem. One uses the WordSim-353 dataset (Finkelstein et al., 2002), which contains human word pair similarity judgements that semantic models should reproduce. However, the word pairs are given without context, and homography is unaddressed. Also, it is unclear how reliable the similarity scores are, as different annotators may interpret the integer scale of similarity scores differently. Recent work uses this dataset mostly for parameter tuning. Another is the lexical paraphrase task of McCarthy and Navigli (2009), in which words are given in the context of the surrounding sentence, and the task is to rank a given list of proposed substitutions for that word. The list of substitutions as well as the correct rankings are elicited from annotators. This task was originally conceived as an applied evaluation of WSD systems, not an evaluation of phrase representations.

Parsing accuracy has been used as a preliminary evaluation of semantic models that produce syntactic structure (Socher et al., 2010; Wu and Schuler, 2011). However, syntax does not always reflect semantic content, and the focus here is specifically on supporting syntactic invariance when doing semantic inference. Also, this type of evaluation is tied to a particular grammar formalism.

The existing evaluations that are most similar in spirit to what I propose are paraphrase

detection tasks that do not assume a restricted syntactic context. Washtell (2011) collected human judgements on the general meaning similarity of candidate phrase pairs. Unfortunately, no additional guidance on the definition of "most similar in meaning" was provided, and it appears likely that subjects conflated lexical, syntactic, and semantic relatedness. Dolan and Brockett (2005) define paraphrase detection as identifying sentences that are in a bidirectional entailment relation. While such sentences do support exactly the same inferences, NLP end applications are typically also interested in the inferences that can be made from similar sentences that are not paraphrases according to this strict definition. Thus, I adopt a less restricted notion of paraphrasis.

4.2.2 An Evaluation Framework

I now describe a simple, general framework for evaluating semantic models using the idea of argument structure invariance. The framework consists of the following components: a semantic model to be evaluated, pairs of sentences that are considered to have high similarity, and pairs of sentences that are considered to have low similarity.

In particular, the semantic model is a binary function, $s = \mathscr{M}(x, x')$, which returns a real-valued similarity score, s, given a pair of arbitrary linguistic units (that is, words, phrases, sentences, etc.), x and x'. Note that this formulation of the semantic model is agnostic to whether the models use compositionality to build a phrase representation from constituent representations, and even to the actual representation used. The model is tested by applying it to each element in the following two sets:

$$H = \{(h, h') | h \text{ and } h' \text{ are linguistic units with high similarity}\} \tag{4.10}$$

$$L = \{(l, l') | l \text{ and } l' \text{ are linguistic units with low similarity}\} \tag{4.11}$$

The resulting sets of similarity scores are:

$$\mathscr{S}^H = \left\{ \mathscr{M}(h,h') | (h,h') \in H \right\} \tag{4.12}$$

$$\mathscr{S}^L = \left\{ \mathscr{M}(l,l') | (l,l') \in L \right\} \tag{4.13}$$

The semantic model is evaluated according to its ability to separate $\mathscr{S}^H$ and $\mathscr{S}^L$. I will define specific measures of separation for the two experimental settings shortly. While the particular definitions of "high similarity" and "low similarity" depend on the task, at the crux of both these evaluations is that two sentences are similar if they express the same semantic relation between a given entity pair, and dissimilar otherwise. This threshold for similarity is closely tied to the argument structure of the sentence, and allows considerable flexibility in the other semantic content that may be contained in the sentence, unlike the bidirectional paraphrase detection task. Yet it ensures that a consistent and useful distinction for inference is being detected, unlike unconstrained similarity judgements.

Also, compared to word similarity assessments or paraphrase elicitation, determining whether a sentence expresses a semantic relation is a much easier task cognitively for human judges. This binary judgement does not involve interpreting a numerical scale or coming up with an open-ended set of alternative paraphrases. It is thus easier to get reliable annotated data.

Below, I present two tasks that instantiate this evaluation framework and choice of similarity threshold. They differ in that the first is targeted towards recognizing declarative sentences or phrases, while the second is targeted towards a question answering scenario, where one argument in the semantic relation is queried.

4.2.3 Task 1: Relation Classification

The first task is a relation classification task. Relation extraction and recognition are central to a variety of other tasks, such as information retrieval, ontology construction, recognizing textual entailment and question answering. This task involves distinguishing sentences that

express some target semantic relation between a given entity pair from those that do not.

To understand the difficulty of this task, several sentences expressing the proposition *Pfizer acquires Rinat Neuroscience* are shown in Examples 4.14 to 4.16. These sentences illustrate the amount of syntactic and lexical variation that the semantic model must recognize as expressing the same semantic relation. In particular, besides recognizing synonymy or near-synonymy at the lexical level, models must also account for subcategorization differences, extra arguments or adjuncts, and part-of-speech differences due to nominalization.

(4.14) *Pfizer buys Rinat Neuroscience to extend neuroscience research and in doing so acquires a product candidate for OA.* (lexical difference)

(4.15) *A month earlier, Pfizer paid an estimated several hundred million dollars for biotech firm Rinat Neuroscience.* (extra argument, subcategorization)

(4.16) *Pfizer to Expand Neuroscience Research With Acquisition of Biotech Company Rinat Neuroscience* (nominalization)

In terms of the framework, the high and the low similarity sentence pairs are constructed in the following manner. First, a target semantic relation, such as *Company X acquires Company Y* is chosen, and entities are chosen for each slot in the relation, such as *Company X=Pfizer* and *Company Y=Rinat Neuroscience*. Then, sentences containing these entities are extracted and divided into two subsets. In one of them, E, the entities are in the target semantic relation, while in the other, *NE*, they are not. Examples 4.14 to 4.16 show several sentences that would belong to the set E, whereas the following examples shows a sample sentence in *NE*:

(4.17) *He has also received consulting fees from Alpharma, Organon, Eli Lilly and Company, Pfizer, Wyeth Pharmaceuticals, Janssen, Ortho-McNeil, Rinat Neuroscience, Elan Pharmaceuticals, and Forest Laboratories.*

The evaluation sets H and L are then constructed as follows:

$$H = E \times E \setminus \{(e,e)|e \in E\} \tag{4.18}$$

$$L = E \times NE \tag{4.19}$$

In other words, the high similarity sentence pairs are all the pairs where both express the target semantic relation, except the pairs between a sentence and itself, while the low similarity pairs are all the pairs where exactly one of the two sentences expresses the target relation.

Since the goal is to measure the models' ability to separate $\mathscr{S}^H$ and $\mathscr{S}^L$ in an unsupervised setting, standard supervised classification accuracy is not applicable. Instead, I employ the area under a ROC curve (AUC), which does not depend on choosing an arbitrary classification threshold. A ROC curve is a plot of the true positive versus false positive rate of a binary classifier as the classification threshold is varied. The area under a ROC curve can thus be seen as the performance of linear classifiers over the scores produced by the semantic model. The AUC can also be interpreted as the probability that a randomly chosen positive instance will have a higher similarity score than a randomly chosen negative instance. A random classifier is expected to have an AUC of 0.5.

AUC is not calculated on the scores of $\mathscr{S}^H$ and $\mathscr{S}^L$ directly, because these scores are not independent and there are quadratically many of them. Instead, the similarity scores associated with each element of E and NE are first averaged so that there is now one similarity score per element. AUC is then calculated on these average similarity scores.

4.2.4 Task 2: Restricted QA

The second task is a restricted form of question answering in the biomedical domain. For example, a question-answer pair might be the following:

(4.20) Q: *What does il-2 activate?*

A: *PI3K*

Sentence: *Phosphatidyl inositol 3-kinase (PI3K) is activated by IL-2.*

In this task, the system is given a question q and a document $\mathscr{D}$ consisting of a list of sentences, in which one of the sentences contains the answer to the question. The goal of the distributional semantic model is then to identify this sentence. I define:

$$H = \{(q,d)|d \in \mathscr{D} \text{ and } d \text{ answers } q\} \tag{4.21}$$

$$L = \{(q,d)|d \in \mathscr{D} \text{ and } d \text{ does not answer } q\} \tag{4.22}$$

In other words, the sentences are divided into two subsets; those that contain the answer to q should be similar to q, while those that do not should be dissimilar. I also assume that only one sentence in each document contains the answer, so H contains only one sentence.

Unrestricted question answering is a difficult problem that forces a semantic representation to deal sensibly with a number of other semantic issues such as coreference and information aggregation which still seem to be out of reach for contemporary distributional models of meaning. Since this work focuses on argument structure semantics, I restrict the question-answer pairs to those that only require dealing with paraphrases of this type.

This is accomplished by semi-automatically restricting the question-answer pairs using the manually corrected output of an unsupervised clustering semantic parser (Poon and Domingos, 2009). The semantic parser clusters semantic sub-expressions derived from a dependency parse of the sentence, so that those sub-expressions that express the same semantic relations are clustered. The parser is used to answer questions, and the output of the parser is manually checked. I use only those cases that have thus been determined to be correct question-answer pairs. As a result of this restriction, this task is rather more like Task 1 in how it tests a model's ability to recognize lexical and syntactic paraphrases. This task also involves recognizing voicing alternations, which were automatically extracted by the semantic parser, as demonstrated by Example 4.20.

Since there is only one element in H and hence $\mathscr{S}^H$ for each question and document, I measure the separation between $\mathscr{S}^H$ and $\mathscr{S}^L$ using the rank of the score of the answer-bearing sentence among the scores of all the sentences in the document. I normalize the rank so that it is between 0 (ranked least similar) and 1 (ranked most similar). Where ties occur, the sentence is ranked as if it were in the median position among the tied sentences. If the question-answer pairs are zero-indexed by i, $answer(i)$ is the index of the sentence containing the answer for the ith pair, and $length(i)$ is the number of sentences in the document, then the mean normalized rank score of a system is:

$$\overline{NormRank} = \mathop{\mathbf{E}}_{i}\left[1 - \frac{answer(i)}{length(i) - 1}\right] \tag{4.23}$$

4.3 Experiments

I reimplemented the models described in Section 4.1.1, setting the parameter as described in previous work where possible, which were typically tuned for the lexical similarity task of Finkelstein et al. (2002). In training the term-context matrix, the 50,000 most frequent lemmata are modelled as target words. Context vectors are constructed using a symmetric window of 5 words, and their dimensions represent the 3000 most frequent lemmatized context words excluding stop words. The raw counts in the term-context matrix are converted to positive pointwise mutual information scores, which has been shown to improve word similarity correlation results (Turney and Pantel, 2010).

The models were trained on the Annotated Gigaword corpus (Napoles et al., 2012), which is a version of the 5th edition of Gigaword (~4B tokens) that has been automatically preprocessed (i.e., tokenized, lemmatized, POS-tagged, and parsed). All models use cosine to measure the similarity between representations, except for the baseline model.

The E&P and D&L models were originally designed for constructing word vector representations in context. I extended them to compositionally construct phrase representations using

component-wise vector addition and multiplication. Since the focus of this chapter is on evaluation methods for such models, I did not experiment with other compositionality operators. Note, however, that component-wise operators have been popular in recent literature, and have been applied across unrestricted syntactic contexts (Mitchell and Lapata, 2009).

I implemented the latent Dirichlet allocation version of the D&L method. The contextualization operator in the E&P model was the component-wise multiplication in the original work. However, I found that this method interacted poorly with the component-wise composition operators, especially multiplication, because it is often the case that most of the dimensions "zero out", resulting in zero vectors that do not represent any useful semantic information. Instead, I defined the contextualization operator to be component-wise addition after dividing by the L2-norm. Suppose vector x is to be contextualized by vector x'. Then,

$$x \odot x' = x + x'/||x'||. \tag{4.24}$$

The distributional models are compared against a **Lemma Overlap** baseline. This baseline simply represents a sentence as the counts of each lemma present in the sentence after removing stop words. Let a sentence x consist of lemma-tokens $m_1, \ldots, m_{|x|}$. The similarity between two sentences is then defined as

$$\mathscr{M}(x,x') = \#In(x,x') + \#In(x',x) \tag{4.25}$$

$$\text{where } \#In(x,x') = \sum_{i=1}^{|x|} \mathbf{1}_{x'}(m_i \in x') \tag{4.26}$$

and $\mathbf{1}_{x'}(m_i \in x')$ is an indicator function that returns 1 if $m_i \in x'$ or 0 otherwise. This definition accounts for multiple occurrences of a lemma.

In addition, I tested hybrid models that combine the lemma overlap baseline with a distributional semantic model by summing the similarity scores from the two. These models give an idea of how distributional semantics could be a complement to shallow word-based represen-

Entities {X, Y}	+	**N**
Relation: acquires		
{Pfizer, Rinat Neuroscience}	41	50
{Yahoo, Inktomi}	115	433
Relation: was born in		
{Luc Besson, Paris}	6	126
{Marie Antoinette, Vienna}	39	105

Table 4.1: Task 1 dataset characteristics. *N* is the total number of sentences. + is the number of sentences that express the relation.

tations.

4.3.1 Task 1

Data I tested the semantic models on the relation extraction dataset of Bunescu and Mooney (2007), which contains sentences with entity pairs that may be in some target semantic relation. The dataset is separated into subsets depending on the target binary relation (*Company X acquires Company Y* or *Person X was born in Place Y*) and the entity pair (e.g., *Yahoo* and *Inktomi*) (Table 4.1). The dataset was constructed semi-automatically using a web search for the two entities in the prescribed order with up to seven content words in between. Then the extracted sentences were manually labelled by Bunescu and Mooney to indicate whether they express the target relation. Because the order of the entities has been fixed, passive alternations do not appear in this dataset.

Unlike other similar datasets such as that of Roth and Yih (2002), this dataset has a large number of candidate sentences for each subset of entity pair and relation. For each semantic model to be tested, I conducted the AUC-based evaluation on each of the four subsets, then averaged the results to compute the final evaluation score. This procedure in effect controls for the target entity pair, and makes the task more difficult, because the semantic model cannot make use of distributional information about the entity pair itself for inference.

Model	AUC
Overlap	0.7592
M&L add	0.7448
M&L mult	0.6420^{-}
D&L add	**0.7762**
D&L mult	0.4847^{-}
E&P add	0.6863
E&P mult	0.6867
Hybrid Models	
Overlap + M&L add	0.7619
Overlap + M&L mult	0.7671
Overlap + D&L add	0.7712^{+}
Overlap + D&L mult	0.7654
Overlap + E&P add	0.7603
Overlap + E&P mult	0.7697^{+}

Table 4.2: Task 1 results in AUC scores, averaged over the four subsets. The expected random baseline performance is 0.5. The superscripts $^{-}$ and $^{+}$ indicate statistically significantly worse or better performance than the overlap baseline respectively, according to a randomized bootstrap test at $p < 0.05$.

Results The results for Task 1 are given in Table 4.2. The D&L addition model performs the best, though the lemma overlap model presents a strong baseline. The simple M&L addition model performs quite well, while the syntax-modulated E&P model performs poorly on this task. Combining lemma overlap with distributional semantics seems to be beneficial, and two of the hybrid models are significantly better than the lemma overlap baseline. The pure D&L addition model is not significantly better, indicating a higher variance in its performance according to the randomized bootstrap test (Berg-Kirkpatrick et al., 2012).

Overall, some of the datasets are easier for the models than others. For example, the Overlap + D&L add model achieves an AUC of 0.8632 on the *Antoinette* dataset, but 0.6430 on *Yahoo*. More entity pairs and relations would be needed to investigate the models' variance across datasets.

Model	Full	Subset
Overlap	0.8830	0.7843
M&L add	0.7249^{-}	0.6962^{-}
M&L mult	0.4853^{-}	0.4962^{-}
D&L add	0.7106^{-}	0.6609^{-}
D&L mult	0.5583^{-}	0.5848^{-}
E&P add	0.8466^{-}	0.7639
E&P mult	0.5895^{-}	0.5972^{-}
Hybrid Models		
Overlap + M&L add	0.8857	0.7893
Overlap + M&L mult	0.8860	0.7898
Overlap + D&L add	0.8781	0.7752
Overlap + D&L mult	0.8855	0.7889
Overlap + E&P add	0.8910^{+}	0.7991^{+}
Overlap + E&P mult	$\mathbf{0.9012}^{+}$	$\mathbf{0.8179}^{+}$

Table 4.3: Task 2 results, in normalized rank scores. **Subset** is the cases where lemma overlap does not achieve a perfect score. The expected random baseline performance is 0.5. Significance testing against the overlap baseline was done by Wilcoxon signed rank tests ($^{-}$ and $^{+}$ indicate statistically significantly worse and better performance respectively).

4.3.2 Task 2

Data I used the question-answer pairs extracted by the Poon and Domingos (2009) semantic parser from the GENIA biomedical corpus that have been manually checked to be correct (295 pairs). Because the models were trained on newspaper text, they required adaptation to this specialized domain. Thus, I trained the M&L, and E&P models on the GENIA corpus, backing off to word vectors from the GENIA corpus when a word vector could not be found in the Gigaword-trained model. I could not do this for the D&L model, since the global latent senses that are found by latent Dirichlet allocation training do not have any absolute meaning that holds across multiple runs. Instead, I updated the Gigaword-trained D&L model by feeding in the additional word contexts from GENIA as training data.

Results The results are presented in Table 4.3. Lemma overlap again presents a strong baseline, but the hybridized models are able to outperform simple lemma overlap. Unlike in Task 1,

the hybrid E&P model achieves the best result, likely due to the need to more precisely distinguish syntactic roles in this task. The D&L addition model, which achieved the best performance in Task 1, does not perform as well in this task.

Even compared to Task 1, the pure distributional models underperform, compared to the baseline. Distributional models have problems in dealing with named entities which are common in this corpus, such as the names of genes and proteins, so the information from the lemma overlap is important for good performance. Nevertheless, the distributional semantic models are able to complement the lemma information, indicating their potential as part of the core semantic representation used in complex NLP tasks.

4.4 Conclusions

This chapter has introduced an evaluation framework for distributional models of semantics which build phrase- and sentence-level representations, and instantiated two evaluation tasks which test for the crucial ability to recognize whether sentences express the same semantic relation. These results demonstrate that compositional distributional models of semantics already have some utility in the context of more empirically complex semantic tasks than WSD-like lexical substitution tasks, in which compositional invariance is a requisite property. Simply computing lemma overlap, however, is a very competitive baseline, due to issues in these protocols with named entities and domain adaptivity. The better performance of the mixture models shows that such weaknesses can be addressed by hybrid semantic models. Future work should investigate more refined versions of such hybridization, as well as extend this idea to other semantic phenomena like coreference, negation and modality.

It should be noted that no single model or composition operator performs best for all tasks and datasets. A more thorough investigation of the factors that can predict the performance and/or invariance of a given composition operator is warranted. However, these results do indicate that current distributional semantic models are ready to be integrated into systems that

solve complex NLP tasks such as automatic summarization, which I will begin to examine in the next chapter.

Chapter 5

Distributional Semantic Hidden Markov Models

Two main issues have arisen so far in this dissertation. The first is the insufficient use of domain knowledge in current automatic summarization systems, which was discussed in Chapter 3. The second is the evaluation of distributional semantics in order to demonstrate their potential to support inference and complex NLP tasks, as demonstrated in Chapter 4.

In this chapter, I begin to address both issues by showing that distributional semantics can be used to improve the learning of structured domain representations, which are then used as the basis of a summarization method. The approach that I take to learning about a domain is **content modelling**, which attempts to discover the typical topics and the way these topics are structured from unannotated texts belonging to the target domain.

Generative probabilistic models have been one popular approach to content modelling. An important advantage of this approach is that the structure of the model can be adapted to fit the assumptions about the structure of the domain and the nature of the end task. As this field has progressed, the formal structures that are assumed to represent a domain have increased in complexity and become more hierarchical. Earlier work assumed a flat set of topics (Barzilay and Lee, 2004), which are expressed as states of a latent random variable in the model. Later

work organizes topics into a hierarchy from general to specific (Haghighi and Vanderwende, 2009; Celikyilmaz and Hakkani-Tur, 2010). Recently, Cheung et al. (2013) formalized a domain as a set of **frames** consisting of prototypical sequences of **events**, **slots**, and **slot fillers** or **entities**, inspired by classical AI work such as Schank and Abelson's (1977) scripts. I adopt much of this terminology in this work. For example, in the **Criminal Investigations** domain, there may be events such as an investigation of the crime, an arrest, a trial, and a conviction or exoneration. These would be indicated by **event heads** such as *arrest*, *charge*, *plead*, and *convict*. Relevant slots would include VICTIM, SUSPECT, AUTHORITIES, PLEA, etc.

One problem faced by this line of work is that, by their nature, these models are typically trained on a small corpus from the target domain, on the order of hundreds of documents. The small size of the training corpus makes it difficult to estimate reliable statistics, especially for more powerful features such as higher-order *n*-gram features or syntactic features.

By contrast, recall from previous chapters that distributional semantic models are trained on large, domain-general corpora, and they provide a notion of word similarity by way of vector similarity measures such as cosine. Furthermore, contextualization and co-compositional operators such as those proposed by (Mitchell and Lapata, 2008) can modify the meaning of a word according to the specific context in which that word appears. These models have been found to improve performance in tasks like lexical substitution and word sense disambiguation (Thater et al., 2011), as discussed in Chapter 4.

In this chapter, I propose to inject contextualized distributional semantic vectors into generative probabilistic models in order to combine their complementary strengths for domain modelling. There are a number of potential advantages that distributional semantic models offer. First, they provide domain-general representations of word meaning that cannot be reliably estimated from the small target-domain corpora on which probabilistic models are trained. Second, the contextualization process allows the semantic vectors to implicitly encode disambiguated word sense and syntactic information without further adding to the complexity of the generative model.

The proposed model, the Distributional Semantic Hidden Markov Model (DSHMM), is a novel variant of hidden Markov models that incorporates contextualized distributional semantic vectors into a generative probabilistic model as observed emissions. I demonstrate the effectiveness of this model in two domain modelling tasks. The first is slot induction, in which the goal is to find coherent entity clusters on a guided summarization data set over five different domains. I show that DSHMM outperforms a baseline version of the method that does not use distributional semantic vectors, as well as a recent state-of-the-art template induction method. The second task is multi-document summarization, in which the model must determine which event and slot topics are appropriate to include in a summary. Here, DSHMM outperforms previous methods that do not rely on manually encoded knowledge about the domains, as well as a previous content modelling approach (Li et al., 2011). From a modelling perspective, these results show that probabilistic models for content modelling and template induction benefit from distributional semantics trained on a much larger corpus. From the perspective of distributional semantics, this work broadens the variety of problems to which distributional semantics can be applied, and proposes methods to perform inference in a probabilistic setting beyond geometric measures such as cosine similarity.

5.1 Related Work

Unsupervised information and relation extraction based on heuristic clustering procedures have been used in automatic summarization (Filatova and Hatzivassiloglou, 2004; Hachey, 2009). Probabilistic content models were proposed by Barzilay and Lee (2004), and related models have since become popular for summarization (Fung and Ngai, 2006; Haghighi and Vanderwende, 2009), and information ordering (Elsner et al., 2007; Louis and Nenkova, 2012). Other related generative models include topic models and structured versions thereof (Blei et al., 2003; Gruber et al., 2007; Wallach, 2008). In terms of domain learning in the form of template induction, heuristic methods involving multiple clustering steps have been proposed (Fi-

latova et al., 2006; Chambers and Jurafsky, 2011). Most recently, Cheung et al. (2013) propose PROFINDER, a probabilistic model for frame induction inspired by content models. This work is similar in that I assume much of the same structure within a domain and consequently in the model as well (Section 5.2), but whereas PROFINDER focuses on finding the "correct" number of frames, events, and slots with a nonparametric method, this work focuses on integrating global knowledge in the form of distributional semantics into a probabilistic model. I adopt one of their evaluation procedures and use it to compare DSHMM to PROFINDER in Section 5.4.

Li et al. (2011) propose a hierarchical topic model that clusters sentences for automatic summarization. Crucially, their **event-aspect** model is trained over the documents of an entire domain as in DSHMM, and extends the HIERSUM method of Haghighi and Vanderwende (2009) by including a level of topics that is shared across a domain, but does not use any distributional semantic information. I will compare directly against this method in the summarization experiments.

Combining distributional information and probabilistic models has actually been explored in previous work. Usually, an ad-hoc clustering step precedes training and is used to bias the initialization of the probabilistic model (Barzilay and Lee, 2004; Louis and Nenkova, 2012), or the clustering is interleaved with iterations of training (Fung et al., 2003). By contrast, the proposed method better modularizes the two, and provides a principled way to train the model. More importantly, previous ad-hoc clustering methods use only distributional information derived from the target domain itself, because initializing based on domain-general distributional information can be problematic if it biases training towards a local optimum that is inappropriate for the target domain.

5.2 Distributional Semantic Hidden Markov Models

The DSHMM model is a directed probabilistic graphical model with a structure that is depicted in Figure 5.1. As a graphical model, the nodes in its graphical representation represent random

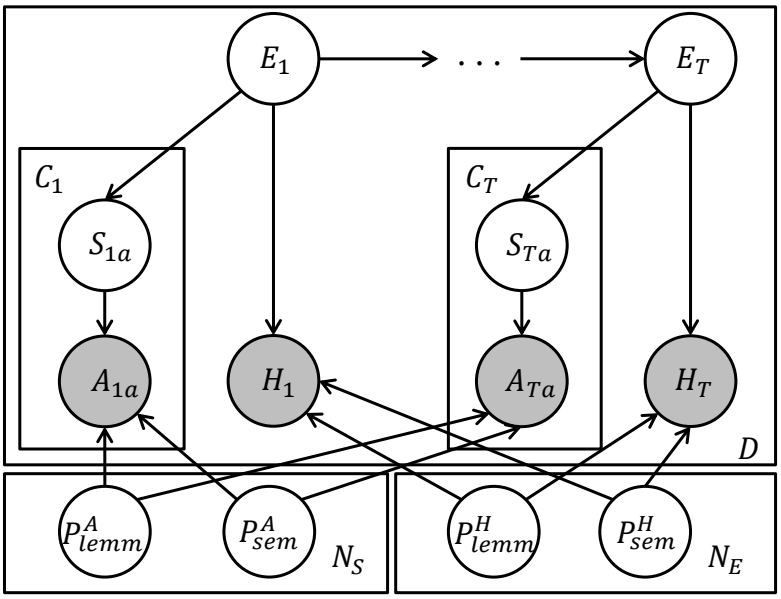


Figure 5.1: Basic graphical representation of DSHMM. Distributions that generate the latent variables and hyperparameters are omitted for clarity.

variables, and the directed edges encode the conditional dependency assumptions between the random variables. DSHMM can thought of as a variant of a standard HMM whose structure has been adapted to reflect the assumptions about the domain representation to be learned and the structure of the text to be modelled. The latent variables of DSHMM (the unshaded nodes in Figure 5.1) represent either parts of the domain structure being learned, such as the events and the slots, or the probability distributions that are being used to generate the observed text. The observed variables (the shaded nodes) represent the parts of the text that are generated or emitted by the model; i.e., the event heads and the entities. In contrast to a standard HMM, DSHMM contains two layers of latent variables that represent events and slots. In the generative story, these events and slots can be thought of as generating the parts of the text in their child nodes. For example, an event corresponding to a charge might generate event heads such as *charge* or *indict*, whereas a SUSPECT slot might generate entities such as *suspect*, *accused*, or the name of the suspect.

More precisely, given a document consisting of a sequence of T clauses headed by proposi-

tional heads $\vec{H}$ (verbs or event nouns), and argument noun phrases $\vec{A}$, DSHMM models the joint probability of observations $\vec{H}$, $\vec{A}$, and latent random variables $\vec{E}$ and $\vec{S}$ representing domain events and slots respectively; i.e., $P(\vec{H},\vec{A},\vec{E},\vec{S})$.

The basic structure of the model is also similar to that of PROFINDER. Each timestep in the model generates one clause in the document. More specifically, it generates the event heads and arguments which are crucial in identifying events and slots. I assume that event heads are verbs or event nouns, while arguments are the head words of their syntactically dependent noun phrases. I also assume that the sequence of clauses and the clause-internal syntactic structure are fixed, for example by applying a dependency parser. However, DSHMM differs from PROFINDER in not further distinguishing the latent events into a frame level and an event level within a frame, and in making use of distributional semantic vectors.

Within each clause, a hierarchy of latent and observed variables maps to corresponding elements in the clause (Table 5.1), as follows:

Event Variables At the top-level, a categorical latent variable E_t with N_E possible states represents the event that is described by clause t. Its value is conditioned on the previous time step's event variable, following the standard, first-order Markov assumption ($P^E(E_t|E_{t-1})$, or $P^E_{init}(E_1)$ for the first clause). The internal structure of the clause is generated by conditioning on the state of E_t, including the head of the clause, and the slots for each argument in the clause.

Slot Variables In addition to events, a clause t also contains a number of slots that represent the entities involved in the event. Let C_t be the number of slots. These slots are represented by categorical latent variables with N_S possible states, and are conditioned on the event variable in the clause, E_t (i.e., $P^S(S_{ta}|E_t)$, for the ath slot variable, where a ranges from 1 to C_t). The state of S_{ta} is then used to generate an argument A_{ta}.

Head and Argument Emissions The head of the clause H_t is conditionally dependent on E_t, and each argument A_{ta} is likewise conditioned on its slot variable S_{ta}. Unlike in most

Node	Component	Textual unit
E_t	Event	Clause
S_{ta}	Slot	Noun phrase
H_t	Event head	Verb/event noun
A_{ta}	Event argument	Noun phrase

Table 5.1: The correspondence between nodes in DSHMM, the domain components that they model, and the related elements in the clause.

applications of HMMs in text processing, in which the representation of a token is simply its word or lemma identity, tokens in DSHMM are also associated with a vector representation of their meaning *in context* according to a distributional semantic model. Thus, the emissions can be decomposed into pairs $H_t = (lemma(H_t), sem(H_t))$ and $A_{ta} = (lemma(A_{ta}), sem(A_{ta}))$, where *lemma* and *sem* are functions that return the lemma identity and the semantic vector respectively. The probability of the head of a clause is thus:

$$P^H(H_t|E_t) = P^H_{lemm}(lemma(H_t)|E_t) \times P^H_{sem}(sem(H_t)|E_t),$$

and the probability of a clausal argument is likewise:

$$P^A(A_{ta}|S_{ta}) = P^A_{lemm}(lemma(A_{ta})|S_{ta}) \times P^A_{sem}(sem(A_{ta})|S_{ta}).$$

All categorical distributions are smoothed using add-δ smoothing (i.e., uniform Dirichlet priors). Based on the independence assumptions described above, the joint probability distribution can be factored into:

$$\begin{aligned} P(\vec{H},\vec{A},\vec{E},\vec{S}) = {} & P^E_{init}(E_1) \\ & \times \prod_{t=2}^{T} P^E(E_t|E_{t-1}) \prod_{t=1}^{T} P^H(H_t|E_t) \\ & \times \prod_{t=1}^{T} \prod_{a=1}^{C_t} P^S(S_{ta}|E_t) P^A(A_{ta}|S_{ta}). \end{aligned} \tag{5.1}$$

5.2.1 Contextualization

I use the distributional semantic models and methods to contextualize word vector representations described in Section 4.1.1. Here, I recapitulate them with notation specific to their use in DSHMM.

SIMPLE Let event head h be the syntactic head of a number of arguments $a_1, a_2, ...a_m$. Given a distributional semantic model trained from a term-context matrix as described in Section 4.3, I call their respective vector representations $\vec{v}_h, \vec{v}_{a_1}, \vec{v}_{a_2}, ...\vec{v}_{a_m}$. The first distributional semantic model that I will test is to use these context-independent vectors in DSHMM, which I call the SIMPLE method.

M&L The Mitchell and Lapata (2008) method creates contextualized vectors $\vec{c}_h^{M\&L}, \vec{c}_{a_1}^{M\&L}, ...\vec{c}_{a_m}^{M\&L}$ as follows:

$$\vec{c}_h^{M\&L} = \vec{v}_h \odot (\bigodot_{i=1}^{m} \vec{v}_{a_m}) \tag{5.2}$$

$$\vec{c}_{a_i}^{M\&L} = \vec{v}_{a_i} \odot \vec{v}_h, \forall i = 1...m, \tag{5.3}$$

where $\odot$ represents a component-wise operator, addition or multiplication, and $\bigodot$ represents its repeated application. I tested component-wise addition (M&L+) and multiplication (M&L$\times$).

E&P Erk and Padó (2008) incorporate inverse selectional preferences into their contextualization function. Formally, let h take a as its argument in relation r. Then:

$$\vec{c}_h^{E\&P} = \vec{v}_h \times \prod_{i=1}^{m} \sum_{w \in \mathscr{L}} freq(w, r, a_i) \cdot \vec{v}_w, \tag{5.4}$$

$$\vec{c}_a^{E\&P} = \vec{v}_a \times \sum_{w \in \mathscr{L}} freq(h, r, w) \cdot \vec{v}_w, \tag{5.5}$$

where $freq(h,r,a)$ is the frequency of h occurring as the head of a in relation r in the training corpus, $\mathscr{L}$ is the lexicon, and $\times$ represents component-wise multiplication. Where a head occurs with more than one argument, I apply the contextualization procedure to the head vector to each argument in sequence.

D&L The method of Dinu and Lapata (2010) learns global latent sense "topics" using a topic modelling method such as latent Dirichlet allocation. This results in two distributions: the probability of a latent sense z_k given a target word w_1, $P_1(z_k|w_1)$, and the probability of a context word w_2 given a latent sense, $P_2(w_2|z_k)$. Then, given a head h with argument a, their contextualized representations up to normalization are:

$$\vec{c}_h^{D\&L} \propto \langle P_1(z_1|h)P_2(a|z_1), ..., P_1(z_K|h)P_2(a|z_K)\rangle \tag{5.6}$$

$$\vec{c}_a^{D\&L} \propto \langle P_1(z_1|a)P_2(h|z_1), ..., P_1(z_K|a)P_2(h|z_K)\rangle \tag{5.7}$$

where $z_{1...K}$ are the latent senses. I tested both the uncontextualized learned vectors (D&L) and the contextualized vectors (D&L-Cont). As above, if h occurs with more than one argument, its vector representation is contextualized by each of the arguments in sequence.

Dimensionality Reduction and Vector Emission After contextualization, I apply singular value decomposition (SVD) for dimensionality reduction to reduce the number of model parameters, keeping the k most significant singular values and vectors. In particular, I apply SVD to the m-by-n term-context matrix M produced by the SIMPLE method, resulting in the truncated matrices $M \approx U_k\Sigma_k V_k^T$, where U_k is a m-by-k matrix, Σ_k is k-by-k, and V_k is n-by-k. This takes place after contextualization, so the component-wise operators apply in the original semantic space. Afterwards, the contextualized vector in the original space, $\vec{c}$, can be transformed into a vector in the reduced space, $\vec{c}^R$, by $\vec{c}^R = \Sigma_k^{-1}V_k^T\vec{c}$.

Distributional semantic vectors are traditionally compared by measures which ignore vector magnitudes, such as cosine similarity, but a multivariate Gaussian is sensitive to magnitudes.

Thus, the final step is to normalize $\vec{c}^R$ into a unit vector by dividing it by its L2 norm, $||\vec{c}^R||$.

DSHMM models the emission of these contextualized vectors as multivariate Gaussian distributions, so the semantic vector emissions can be written as $P^H_{sem}, P^A_{sem} \sim \mathcal{N}(\mu, \Sigma)$, where $\mu \in \mathbb{R}^k$ is the mean and $\Sigma \in \mathbb{R}^{k \times k}$ is the covariance matrix. Regularization of the covariance matrix is performed using its conjugate prior, the Inverse-Wishart distribution, following the "neutral" setting of hyperparameters given by Ormoneit and Tresp (1995), so that the MAP estimate for the covariance matrix for (event or slot) state i becomes:

$$\Sigma_i = \frac{\sum_j r_{ij}(x_j - \mu_i)(x_j - \mu_i)^T + \beta I}{\sum_j r_{ij} + 1}, \tag{5.8}$$

where j indexes all the relevant semantic vectors x_j in the training set, r_{ij} is the posterior responsibility of state i for vector x_j, and β is the remaining hyperparameter that adjusts the amount of regularization. To further reduce model complexity, all off-diagonal entries of the resulting covariance matrix are set to zero.

5.2.2 Training and Inference

Inference in DSHMM is accomplished by the standard Inside-Outside and tree-Viterbi algorithms, except that the tree structure is fixed, so there is no need to sum over all possible subtrees. Model parameters are learned by the Expectation-Maximization (EM) algorithm. I tune the hyperparameters $(N_E, N_S, \delta, \beta, k)$ and the number of EM iterations by two-fold cross-validation[1].

5.2.3 Summary and Generative Process

In summary, the following steps are applied to train a DSHMM:

1. Train a distributional semantic model on a large, domain-general corpus.

[1]The topic cluster splits and the hyperparameter settings are available at `http://www.cs.toronto.edu/~jcheung/dshmm/dshmm.html`.

2. Preprocess and generate contextualized vectors of event heads and arguments in the small corpus in the target domain.
3. Train the DSHMM using the EM algorithm.

The formal generative process is as follows:

1. Draw categorical distributions P^E_{init}; P^E, P^S, P^H_{lemm} (one per event state); P^A_{lemm} (one per slot state) from Dirichlet priors.
2. Draw multivariate Gaussians P^H_{sem}, P^A_{sem} for each event and slot state, respectively.
3. Generate the document, clause by clause.

Generating a clause at position t consists of these steps:

1. Generate the event state $E_t \sim P^E$ (or P^E_{init}).
2. Generate the event head components $lemm(H_t) \sim P^H_{lemm}$, $sem(H_t) \sim P^H_{sem}$.
3. Generate a number of slot states $S_{ta} \sim P^S$.
4. For each slot, generate the argument components $lemm(A_{ta}) \sim P^A_{lemm}$, $sem(A_{ta}) \sim P^A_{sem}$.

5.3 Experiments

After training the distributional semantic models as described above, the DSHMM can then be trained on the target domain corpus. In the following experiments, the evaluations are performed on the TAC 2010 guided summarization data set (Owczarzak and Dang, 2010). Lemmatization and extraction of event heads and arguments are done by preprocessing with the Stanford CoreNLP tool suite (Toutanova et al., 2003; de Marneffe et al., 2006). The TAC 2010 data set contains 46 topic clusters of 20 articles each, grouped into five topic categories or domains. For example, one topic cluster in the ATTACK category is about the *Columbine Massacre*. Each topic cluster contains eight human-written model summaries. Half of the articles and model summaries in a topic cluster are used in the guided summarization task, and the rest are used in the update summarization task.

There are several advantages to choosing this data set. First, templates for the domains are

provided, and the model summaries are annotated with slots from the template, allowing for an intrinsic evaluation of slot induction (Section 5.4). Second, each topic cluster comes annotated with eight model summaries, allowing for an extrinsic evaluation of DSHMM by automatic summarization (Section 5.5).

5.4 Guided Summarization Slot Induction

In the first evaluation, the models are tested on their ability to produce coherent clusters of entities belonging to the same slot, following the experimental procedure of Cheung et al. (2013).

As part of the official TAC evaluation procedure, model summaries were manually segmented into *contributors*, and labelled with the slot in the TAC template that the contributor expresses. For example, a summary fragment such as *On 20 April 1999, a massacre occurred at Columbine High School* is segmented into the contributors: (*On 20 April 1999*, WHEN); (*a massacre occurred*, WHAT); and (*at Columbine High School*, WHERE).

In the slot induction evaluation, this annotation is used as follows. First, the maximal noun phrases are extracted from the contributors and clustered based on the TAC slot of the contributor. These clusters of noun phrases then become the gold standard clusters against which automatic systems are compared. Noun phrases are considered to be matched if the lemmata of their head words are the same and they are extracted from the same summary. This accounts for the fact that human annotators often only label the first occurrence of a word that belongs to a slot in a summary, and follows the standard evaluation procedure in previous information extraction tasks, such as MUC-4. Pronouns and demonstratives are ignored in this evaluation. This extraction process is noisy, because the meaning of some contributors depends on an entire verb phrase, but I keep this representation to allow a direct comparison to previous work.

The clusters produced by the unsupervised systems are not labelled, and must be matched

Method	P	R	F1
HMM w/o semantics	13.8	64.1	22.6^{-}
DSHMM w/ D&L	20.3	26.6	23.0^{-}
DSHMM w/ SIMPLE	20.9	27.5	23.7
DSHMM w/ E&P	20.7	27.9	23.8
PROFINDER	**23.7**	25.0	24.3
DSHMM w/ D&L-Cont	20.4	32.8	25.1
DSHMM w/ M&L+	19.7	**36.3**	25.6^{+}
DSHMM w/ M&L×	22.1	33.2	$\mathbf{26.5}^{+}$

Table 5.2: Slot induction results on the TAC guided summarization data set, ordered by increasing performance by F1. Superscripts ($^{-}$ or $^{+}$) indicate that the model is statistically significantly worse or better than PROFINDER in terms of F1 at $p < 0.05$.

to the gold standard clusters before evaluation can be performed. This matching is performed by mapping to each gold cluster the best system cluster according to F1. The same system cluster may be mapped multiple times, because several TAC slots can overlap. For example, in the **Natural Disasters** domain, an *earthquake* may fit both the WHAT slot as well as the CAUSE slot, because it generated a *tsunami*.

A DSHMM is trained for each of the five domains with different semantic models, tuning hyperparameters by two-fold cross-validation. The trained models are then used to extract noun phrase clusters from the model summaries according to the slot labels produced via the Viterbi algorithm.

Results I compared DSHMM to two baselines. The first baseline is PROFINDER, a state-of-the-art template inducer which Cheung et al. (2013) showed to outperform the previous heuristic clustering method of Chambers and Jurafsky (2011). The second baseline is the DSHMM model, without the semantic vector component, (HMM w/o semantics). To calculate statistical significance, I use the paired bootstrap method, which can accommodate complex evaluation metrics like F1 (Berg-Kirkpatrick et al., 2012).

Table 5.2 shows the performance of the models. Overall, PROFINDER significantly out-

performs the HMM baseline and the DSHMM model with D&L's uncontextualized vectors, but most of the DSHMM models outperform PROFINDER. DSHMM with contextualized semantic vectors achieves the highest F1s, and the ones based on M&L are significantly better than PROFINDER. All of the differences in precision and recall between PROFINDER and the other models are significant. The baseline HMM model has highly imbalanced precision and recall. This is likely because the model is unable to successfully produce coherent clusters, so the best-case mapping procedure during evaluation picked large clusters that have high recall. PROFINDER has slightly higher precision, which may be due to its non-parametric split-merge heuristic. I plan to investigate whether this learning method could improve DSHMM's performance further. Importantly, the contextualization of the vectors seems to be beneficial, at least with the M&L component-wise operators. In the next section, I show that the improvement from contextualization transfers to multi-document summarization results.

5.5 Multi-document Summarization: An Extrinsic Evaluation

The second experiment is an extrinsic evaluation in the setting of extractive, multi-document summarization. To use the trained DSHMM for extractive summarization, a decoding procedure is needed for selecting sentences in the source text to include in the summary. Inspired by the KLSUM and HIERSUM methods of Haghighi and Vanderwende (2009), I develop a criterion based on Kullback-Leibler (KL) divergence between distributions estimated from the source text, and those estimated from the summary. The assumption here is that these distributions should match in a good summary. Below, I present two methods to use this criterion: a basic unsupervised method (Section 5.5.1), and a supervised variant that makes use of in-domain summaries to learn the salient slots and events in the domain (Section 5.5.2).

5.5.1 A KL-based Criterion

There are four main component distributions of DSHMM that should be considered during extraction: (1) the distribution of events, (2) the distribution of slots, (3) the distribution of event heads, and (4) the distribution of arguments. I estimate (1) as the context-independent probability of being in a certain event state, which can be calculated using the Inside-Outside algorithm. Given a collection of documents D which make up the source text, the distribution of event topics $\hat{P}^E(E)$ is estimated as:

$$\hat{P}^E(E=e) = \frac{1}{Z}\sum_{d\in D}\sum_{t}\frac{In_t(e)Out_t(e)}{P(d)}, \tag{5.9}$$

where $In_t(e)$ and $Out_t(e)$ are the values of the inside and outside trellises at timestep t for some event state e, and Z is a normalization constant. The distribution for a set of sentences in a candidate summary, $\hat{Q}^E(E)$, is identical, except the summation is over the clauses in the candidate summary. Slot distributions $\hat{P}^S(S)$ and $\hat{Q}^S(S)$ (2) are defined analogously, where the summation occurs along all the slot variables.

For (3) and (4), I simply use the MLE estimates of the lemma emissions, where the estimates are made over the source text and the candidate summary instead of over the entire training set. All of the candidate summary distributions (i.e., the "$\hat{Q}$ distributions") are smoothed by a small amount, so that the KL-divergence is always finite. The KL criterion combines the above components linearly, weighting the lemma distributions by the probability of their respective event or slot state:

$$\begin{aligned} KLScore = {} & \mathscr{D}_{KL}(\hat{P}^E||\hat{Q}^E) + \mathscr{D}_{KL}(\hat{P}^S||\hat{Q}^S) \\ & + \sum_{e=1}^{N_E}\hat{P}^E(e)\mathscr{D}_{KL}(\hat{P}^H(H|e)||\hat{Q}^H(H|e)) \\ & + \sum_{s=1}^{N_S}\hat{P}^S(s)\mathscr{D}_{KL}(\hat{P}^A(A|s)||\hat{Q}^A(A|s)) \end{aligned} \tag{5.10}$$

Method	ROUGE-1		ROUGE-2		ROUGE-SU4	
	unsup.	*sup.*	*unsup.*	*sup.*	*unsup.*	*sup.*
Leading baseline	28.0	–	5.39	–	8.6	–
HMM w/o semantics	32.3	32.7	6.45	6.49	10.1	10.2
DSHMM w/ SIMPLE	32.1	32.7	5.81	6.50	9.8	10.2
DSHMM w/ M&L+	32.1	33.4	6.27	6.82	10.0	10.6
DSHMM w/ M&L×	32.4	**34.3**$^{+}$	6.35	7.11ˆ	10.2	**11.0**$^{+}$
DSHMM w/ E&P	**32.8**	33.8^{+}	6.38	**7.31**$^{+}$	**10.3**	10.8^{+}
DSHMM w/ D&L	31.7	33.5	5.79	6.92	9.8	10.7^{+}
DSHMM w/ D&L-Cont	31.5	33.4	5.51	7.05^{+}	9.4	10.6ˆ
HIERSUM	28.7	–	5.50	–	8.9	–
Event-aspect	32.6	–	**6.51**	–	10.1	–

Table 5.3: TAC 2010 summarization results by three settings of ROUGE. The superscript $^{+}$ indicates that the model is statistically significantly better than the HMM model without semantics at a 95% confidence interval, a caret ˆ indicates that the value is significant at a 90% confidence interval.

To produce a summary, sentences from the source text are greedily added such that *KLScore* is minimized at each step, until the desired summary length is reached, discarding sentences with fewer than five words.

5.5.2 Supervised Learning

The above unsupervised method results in summaries that closely mirror the source text in terms of the event and slot distributions, but this ignores the fact that not all such topics should be included in a summary. It also ignores genre-specific, stylistic considerations about characteristics of good summary sentences. For example, Woodsend and Lapata (2012) find several factors that indicate sentences should *not* be included in an extractive summary, such as the presence of personal pronouns. Thus, I implemented a second method, in which I modify the KL criterion above by estimating $\hat{P}^E$ and $\hat{P}^S$ from other model summaries that are drawn from the same domain (i.e. topic category), except for those summaries that are written for the specific topic cluster to be used for evaluation.

5.5.3 Method and Results

I used the best performing models from the slot induction task and the above unsupervised and supervised methods based on KL-divergence to produce 100-word summaries of the guided summarization source text clusters. I compared against several baselines. The first is the leading baseline, a well-known, non-trivial one for news articles. In this baseline, the leading sentences from the most recent document in the source text cluster are used as the summary, up to the word length limit. The next baselines are the HMM baseline without semantics and DSHMM with SIMPLE distributional semantics, which measure the effect of adding distributional semantics, and contextualization respectively. Finally, I also show the results of the event-aspect model of Li et al. (2011), and their implementation of HIERSUM (Haghighi and Vanderwende, 2009), including a sentence compression component introduced by Li et al. (2011) which was also used in the event-aspect model. The models were applied to the original summarization task only, because they have not been adapted to the update task. All results are in terms of the standard ROUGE suite of automatic evaluation measures (Lin, 2004).

Note that the evaluation conditions of TAC 2010 are different, and thus those results are not directly comparable. For instance, top performing systems in TAC 2010 make use of manually constructed lists of entities known to fit the slots in the provided templates and sample topic statements, which this method automatically learns.

Table 5.3 shows the summarization results for the three most widely-used settings of ROUGE. All of the models outperform the leading baseline by a large margin, demonstrating the effective of the KL-criterion. In terms of unsupervised performance, all of the models perform similarly. Because the unsupervised method mimics the distributions in the source text at all levels, the method may negate the benefit of learning and simply produce summaries that match the source text in the word distributions, thus being an approximation of KLSUM. The event-aspect model is not signficantly better, despite a complex summarization pipeline that includes a sentence compression step.

Looking at the supervised results, the semantic vector models show clear gains in ROUGE,

whereas the baseline method does not obtain much benefit from supervision. As in the previous evaluation, the models with contextualized semantic vectors provide the best performance. M&L× performs very well, as in slot induction, but E&P also performs well, unlike in the previous evaluation. This result reinforces the importance of the contextualization procedure for these distributional semantic models. The effect of contextualization is more mixed for the D&L model, however, as the uncontextualized D&L model achieves similar results as the contextualized one in the supervised setting. More analysis is needed to understand the cause of this result.

Analysis To better understand what is gained by supervision using in-domain summaries, I analyzed the best performing M&L× model's output summaries for one document cluster from each domain. For each event state, I calculated the ratio $\hat{P}^{E}_{summ}(e)/\hat{P}^{E}_{source}(e)$, for the probability of an event state e as estimated from the training summaries and the source text respectively. Likewise, I calculated $\hat{P}^{S}_{summ}(s)/\hat{P}^{S}_{source}(s)$ for the slot states. This ratio indicates the change in state's probability after supervision; the greater the ratio, the more preferred that state becomes after training. I selected the most preferred and dispreferred event and slot for each document cluster, and took the three most probable lemmata from the associated lemma distribution (Table 5.4). It seems that supervision is beneficial because it picks out important event heads and arguments in the domain, such as *charge*, *trial*, and *murder* in the **Trials** domain. It also helps the summarizer avoid semantically generic words (*be* or *have*), pronouns, quotatives, and common but irrelevant words (*home*, *city*, *restaurant* in **Trials**).

5.6 Discussion

I have shown that contextualized distributional semantic vectors can be successfully integrated into a generative probabilistic model for domain modelling, as demonstrated by improvements in slot induction and multi-document summarization. The effectiveness of the model stems from the use of a large domain-general corpus to train the distributional semantic vectors, and

Domain	Event Heads		Slot Arguments	
	+	−	+	−
ATTACKS	*say*[a], *cause*, *doctor*	*say*[a], *be*, *have*	*attack*, *hostage*, *troops*	*he*, *it*, *they*
TRIALS	*charge*, *trial*, *accuse*	*say*, *be*, *have*	*prison*, *murder*, *charge*	*home*, *city*, *restaurant*
RESOURCES	*reduce*, *increase*, *university*	*say*, *be*, *have*	*government*, *effort*, *program*	*he*, *they*, *it*
DISASTERS	*flood*, *strengthen*, *engulf*	*say*, *be*, *have*	*production*, *statoil*, *barrel*	*he*, *it*, *they*
HEALTH	*be*, *department*, *have*	*say*, *do*, *make*	*food*, *product*, *meat*	*she*, *people*, *way*

[a]The event head *say* happens to appear in both the most preferred and dispreferred events in the ATTACKS domain.

Table 5.4: Analysis of the most probable event heads and arguments in the most preferred (+) and dispreferred (−) events and slots after supervised training.

the implicit syntactic and word sense information provided by the contextualization process. The approach is modular, and allows principled training of the probabilistic model using standard techniques.

The structure of the DSHMM model is very flexible and can be extended in a number of ways. For example, it would be possible to differentiate multiple levels of event and slot topics as in hierarchical models such as Li et al. (2011). While I have focused on the overall clustering of entities and the distribution of event and slot topics in this work, I would also like to investigate discourse modelling and content structuring. Finally, this work shows that the application of distributional semantics to NLP tasks need not be confined to lexical disambiguation, further broadening the variety of applications of distributional semantic methods.

Chapter 6

Sentence Enhancement for Automatic Summarization

This chapter presents **sentence enhancement** as a novel technique for text-to-text generation in abstractive summarization. Compared to extraction or previous approaches to sentence fusion, sentence enhancement increases the range of possible summary sentences by allowing the combination of dependency subtrees from any sentence in the source text. I present experiments that indicate the approach yields summary sentences that are competitive with a sentence fusion baseline in terms of content quality, but better in terms of grammaticality, and that the benefit of sentence enhancement relies crucially on an event coreference resolution algorithm using distributional semantics.

6.1 Sentence Revision for Abstractive Summarization

Sentence fusion is the technique of merging several input sentences into one output sentence while retaining the important content (Barzilay and McKeown, 2005; Filippova and Strube, 2008; Thadani and McKeown, 2013). For example, the sections of the following input sentences in bold may be fused into one output sentence:

Input: *Bil Mar Foods Co., a meat processor owned by Sara Lee, announced a recall of certain* ***lots of hot dogs and packaged meat****.*

Input: ***The outbreak led to the recall on Tuesday of*** *15 million pounds of hot dogs and cold cuts* ***produced at the Bil Mar Foods plant.***

Output: *The outbreak led to the recall on Tuesday of lots of hot dogs and packaged meats produced at the Bil Mar Foods plant.*

As a text-to-text generation technique, sentence fusion is attractive because it provides an avenue for moving beyond sentence extraction in automatic summarization, while not requiring deep semantic analysis beyond, say, a dependency parser and lexical semantic resources.

The overall trajectory pursued in the field can be characterized as a move away from local contexts relying heavily on the original source text towards more global contexts involving reformulation of the text. Whereas sentence extraction and sentence compression (Knight and Marcu, 2000, for example) involve taking one sentence and perhaps removing parts of it, traditional sentence fusion involves reformulating a small number of relatively similar sentences in order to take the union or intersection of the information present therein.

In this chapter, I present **sentence enhancement** as the next step along this path. Sentence enhancement is a novel technique which extends sentence fusion by combining the subtrees of many sentences into the output sentence, rather than just a few. Doing so allows relevant information from sentences that are not similar to the original input sentences to be added during fusion. As the following example shows, the phrase *of food-borne illness* can be added to the previous output sentence, despite originating in a source text sentence that is quite different overall:

Source text: *This fact has been underscored in the last few months by two unexpected outbreaks* ***of food-borne illness****.*

Output: *The outbreak* ***of food-borne illness*** *led to the recall on Tuesday of lots of hot dogs and meats produced at the Bil Mar Foods plant.*

Elsner and Santhanam (2011) proposed the first method to fuse disparate sentences. The input to their supervised algorithm consists of small numbers of sentences with compatible information that have manually identified by editors of articles. By contrast, my algorithm is unsupervised, and tackles the problem of identifying compatible event mergers in the entire source text as part of sentence enhancement. My method also takes into account the issue of event coreference to ensure that the predicates that are merged are compatible. It outperforms a previous syntax-based sentence fusion baseline on measures of summary content quality and grammaticality.

A more general argument of this chapter is to view the apparent dichotomy between text-to-text generation and semantics-to-text generation as simply different starting points towards the same end goal of precise and wide-coverage natural language generation. The statistical generation techniques developed by the text-to-text generation community have demonstrated their utility and wide applicability. Yet the sentence enhancement algorithm incorporating event coreference and the results of the studies demonstrate the following point—as text-to-text generation techniques move beyond using local contexts towards more dramatic reformulations of the type that human writers perform, more semantic analysis will be needed in order to ensure that the reformulations preserve the inferences that can be drawn from the input text.

6.2 Related Work

A relatively large body of work exists in sentence compression (Knight and Marcu, 2000; McDonald, 2006; Galley and McKeown, 2007; Cohn and Lapata, 2008; Clarke and Lapata, 2008, inter alia). This line of work models deletions, insertions, and substitutions of words or phrases in a single sentence with applications to automatic summarization and text simplification.

Syntax-based sentence fusion algorithms that merge together a small number of input sentences have been proposed (Barzilay and McKeown, 2005; Filippova and Strube, 2008). The proposed sentence enhancement algorithm builds upon this work, but considers the entire

source text and is not limited to the initial input sentences. There have also been methods for sentence fusion that are primarily word- or n-gram-based (Filippova, 2010; Thadani and McKeown, 2013). While such approaches have shown success, more structured representations will ultimately be needed in order to account for the syntactic and semantic transformations that human summary writers perform.

Few previous papers focus on combining the content of diverse sentences into one output sentence. Wan et al. (2008) propose sentence augmentation by identifying "seed" words in a single original sentence, then adding information from auxiliary sentences based on word co-occurrence counts. Elsner and Santhanam (2011) investigate the idea of fusing disparate sentences with a supervised algorithm, as discussed above. By constraining the input to their algorithm to manually annotated sentence pairs, they avoid the need to perform content selection and deeper semantic analysis, in contrast to the algorithm proposed in this chapter.

6.3 A Sentence Enhancement Algorithm

The basic steps in the sentence enhancement algorithm are as follows:

1. Sentence graph creation
2. Sentence graph expansion
3. Tree generation
4. Linearization

At a high level, the proposed method for sentence enhancement algorithm is similar to and inspired by the syntactic sentence fusion approach of Filippova and Strube (2008) (henceforth, F&S) in that it takes as input the dependency parses of a small number of sentences and returns an output sentence which fuses parts of the input sentences. These initial input sentences, which I call **core sentences**, should have a high degree of similarity with each other, and should form the core of a new sentence to be generated. In order to accomplish this, the dependency trees of the core sentences are fused into an intermediate **sentence graph** (Step 1), a directed

acyclic graph from which the final sentence will be generated (Steps 3 and 4).

However, unlike F&S or other previous approaches to sentence fusion, the sentence enhancement algorithm may also avail itself of the dependency parses of all of the other sentences in the source text, which expands the range of possible sentences that may be produced. In particular, while the overall similarity of these sentences to the core sentences may be low, making these sentences inappropriate for traditional align-and-fuse algorithms, there may be parts of these sentences that contain information that could be usefully incorporated into the sentence graph (Step 2). One important issue during this step is that the expansion of the sentence graph must be modulated by an event coreference component to ensure that the merging of information from different points in the source text is compatible and does not result in incorrect or nonsensical inferences.

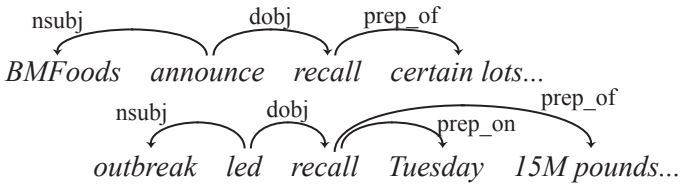
6.3.1 Sentence Graph Creation

The first step of the algorithm is to align the nodes of the dependency trees of the core input sentences in order to create the initial sentence graph. The input to this step is the collapsed dependency tree representations of the core sentences produced by the Stanford parser[1]. Thus, preposition nodes are collapsed into the label of the dependency edge between the functor of the prepositional phrase and the prepositional object. Chains of conjuncts are also split, and each argument is attached to the parent. In addition, auxiliary verbs, negation particles, and noun-phrase-internal elements[2] are collapsed into their parent nodes. Figure 6.1a shows the abbreviated dependency representations of the input sentences from above.

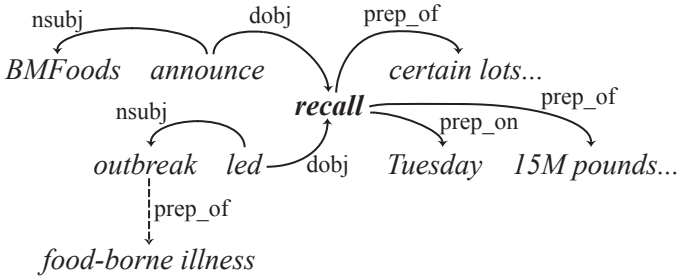
Then, a sentence graph is created by merging nodes that share a common lemma and part-of-speech tag. In addition, I allow synonyms that belong to the same WordNet synset to be merged. Merging is blocked if the word is a stop word, which includes function words as well as a number of very common verbs (e.g., *be*, *have*, *do*). Throughout the sentence graph

[1] As part of the CoreNLP suite: `http://nlp.stanford.edu/software/corenlp.shtml`

[2] As indicated by the dependency edge label *nn*.



(a) Abbreviated dependency trees.



(b) Sentence graph after merging the nodes with lemma *recall* (in bold), and expanding the node *outbreak* (dashed outgoing edge).

Figure 6.1: An example of the input dependency trees for sentence graph creation and expansion.

creation and expansion process, the algorithm disallows the addition of edges that would result in a cycle in the graph.

6.3.2 Sentence Graph Expansion

The initial sentence graph is expanded by merging in subtrees from dependency parses of non-core sentences drawn from the source text. First, expansion candidates are identified for each node in the sentence graph by finding all of the dependency edges in the source text from non-core sentences in which the governor of the edge shares the same lemma and POS tag as the node in the sentence graph.

Then, these candidate edges are pruned according to two heuristics. First, the candidate edges for each dependency relation type are ranked by a standard informativeness score (Section 6.3.3), and only the edge with the highest score is kept. Ties are broken such that the edge that has a subtree with a fewer number of nodes is ranked higher. The second is to per-

form event coreference in order to prune away those candidate edges which are unlikely to be describing the same event as the core sentences, as explained in the next section. Finally, any remaining candidate edges are fused into the sentence graph, and the subtree rooted at the dependent of the candidate edge is added to the sentence graph as well. See Figure 6.1b for an example of sentence graph creation and expansion.

Event Coreference

One problem of sentence fusion is that the different inputs of the fusion may not refer to the same event, resulting in an incorrect merging of information, as would be the case in the following example:

(6.1) *Officers pled not guilty but **risked** 25 years to life.*

(6.2) *Officers recklessly engaged in conduct which seriously **risked** the lives of others.*

Here, the first usage of *risk* refers to the potential sentence imposed if the officers are convicted in a trial, whereas the second refers to the potential harm caused by the officer.

In order to ensure that sentence enhancement and fusion do not lead to the merging of such incompatible events, I designed an unsupervised method to approximate event coreference resolution. This method is based on the intuition that different mentions of an event should contain many of the same participants. Thus, by measuring the similarity of the arguments in the syntactic contexts of the node in the sentence graph and in the candidate edge, I can obtain a measure of the likelihood that they refer to the same event. It would be interesting to integrate existing event coreference resolution systems into this step in the future, such as the unsupervised method of Bejan and Harabagiu (2010).

I measure the similarity of these syntactic contexts by aligning the arguments in the syntactic contexts and computing the similarity of the aligned arguments. These problems can be jointly solved as a maximum-weight bipartite graph matching problem (Figure 6.2).

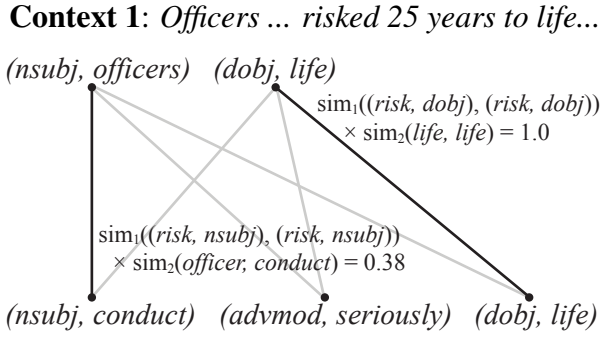


Figure 6.2: Event coreference resolution as a maximum-weight bipartite graph matching problem. All the nodes share the predicate *risk*. The edges in black are the optimal matching.

Formally, let a syntactic context be a list of dependency triples (h, r, a), consisting of a governor or head node h and a dependent argument a in the dependency relation r, where head node h is fixed across each element of the list. Then, each of the two input syntactic contexts forms one of the two disjoint sets in a complete weighted bipartite graph where each node corresponds to one dependency triple.

I define the edge weights according to the similarities of the edge's incident nodes; i.e., between two dependency triples (h_1, r_1, a_1) and (h_2, r_2, a_2). I also decompose the similarity into the similarities between the head and relation types ((h_1, r_1) and (h_2, r_2)), and the arguments (a_1 and a_2). The edge weight function can thus be written as:

$$\text{sim}((h_1, r_1, a_1), (h_2, r_2, a_2)) = \text{sim}_1((h_1, r_1), (h_2, r_2)) \times \text{sim}_2(a_1, a_2), \tag{6.3}$$

where sim_1 and sim_2 are binary functions that represent the similarities between governor-relation pairs and dependents, respectively. I train models of distributional semantics using a large background corpus—the Annotated Gigaword corpus (Napoles et al., 2012). For sim_1, I create a vector of counts of the arguments that are seen filling each (h, r) pair, and define the similarity between two such pairs to be the cosine similarity between their argument vectors. For sim_2, I create a basic vector-space representation of a word a according to words that are

found in the context of word *a* within a five-word context window, and likewise compute the cosine similarity between the word vectors. These methods of computing distributional similarity are well attested in lexical semantics for measuring the relatedness of words and syntactic structures (Turney and Pantel, 2010), and similar methods have been applied in text-to-text generation by Ganitkevitch et al. (2012), though the focus of that work is to use paraphrase information thus learned to improve sentence compression.

The resulting graph matching problem is solved using the NetworkX package for Python[3]. The final similarity score is an average of the similarity scores from Equation 6.3 that participate in the selected matching, weighted by the product of the IDF scores of the dependent nodes of each edge. This final score is used as a threshold that candidate contexts from the source text must meet in order to be eligible for being merged into the sentence graph. This threshold is tuned by cross-validation.

6.3.3 Tree Generation

The next major step of the algorithm is to extract an output dependency tree from the expanded sentence graph. I formulate this as an integer linear program, in which variables correspond to edges of the sentence graph, and a solution to the linear program determines the structure of an output dependency tree. I use ILOG CPLEX to solve all of the integer linear programs in the experiments.

A good dependency tree must at once express the salient or important information present in the input text as well as be grammatically correct and of a manageable length. These desiderata are encoded into the linear program as constraints or as part of the objective function.

[3] `http://networkx.github.io/`

Objective Function

Based on previous work in sentence compression and sentence fusion, I designed an objective function that measures the informativeness of the words that are selected as well as the probability of a syntactic relation being present for a given head word.

Let X be the set of variables in the program, and let each variable in X take the form $x_{h,r,a}$, a binary variable that represents whether an edge in the sentence graph from a head node with lemma h to an argument with lemma a in relation r is selected. Then, the original objective function defined by F&S is:

$$\max \sum_{x_{h,r,a} \in X} x_{h,r,a} \cdot P(r|h) \cdot I(a), \tag{6.4}$$

where $P(r|h)$ is the probability that head h projects the dependency relation r, and $I(a)$ is the informativeness score for word a as defined by Clarke and Lapata (2008). For the technical details of how this and other aspects of the linear program described below are implemented, see Section 6.6.

While this formulation works well for fusing a few core sentences, avoiding redundancy becomes a bigger issue in the expanded sentence graph, because many occurrences of an important word appear in the sentence graph. Thus, I modify the objective function to allow it to only score once for each lemma w in the lexicon Σ:

$$\max \sum_{w \in \Sigma} \max_{x_{h,r,a} \in X s.t. a=w} (x_{h,r,w} \cdot P(r|h) \cdot I(w)) \tag{6.5}$$

This objective function can be rewritten as a standard linear program by the addition of auxiliary variables and constraints.

Well-formedness Constraints

The first set of constraints are taken directly from F&S, and simply ensure that the set of selected edges is a well-formed tree; i.e., each selected node except the root has exactly one governor, and the set of selected nodes is connected. Another constraint specifies the number of content nodes in the tree, which I set at 11 to correspond to the average number of content nodes in human-written summary sentences in the data set.

Syntactic Constraints

F&S propose a syntactic constraint to ensure that a subordinating conjunction only appears in the output sentence if the associated subordinate clause remains a subordinate clause in the output. I propose two further syntactic constraints. The first ensures that a nominal or adjectival predicate must be selected with a copular construction at the top level of a non-finite clause. The second ensures that transitive verbs retain both of their complements in the output; that is, if there exists at least one subject and one direct object relation from a governor node, then if one is selected, so must the other[4].

Semantic Constraints

Semantic constraints ensure that only noun phrases of sufficiently high similarity which are not in a hypernym-hyponym or holonym-meronym relation with each other may be joined by coordination.

6.3.4 Linearization

The final step is to convert the dependency tree from the previous step into the final linear sequence of words, which is known as **linearization** or **surface realization**. While I could have

[4]I did not experiment with changing the grammatical voice in the output tree, such as introducing a passive construction if only a direct object is selected, but this is one possible extension of the algorithm.

used an off-the-shelf surface realization system, I chose to implement my own method, because much of the ordering information can be inferred from the original source text sentences.

My linearization algorithm proceeds top-down from the root of the dependency tree to the leaves. At each node of the tree, linearization consists of realizing the previously collapsed elements such as prepositions, determiners and noun compound elements, then ordering the dependent nodes with respect to the root node and each other. Restoring the collapsed elements is accomplished by simple heuristics. For example, the preposition and the determiner always precede the realization of the noun phrase itself.

The dependent nodes are ordered by a sorting algorithm, where the order between two syntactic relations and argument nodes (r_1, a_1) and (r_2, a_2) is determined as follows. First, if a_1 and a_2 originated from the same source text sentence, then they are ordered according to their order of appearance in the source text. Otherwise, I consider the probability $P(r_1 \text{ precedes } r_2)$, and order a_1 before a_2 iff $P(r_1 \text{ precedes } r_2) > 0.5$. This distribution, $P(r_1 \text{ precedes } r_2)$, is estimated by counting and normalizing the order of the relation types in the source text corpus. For the purposes of ordering, the governor node is treated as if it were a dependent node with a special syntactic relation label *self*. This algorithm always produces an output ordering with a projective dependency tree, which is a reasonable assumption for English.

6.4 Experiments

6.4.1 Method

Recent approaches to sentence fusion have often been evaluated as an isolated component separate from their use in a summary of some source text. For example, F&S evaluate the output sentences by asking human judges to rate the sentences' informativeness and grammaticality according to a 1–5 Likert scale rating. Thadani and McKeown (2013) combine manual grammaticality ratings with an automatic evaluation which compares the system output against gold-standard sentences drawn from summarization data sets. However, this evaluation setting

still does not reflect the utility of sentence fusion in summarization, because the input sentences come from human-written summaries rather than the original source text.

I adopt a more realistic setting of using sentence fusion in automatic summarization by drawing the input or core sentences automatically from the source text, then evaluating the output of the fusion and enhancement algorithms directly as one-sentence summaries according to standard summarization evaluation measures of content quality.

Data preparation The experiments are conducted on the TAC 2010 and TAC 2011 Guided Summarization corpora (Owczarzak and Dang, 2010), on the initial summarization task. To generate the core sentence clusters for the fusion and enhancement algorithms, I first identify clusters of similar sentences, then rank the clusters according to their salience. The top cluster in each document cluster is selected to be the input to the sentence fusion algorithms.

Sentence alignment is performed by complete-link agglomerative clustering, which requires a measure of similarity between sentences. I define the similarity between two sentences to be the IDF-weighted cosine similarity between the lemmas of the sentences. The clusters are scored according to the signature term method of Lin and Hovy (2000), which assigns an importance score to each term according to how much more often it appears in the source text compared to some irrelevant background text using a log-likelihood ratio. Specifically, the score of a cluster is equal to the sum of the importance scores of the set of lemmas in the cluster.

Evaluation measures I evaluate summary content quality using the word-overlap measures ROUGE-1 and ROUGE-2, as is standard in the summarization community. I also measure the quality of sentences at a syntactic or shallow semantic level that operates at the level of dependency triples by a measure that I call **Pyramid BE**. Specifically, I extract all of the dependency triples of the form $t = (h, r, a)$ from the sentence under evaluation and the gold-standard summaries, where h and a are the lemmas of the head and the argument, and r is the syntactic relation, normalized for grammatical voice.

Then, I perform a matching between the set of triples T^{eval} in the sentence under evaluation and in a reference summary T^{ref} following the Transformed BE method of Tratz and Hovy (2008). Let x_{ij} be a binary variable corresponding to the ith triple in T^{eval} and the jth triple in T^{ref}. Then, the matching problem can be written as:

$$\begin{aligned} &\max \sum_{i,j} I(t_i = t_j) W(t_j) x_{ij} \\ &\text{s. t.} \forall j, \sum_i x_{ij} \leq 1 \\ &\forall i, \sum_j x_{ij} \leq 1, \end{aligned} \tag{6.6}$$

where $I(t_i = t_j)$ is an indicator function that returns one if and only if t_i and t_j match, and $W(t_i)$ is a weighting function for the dependency triple which is equal to the number of reference summaries in which t_j appears (the **total** weighting scheme).

This matching is performed between the sentence and every gold-standard summary, and the maximum of these scores is taken. This score is then divided by the maximum score that is achievable using the number of triples present in the input sentence, as inspired by the Pyramid method. This denominator is more appropriate than the original method used in Transformed BE, which is designed for the case where the evaluated summary and the reference summaries are of comparable length.

For grammaticality, I parse the output sentences using the Stanford parser, and use the log likelihood of the most likely parse of the sentence as a coarse estimate of grammaticality. Parse log likelihoods have been shown to be useful in determining grammaticality (Wagner et al., 2009), and many of the problems associated with using it do not apply in the evaluation, because the sentences have a fixed number of content nodes, and contain similar content. While I could have conducted a user study to elicit Likert-scale grammaticality judgements, such results are difficult to interpret and the scores depend heavily on the set of judges and the precise evaluation setting (Napoles et al., 2011).

Method	Pyramid BE	ROUGE-1	ROUGE-2	Log Likelihood
Fusion (F&S)	10.61	**10.07**	**2.15**	-159.31
Enhancement	8.82	9.41	1.82	-157.46
+Event coref	**11.00**	9.76	1.93	**-156.20**

Table 6.1: Results of the sentence enhancement and fusion experiments on TAC 2010 and TAC 2011.

6.4.2 Results and Discussion

The results are presented in Table 6.1. As can be seen, sentence enhancement with coreference outperforms the sentence fusion algorithm of F&S in terms of the Pyramid BE measure and the baseline enhancement algorithm, though only the latter difference is statistically significant ($p < 0.022$[5]). In terms of the ROUGE word overlap measures, fusion achieves a better performance, but it only outperforms the enhancement baseline significantly (ROUGE-1: $p < 0.021$, ROUGE-2: $p < 0.012$). Note that the ROUGE scores are low because they involve comparing a one-sentence summary against a paragraph-long gold standard.

The average log likelihood result suggests that sentence enhancement with event coreference produces more grammatical sentences than traditional fusion, and this difference is statistically significant ($p < 0.044$).

These results are positive in that they show that sentence enhancement with event coreference is competitive with a strong previous sentence fusion method in terms of content, despite having to combine information from more diverse sentences, and that this does not come at the expense of grammaticality. In fact, it seems that having a greater possible range of output sentences may even improve the grammaticality of the output sentences.

There is still much room for improvement in the grammaticality of sentences in these models, however, and this will require modelling contexts larger than individual predicates and their arguments. To see the importance of considering more than individual predicate-argument

[5]All statistical significance results in this section are for Wilcoxon signed-rank tests.

pairs, consider the following output of the sentence enhancement with event coreference system:

(6.7) *The government has launched an investigation into Soeharto's wealth by the Attorney General's office on the wealth of former government officials.*

This sentence suffers from coherence problems because two of the expected slots in the domain are duplicated. The first is the subject of the investigation, which is expressed by two prepositional objects of *investigation* with the prepositions *into* and *on*. The second, more subtle incoherence is in the body that is responsible for the investigation, which is expressed both by the subject of *launch* (***The government*** *has launched an investigation*), and the *by*-prepositional object of *investigation* (an investigation ... ***by the Attorney General's office***). Clearly, a model that makes fewer independence assumptions about the relation between different edges in the sentence graph is needed.

6.5 Discussion

In this chapter, I introduced sentence enhancement as a method to incorporate information from multiple points in the source text into one output sentence in a fashion that is more flexible than previous sentence fusion algorithms. The results of my experiments show that the sentence enhancement method is competitive with a previous syntax-based sentence fusion approach in content quality and generates more grammatical summary sentences as well.

As suggested by the studies in Chapter 3, human summary writers do not restrict themselves to the source text, performing text reformulations that seem to be captured by considering in-domain articles external to the source text. More sophisticated semantic techniques will be needed in order to exploit such in-domain articles for text-to-text generation in summarization.

6.6 Appendix: ILP Encoding

Below, I describe how the novel objective function and syntactic constraints in the integer linear programming formulation of sentence generation are implemented.

6.6.1 Informativeness Score

The word informativeness function $I(a)$ depends on the frequency of the word in question, as well as the syntactic depth at which the word a is found in the source text, in terms of the level of embedding by the number of clause boundaries crossed from the root of the tree:

$$I(a) = \frac{depth(a)}{max\_depth} freq(a) \times \log \frac{F_{ALL}}{F_a}, \tag{6.8}$$

where $depth(a)$ is the syntactic depth of the argument node a in the parse tree of the sentence by the number of clause boundaries crossed on the path to the root of the tree, *max_depth* is the maximum depth of the sentence, $freq(a)$ is the frequency of word a in the document cluster, F_{ALL} is the total count of content words in the entire corpus, and F_a is the total count of word a in the corpus. The first factor captures the intuition that words that are more deeply embedded tend to be more important, at least in news text, where there are many uninformative reporting verbs near the top of a parse tree such as *said* or *announced*. The other factors are an instantiation of tf-idf as a method of determining term importance.

6.6.2 Objective Function

My modified objective function better avoids redundancy when deciding on the most informative tree to extract from the expanded sentence graph:

$$\max \sum_{w \in \Sigma} \max_{x_{h,r,a} \in X s.t. a=w} (x_{h,r,w} \cdot P(r|h) \cdot I(w)) \tag{6.9}$$

Since this is an integer linear program, the inner max must be factored out of the objective

function by the introduction of auxiliary variables and constraints. First, I introduce an auxiliary variable $y_{h,r,a}$ for each original variable $x_{h,r,a}$. Call the set of these auxiliary variables Y. I rewrite the objective function in terms of these auxiliary variables, removing the inner max function:

$$\max \sum_{y_{h,r,a} \in Y} y_{h,r,a} \cdot P(r|h) \cdot I(a). \tag{6.10}$$

I then add constraints in order to relate the auxiliary variables to their corresponding original variables, and to ensure that each lemma is only scored once. For the former, I constrain the auxiliary variables to be at most the value of the original:

$$y_{h,r,a} \leq x_{h,r,a}. \tag{6.11}$$

Then, I add a constraint for each lemma w in the lexicon Σ, such that at most one auxiliary variable may be "on" for each lemma:

$$\forall w \in \Sigma, \sum_{y_{h,r,a} \in Y s.t. a=w} y_{h,r,w} \leq 1. \tag{6.12}$$

The modified objective is equivalent to the original if the program is solved optimally, as the auxiliary variables will be set such that only the highest scoring $y_{h,r,a}$ variable for each lemma a contributes a positive value to the objective function.

6.6.3 Syntactic Constraints

Nominal and adjectival predicate In Stanford's collapsed dependency representation, nominal and adjectival predicates are indicated by a *nsubj* relation from the predicate head to the argument, and a *cop* relation to the copular, usually some form of the verb *to be*. I add a constraint to ensure these pairs are selected together, and furthermore that the construction is found at the top level of a finite clause.

Transitive verbs In order to ensure transitive verbs take both of their expected arguments, I need to implement the constraint for each relevant node that the number of dependents with the relation *nsubj* is greater than 0 if and only if the number of *dobj* children is greater than 0.

For a particular transitive verb node n in the expanded sentence graph, let the sets of variables in X that represent the *nsubj* children be denoted as $X_{n,nsubj}$. Then, I introduce a variable $h_{n,nsubj}$ that has value 1 if and only if at least one variable in $X_{n,nsubj}$ is 1:

$$\forall x \in X_{n,nsubj}, h_{n,nsubj} \geq x \tag{6.13}$$

$$h_{n,nsubj} \leq \sum_{x \in X_{n,nsubj}} x. \tag{6.14}$$

I likewise introduce constraints for $X_{n,dobj}$ and $h_{n,dobj}$. Then, I simply enforce that:

$$h_{n,nsubj} = h_{n,dobj}. \tag{6.15}$$

6.6.4 Semantic Constraints

I followed F&S in disallowing noun phrases that are in a hyponym/hypernym or holonym/meronym relation from being coordinated, as indicated by WordNet. I also disallowed noun phrases whose heads are dissimilar, according to the distributional semantic model described in Section 6.3.2. Here, "dissimilar" means the cosine similarity falls below the observed average of conjunct similarity, which was 0.3317. Rather than embed these constraints into the ILP as F&S, I precomputed the results, and simply added a constraint to the ILP to disallow conjunction between each pair of nodes that may not be conjoined.

Chapter 7

Conclusion

This dissertation has examined a number of issues in distributional semantics and automatic summarization, as well as the relationship between them. Below, I recapitulate the contributions of this dissertation and describe possible future research directions from each of these perspectives.

7.1 Summary of Contributions

7.1.1 Distributional Semantics

In terms of distributional semantics, one contribution of this dissertation is a novel evaluation framework which is based on first principles about the role of semantic representations in NLP systems. The key point is that semantic representations should be evaluated by their ability to support inference, rather than theory-internal considerations such as whether the representation is constructed compositionally or according to syntactic structure. This functional view of semantics contrasts with previous evaluations based on correlations with human judgements of word pair similarity, whose applicability to downstream applications is unclear.

A related contribution is to show empirically that current distributional semantic methods can be useful for complex NLP tasks. This is first demonstrated in two instantiations of the

above evaluation framework, involving relation classification and question answering. In these experiments, methods that are informed by a distributional semantic model consistently outperform those that are not.

Then, additional evidence is provided by the novel Distributional Semantic Hidden Markov Model (DSHMM), which incorporates distributional semantic word vector representations as emissions into a generative probabilistic content model. DSHMM with contextualized distributional semantic vectors outperforms other unsupervised methods in two domain modelling experiments on slot induction and extractive multi-document summarization. Furthermore, the contextualization procedure that is a part of many current distributional semantic models is important to the overall system's performance.

There are several apparent reasons for the success of this approach. First, distributional semantic models are typically trained on a corpus that is much larger than the amount of data available in a target domain, thus injecting some notion of domain-general knowledge into the inference process. Second, the contextualization process allows word-sense and syntactic information to be implicitly incorporated, but does not further add to the model complexity.

I have also demonstrated how distributional semantics can be applied specifically to automatic summarization at several points throughout this dissertation. It was used in the analysis of whether paraphrasing can account for the lower signature caseframe density in Section 3.3.2, in the learning of DSHMM for extractive multi-document summarization in Chapter 5, and in the abstractive sentence enhancement algorithm proposed in Chapter 6.

7.1.2 Abstractive Summarization

In terms of the other main topic of this dissertation, abstractive summarization across multiple domains, one contribution has been to identify the limiting plateau of the centrality-based extractive paradigm. A study of current summarization systems shows that compared to human summary writers, they are overfitted to the central or representative parts of the source text. At the same time, current automatic systems are still very far away from the type of synthetis

or abstraction that is present in the model summaries. Advances in sentence compression and sentence fusion only partially address the issues of extraction, because a significant portion of model summaries cannot be found in the source text at all.

Instead of focusing on better optimizing centrality measures, I have argued for more use of domain knowledge in automatic summarization. I have supported this argument by further studies that elucidates some of the linguistic factors involved in using in-domain articles outside of the source text. In particular, I find that human summary writers find predicates that are more specific to the semantic content that they wish to convey, and identify several features that could be useful in a future automatic system for making use of in-domain text.

One step on the path towards using non-source-text material in an automatic summary is to be able to combine information from multiple points within the source text with diverse contexts, unlike traditional sentence fusion which requires highly similar contexts as inputs. The sentence enhancement algorithm that I proposed demonstrates the feasibility of this step in a text-to-text generation setting, resulting in more informative summary sentences. Interestingly, the success of the algorithm relies on an event coreference resolution algorithm based on distributional semantics.

7.2 Limitations and Future Work

7.2.1 Distributional Semantics in Probabilistic Models

One of the central arguments of this dissertation is that models of distributional semantics have not been properly evaluated for their ability to support semantic inference for complex NLP applications. This dissertation has addressed these issues with novel evaluation methods and techniques to embed distributional semantics into automatic summarization, which lays the foundation for the development for more sophisticated distributional semantic models than the established ones used in this work, such as the simple component-wise contextualization methods.

The incorporation of distributional semantics into DSHMM is possible because of the flexibility of generative probabilistic models. This modularity and exhibited by DSHMM lends itself to many other possible extensions. Some examples include further investigation into variations on the structure of the graphical model, a non-parametric method to learn the number of domain components, more careful modelling of the discourse transitions between clauses beyond a simple first-order Markov independence assumption, etc. It would also be worth investigating whether an approach similar to DSHMM could be beneficial for other NLP tasks that rely on probabilistic modelling, such as parsing, discourse modelling, or machine translation.

7.2.2 Abstractive Summarization

The results of the studies in Chapter 3 lays the groundwork for a future system which makes use of in-domain articles outside of the source text. A major challenge will be to incorporate more semantic knowledge in order to select appropriate source-text-external components and to preserve the inferences that can be drawn from the source text.

A wide-coverage and semantically precise abstractive summarization system opens up avenues of future research; many of the issues that have been investigated in the past several years for extractive summarization can now be revisited in the context of an abstractive system. For example, whereas there is relatively little that can be done in terms of creating a coherent extractive summary, in an abstractive setting, many more options are possible, such as by using cohesive devices like pronouns and discourse cues.

Another important question is how to best evaluate abstractive summarization systems. Current automatic measures such as ROUGE were validated primarily on extractive automatic systems' correlation with human responsiveness judgements. It is an open question whether these correlations extend to comparisons of abstractive systems, or between extractive and abstractive systems. Furthermore, most existing evaluations do not consider the ability of summaries to synthesize and aggregate information in the source text. A more targeted evaluation is needed to focus on the specific aspects of summarization systems that require abstraction.

Bibliography

Proceedings of the Fourth Message Understanding Conference (MUC-4), 1992. Morgan Kaufmann.

DUC 2007 guidelines, 2007. URL `http://duc.nist.gov/duc2007/tasks.html`.

Marco Baroni and Alessandro Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721, 2010.

Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193. Association for Computational Linguistics, 2010.

Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. Frege in space: A program for compositional distributional semantics. *Linguistic Issues in Language Technology*, 9:5–110, 2014.

Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17, 1997.

Regina Barzilay and Lillian Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 113–120, 2004.

Regina Barzilay and Kathleen R. McKeown. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328, 2005.

David Bean and Ellen Riloff. Unsupervised learning of contextual role knowledge for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 297–304, 2004.

Cosmin A. Bejan and Sanda Harabagiu. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422. Association for Computational Linguistics, 2010.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In Dawn Holmes and Lakhmi Jain, editors, *Innovations in Machine Learning*, volume 194 of *Studies in Fuzziness and Soft Computing*, pages 137–186. Springer Berlin / Heidelberg, 2006.

Taylor Berg-Kirkpatrick, David Burkett, and Dan Klein. An empirical investigation of statistical significance in NLP. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 995–1005, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

Patrick Blackburn and Johan Bos. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. CSLI, 2005.

William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556. Association for Computational Linguistics, 2012.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

Razvan C. Bunescu and Raymond J. Mooney. Learning to extract relations from the web using

minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, volume 45, pages 576–583, 2007.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336. ACM, 1998.

Giuseppe Carenini, Raymond Ng, and Adam Pauls. Multi-document summarization of evaluative text. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 305–312, 2006.

Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. Summarizing emails with conversational cohesion and subjectivity. In *Proceedings of ACL-08: HLT*, pages 353–361, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Asli Celikyilmaz and Dilek Hakkani-Tur. A hybrid hierarchical model for multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 815–824, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 976–986, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Jackie Chi Kit Cheung and Gerald Penn. Evaluating distributional models of semantics for syntactically invariant inference. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–43, Avignon, France, April 2012. Association for Computational Linguistics.

Jackie Chi Kit Cheung and Gerald Penn. Towards robust abstractive multi-document summarization: A caseframe analysis of centrality and domain. In *Proceedings of the 51st Annual*

Meeting of the Association for Computational Linguistics, pages 1233–1242, August 2013a.

Jackie Chi Kit Cheung and Gerald Penn. Probabilistic domain modelling with contextualized distributional semantic vectors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 392–401, August 2013b.

Jackie Chi Kit Cheung and Gerald Penn. Unsupervised sentence enhancement for automatic summarization. In *Proceedings of the 2014 Conference on Empirical Methods on Natural Language Processing (EMNLP 2014)*, October 2014.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. Probabilistic frame induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 837–846, June 2013.

James Clarke and Mirella Lapata. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research(JAIR)*, 31:399–429, 2008.

Trevor Cohn and Mirella Lapata. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 137–144, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Andrew McCallum and Sam Roweis, editors, *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167. Omnipress, 2008.

Cleo Condoravdi, Dick Crouch, Valeria de Paiva, Reinhard Stolle, and Daniel G. Bobrow. Entailment, intensionality and text understanding. In Graeme Hirst and Sergei Nirenburg, editors, *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning*, pages 38–45, 2003.

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL 2006*

Main Conference Poster Sessions, pages 152–159, Sydney, Australia, July 2006. Association for Computational Linguistics.

Terry Copeck and Stan Szpakowicz. Vocabulary agreement among model summaries and source documents. In *Proceedings of the 2004 Document Understanding Conference (DUC)*, 2004.

Cristian Danescu-Niculescu-Mizil, Justin Cheng, Jon Kleinberg, and Lillian Lee. You had me at hello: How phrasing affects memorability. In *Proceedings of the 50th Annual Meeting on Association for Computational Linguistics*, pages 892–901, July 2012.

Hoa T. Dang. Overview of DUC 2005. In *Proceedings of the Document Understanding Conference*, 2005.

Hal Daumé and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 305–312, Sydney, Australia, July 2006. Association for Computational Linguistics.

Hal Daumé III and Daniel Marcu. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 449–456. Association for Computational Linguistics, July 2002.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 449–454, 2006.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Paramveer S. Dhillon, Dean Foster, and Lyle Ungar. Multi-view learning of word embeddings via CCA. In *Proceedings of the Twenty-Fifth Annual Conference on Neural Information*

Processing Systems (NIPS 2011), volume 24, pages 199–207, 2011.

Georgiana Dinu and Mirella Lapata. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172, 2010.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*, pages 9–16, 2005.

Bonnie J. Dorr, Christof Monz, Stacy President, Richard Schwartz, and David Zajic. A methodology for extrinsic evaluation of text summarization: Does ROUGE correlate? In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 1–8, 2005.

Micha Elsner and Deepak Santhanam. Learning to fuse disparate sentences. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 54–63. Association for Computational Linguistics, 2011.

Micha Elsner, Joseph Austerweil, and Eugene Charniak. A unified local and global model for discourse coherence. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 436–447, Rochester, New York, April 2007. Association for Computational Linguistics.

Katrin Erk and Sebastian Padó. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 897–906, 2008.

Katrin Erk and Sebastian Padó. Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 92–97, 2010.

Güneş Erkan and Dragomir R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479, 2004.

Oren Etzioni, Michele Banko, and Michael J. Cafarella. Machine reading. Technical report, 2007.

Atefeh Farzindar and Guy Lapalme. Legal text summarization by exploration of the thematic structure and argumentative roles. In Stan Szpakowicz and Marie-Francine Moens, editors, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 27–34, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Elena Filatova and Vasileios Hatzivassiloglou. Event-based extractive summarization. In Stan S. Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 104–111, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Elena Filatova, Vasileios Hatzivassiloglou, and Kathleen McKeown. Automatic creation of domain templates. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 207–214, Sydney, Australia, July 2006. Association for Computational Linguistics.

Katja Filippova. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 322–330, Beijing, China, August 2010. Coling 2010 Organizing Committee.

Katja Filippova and Michael Strube. Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 177–185, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

Charles Fillmore. The case for case. In E. Bach and R. T. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Reinhart, and Winston, New York, 1968.

Charles J. Fillmore. Frame semantics. *Linguistics in the Morning Calm*, pages 111–137, 1982.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131, 2002.

John R. Firth. *Papers in Linguistics, 1934-1951*. Oxford University Press, 1957.

Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28:3–71, 1988.

Gottlob Frege. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50, 1892.

Pascale Fung and Grace Ngai. One story, one flow: Hidden Markov story models for multilingual multidocument summarization. *ACM Transactions on Speech and Language Processing (TSLP)*, 3(2):1–16, 2006.

Pascale Fung, Grace Ngai, and Chi-Shun Cheung. Combining optimal clustering and hidden Markov models for extractive summarization. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering*, pages 21–28, Sapporo, Japan, July 2003. Association for Computational Linguistics.

Michel Galley and Kathleen McKeown. Lexicalized Markov grammars for sentence compression. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 180–187, 2007.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Monolingual distributional similarity for text-to-text generation. In *Proceedings of *SEM 2012: The First Joint Conference on Lexical and Computational Semantics*, pages 256–264, Montréal, Canada, June 2012. Association for Computational Linguistics.

Pierre-Etienne Genest, Guy Lapalme, and Mehdi Yousfi-Monod. Hextac: the creation of a manual extractive run. In *Proceedings of the Second Text Analysis Conference, Gaithersburg, Maryland, USA. National Institute of Standards and Technology*, 2009.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English Gigaword Second Edition, 2005.

Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimental support for a categorical com-

positional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

Amit Gruber, Michael Rosen-Zvi, and Yair Weiss. Hidden topic Markov models. volume 2, pages 163–170, March 2007.

Emiliano Guevara. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of the 2010 Workshop on Geometrical Models of Natural Language Semantics*, pages 33–37, 2010.

Ben Hachey. Multi-document summarisation using generic relation extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 420–429, Singapore, August 2009. Association for Computational Linguistics.

Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado, June 2009. Association for Computational Linguistics.

Zeller S. Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

Liwei He, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin. Auto-summarization of audio-video presentations. In *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*, pages 489–498. Association for Computing Machinery, 1999.

Liwei He, Elizabeth Sanocki, Anoop Gupta, and Jonathan Grudin. Comparing presentation summaries: slides vs. reading vs. listening. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 177–184, New York, NY, USA, 2000. ACM.

Suzanne Hidi and William Baird. Strategies for increasing text-based interest and students' recall of expository texts. *Reading Research Quarterly*, 23(4):465–483, 1988.

Wilfred Hodges. The interplay of fact and theory in separating syntax from meaning. In

Workshop on Empirical Challenges and Analytical Alternatives to Strict Compositionality, 2005.

Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. Automated summarization evaluation with Basic Elements. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 899–902, 2006.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics, 2012.

IBM ILOG CPLEX Optimization Studio V12.4. IBM.

Asghar Iran-Nejad. Cognitive and affective causes of interest and liking. *Journal of Educational Psychology*, 79(2):120–130, 1987.

Hongyan Jing and Kathleen R. McKeown. Cut and paste based text summarization. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pages 178–185, 2000.

Walter Kintsch. Learning from text, levels of comprehension, or: Why anyone would read a story anyway. *Poetics*, 9:87–98, 1980.

Walter Kintsch. Predication. *Cognitive Science*, 25(2):173–202, 2001.

Kevin Knight and Daniel Marcu. Statistics-based summarization—step one: Sentence compression. In *Proceedings of the National Conference on Artificial Intelligence*, pages 703–710, 2000.

Mike Lewis and Mark Steedman. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192, May 2013.

Peng Li, Yinglin Wang, Wei Gao, and Jing Jiang. Generating aspect-oriented Multi-Document summarization with event-aspect model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1137–1146, Edinburgh, Scotland, UK.,

July 2011. Association for Computational Linguistics.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In Stan Szpakowicz and Marie-Francine Moens, editors, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. The automated acquisition of topic signatures for text summarization. In *COLING 2000 Volume 1: The 18th International Conference on Computational Linguistics*, COLING '00, pages 495–501, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. The potential and limitations of automatic sentence extraction for summarization. In *Proceedings of the HLT-NAACL 03 on Text Summarization Workshop*, pages 73–80. Association for Computational Linguistics, 2003.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 510–520, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Annie Louis and Ani Nenkova. Automatically evaluating content selection in summarization without human models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 306–314. Association for Computational Linguistics, August 2009.

Annie Louis and Ani Nenkova. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1157–1168, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

Inderjeet Mani. *Automatic Summarization*. John Benjamins Pub Co, 2001.

Inderjeet Mani, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sund-

heim. SUMMAC: a text summarization evaluation. *Natural Language Engineering*, 8(01): 43–68, 2002.

William C. Mann and Sandra A. Thompson. Rhetorical structure theory: A theory of text organization. *ISI Reprint Series*, 1987.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*, chapter 17. Cambridge University Press, 2008.

Daniel Marcu. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT press, 2000.

Anthony McCallum, Cosmin Munteanu, Gerald Penn, and Xiaodan Zhu. Ecological validity and the evaluation of speech summarization quality. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 28–35, Montréal, Canada, June 2012. Association for Computational Linguistics.

Diana McCarthy and Roberto Navigli. The English lexical substitution task. *Language Resources and Evaluation*, 43(2):139–159, 2009.

Ryan T. McDonald. Discriminative sentence compression with soft syntactic evidence. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, 2006.

Kathleen McKeown, Rebecca J. Passonneau, David K. Elson, Ani Nenkova, and Julia Hirschberg. Do summaries help? A task-based evaluation of multi-document summarization. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 210–217. ACM, 2005.

Kathleen R. McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L. Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. Tracking and summarizing news on a daily basis with Columbia's Newsblaster. In *Proceedings of the Second International Conference on Human Language Technology Research*, pages 280–285. Morgan Kaufmann Publishers Inc., 2002.

Yashar Mehdad, Giuseppe Carenini, Frank Tompa, and Ng. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 136–146, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *Proceedings of ACL-08: HLT*, pages 236–244, 2008.

Jeff Mitchell and Mirella Lapata. Language models based on semantic composition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 430–439, 2009.

Richard Montague. Universal grammar. *Theoria*, 36:373–398, 1970.

Richard Montague. English as a formal language. *Formal Philosophy*, pages 188–221, 1974.

Gabriel Murray, Steve Renals, and Jean Carletta. Extractive summarization of meeting recordings. pages 592–595, 2005a.

Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore. Evaluating automatic summaries of meeting recordings. *Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 33–40, 2005b.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. Evaluating sentence compression: Pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 91–97. Association for Computational Linguistics, 2011.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. Annotated Gigaword. In *Proceedings of the NAACL-HLT Joint Workshop on Automatic Knowledge Base Construction & Web-scale Knowledge Extraction (AKBC-WEKEX)*, pages 95–100, 2012.

Ani Nenkova and Kathleen McKeown. References to named entities: a corpus study. In *Companion Volume of the Proceedings of HLT-NAACL 2003 - Short Papers*, pages 70–72. Association for Computational Linguistics, 2003.

Ani Nenkova and Rebecca Passonneau. Evaluating content selection in summarization: The

pyramid method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, volume 2004, pages 145–152, 2004.

Ani Nenkova and Lucy Vanderwende. The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*, 2005.

Ani Nenkova, Julia Hirchberg, and Yang Liu, editors. *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*. Association for Computational Linguistics, Portland, Oregon, June 2011a.

Ani Nenkova, Sameer Maskey, and Yang Liu. Automatic summarization. In *Tutorial Abstracts of ACL/HLT 2011*. Association for Computational Linguistics, 2011b.

Dirk Ormoneit and Volker Tresp. Improved Gaussian mixture density estimates using Bayesian penalty terms and network averaging. In *Advances in Neural Information Processing*, pages 542–548, 1995.

Karolina Owczarzak and Hoa T. Dang. TAC 2010 guided summarization task guidelines, 2010.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. *Stanford Digital Libraries Working Paper*, 1998.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

Emily Pitler, Annie Louis, and Ani Nenkova. Automatic evaluation of linguistic quality in multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 544–554, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1–10, 2009.

James Pustejovsky. The generative lexicon. *Computational Linguistics*, 17(4):409–441, 1991.

James Pustejovsky. Syntagmatic processes. *Handbook of Lexicology and Lexicography*, 2000.

Dragomir R. Radev and Kathleen R. McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):470–500, 1998.

Joseph Reisinger and Raymond J. Mooney. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117, 2010.

Ehud Reiter, Roma Robertson, and Liesl M. Osman. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1-2):41–58, 2003.

Dan Roth and Wen-tau Yih. Probabilistic reasoning for entity & relation recognition. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 835–841, 2002.

Sebastian Rudolph and Eugenie Giesbrecht. Compositional matrix-space models of language. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 907–916. Association for Computational Linguistics, 2010.

Horacio Saggion and Guy Lapalme. Generating indicative-informative summaries with SumUM. *Computational Linguistics*, 28(4):497–526, 2002.

Horacio Saggion, Juan-Manuel Torres-Moreno, Iria Cunha, and Eric SanJuan. Multilingual summarization evaluation without human models. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1059–1067. Association for Computational Linguistics, 2010.

Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals, and Understanding: An Inquiry Into Human Knowledge Structures*. Lawrence Erlbaum, July 1977.

Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. *Proceedings of the Deep Learning and Unsupervised Feature Learning Workshop of NIPS 2010*, pages 1–9,

2010.

Richard Socher, Eric H. Huang, Jeffrey Pennin, Christopher D. Manning, and Andrew Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809, 2011a.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics, 2011b.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

Alfred Tarski. *The concept of truth in formalized languages*, chapter 8, pages 152–278. Hackett, 1956.

Kapil Thadani and Kathleen McKeown. Supervised sentence fusion with single-stage inference. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1410–1418, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957, 2010.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. Word meaning in context: A simple and effective vector model. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1134–1143. Asian Federation of Natural Language Processing, November 2011.

Kristina Toutanova, Dan Klein, Christoper D. Manning, and Yoram Singer. Feature-rich part-

of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 173–180, 2003.

Stephen Tratz and Eduard Hovy. Summarization evaluation using transformed Basic Elements. In *Proceedings of the First Text Analysis Conference (TAC)*, 2008.

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

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

M. Afzal Upal. Role of context in memorability of intuitive and counterintuitive concepts. In *Proceedings of the 27th Annual Meeting of the Cognitive Science Society*, pages 2224–2229, 2005.

Joachim Wagner, Jennifer Foster, and Josef van Genabith. Judging grammaticality: Experiments in sentence classification. *CALICO Journal*, 26(3):474–490, 2009.

Hanna M. Wallach. *Structured Topic Models for Language*. PhD thesis, 2008.

Stephen Wan, Robert Dale, Mark Dras, and Cecile Paris. Seed and grow: Augmenting statistically generated summary sentences using schematic word patterns. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 543–552, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

Justin Washtell. Compositional expectation: A purely distributional model of compositional semantics. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, pages 285–294, 2011.

Michael White, Tanya Korelsky, Claire Cardie, Vincent Ng, David Pierce, and Kiri Wagstaff. Multidocument summarization via information extraction. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 1–7. Association for

Computational Linguistics, 2001.

Dominic Widdows. Semantic vector products: Some initial investigations. In *Second AAAI Symposium on Quantum Interaction*, 2008.

Yuk W. Wong and Raymond J. Mooney. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

Kristian Woodsend and Mirella Lapata. Multiple aspect summarization using integer linear programming. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 233–243, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

Stephen Wu and William Schuler. Structured composition of semantic vectors. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*, pages 295–304, 2011.

Ainur Yessenalina and Claire Cardie. Compositional matrix-space models for sentiment analysis. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 172–182. Association for Computational Linguistics, 2011.