The Hong Kong Polytechnic University

Department of Computing

Coherence-targeted Text Summarization

ZHANG Renxian

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of Doctor of Philosophy

June 2012

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signed)

ZHANG Renxian (Name of Student)

Abstract

For readers, coherence is no less important than informativeness for a summary. This paper is aimed to improve coherence in automatic text summaries by developing coherence models and related techniques. Different from most other efforts to improve summary coherence, my work treats coherence as an analyzable concept with multi-faceted and multi-disciplinary backgrounds. Specifically, I have explored the technical details of three kinds of coherence – shallow content-driven coherence, deep content-driven coherence, and cognitive model-driven coherence. Shallow content consists of words, phrases, sentences, and discourse units and their literal connections or co-occurrence patterns give rise to coherence. Experiments on single-document as well as multi-document news summarization show that coherence driven by words, entities, sentences, and events can help to better arrange selected summary sentences. Deep content is observed on a macro-text level, which is instantiated by news aspects and speech acts. Focusing on the relations among deep content units, I have applied coherence to both selecting and ordering summary sentences. Relying on human cognitive tendencies, cognitive model-driven coherence is understood as a necessary mechanism in text comprehension. The computational modeling of such coherence, coupled with proposition-level extractive summarization, works successfully for narrative text. To model coherence of different kinds, I have developed novel techniques that are suitable for different genres of text,

including newswire, social media messages, and fairy tales. The extensive experimental results on benchmark or self-compiled datasets have validated the efficacy and robustness of the techniques in various circumstances. Among many of its contributions to the summarization community, my work shows that contrary to what is commonly held, coherence plays a pivotal, instead of ancillary, role in automatic summarization. As one of the few large-scale studies of coherence in summarization, my work is expected to herald a complete theory of coherence and more in-depth studies in coherence-targeted text summarization.

Publications Arising from the Thesis

1. **Zhang, R.**, Li, W., Gao, D., and Ouyang Y. 2013. Automatic Twitter Topic Summarization with Speech Acts. *IEEE Trans. Audio, Speech, and Language Processing*, 21(3):649–658.

2. **Zhang, R.**, Li, W., and Gao, D. 2012. Generating Coherent Summaries with Textual Aspects. In *Proceedings of AAAI 2012*.

3. **Zhang, R.**, Gao, D., and Li, W. 2012. Towards Scalable Speech Act Recognition in Twitter: Tackling Insufficient Training Data. In *EACL 2012 Workshop on Semantic Analysis in Social Networks*, pages 18–27.

4. **Zhang, R.**, Li, W., Liu, N., and Qin Lu. 2012, Information Ordering with an Event-Enriched Vector Space Model for Multi-Document News Summarization. *Computational Intelligence*, Article accepted with minor revisions (revision submitted).

5. **Zhang, R.**, Li, W., and Gao, D. 2013. Towards Content-level Coherence with Aspect-Guided Summarization. *ACM Trans. Speech and Language Processing*, Article accepted.

6. **Zhang, R.**, Li, W., Gao, D., and Hou Y. 2012, Coherent Narrative Summarization with a Cognitive Model. *Computational Linguistics*, Article submitted.

7. Cai, X., Li, W., and **Zhang R.** 2012. Enhancing Diversity and Coverage of Document Summaries through Subspace Clustering and Clustering-based Optimization. *IEEE Trans. Cybernetics*, Article submitted.

8. Cai, X., Li, W., and **Zhang R.** 2012. Combining Co-clustering with Noise Detection for Theme-based Summarization. *ACM Trans. Speech and Language Processing*, Article submitted.

9. Gao, D., Li, W., and **Zhang, R.** 2012. Twitter Hyperlink Recommendation with User-Tweet-Hyperlink Three-way Clustering. In *Proceedings of CIKM 2012*.

10. Gao, D., Li, W., and **Zhang, R.** 2012. Hyperlink Recommendation in Twitter Based on Tensor Clustering. *CIKM 2012 Poster Track*, Article submitted.

11. Gao, D., Li, W., and **Zhang, R.** 2012. Time-Oriented Summarization: A New Application for Timely Updated Twitter Trending Topics. *CIKM 2012 Poster Track*, Article submitted.

12. **Zhang, R.**, 2011. Sentence Ordering Driven by Local and Global Coherence for Summary Generation. In *ACL 2011: Student Session*, pages 6–11.

13. **Zhang, R.**, Gao, D., and Li, W. 2011. What Are Tweeters Doing: Recognizing Speech Acts in Twitter. In *AAAI-11 Workshop on Analyzing Microtext*, pages 86–91.

14. **Zhang, R.**, Ouyang, Y., and Li, W. 2011, Guided Summarization with Aspect Recognition. In *Proceedings of TAC 2011*.

15. Cai, X., **Zhang, R.**, and Gao, D., 2011. Simultaneous Clustering and Noise Detection for Theme-based Summarization. In *IJCNLP 2011*, pages 491–499.

16. Gao, D., **Zhang, R.**, Li, W., Lao, Y. K., and Wong, K. F. 2011. Learning Features through Feedback for Blog Distillation. In *Proceedings of ACM SIGIR 2011*, pages 1085–1086.

17. Ouyang, Y., Li, W., **Zhang, R.**, Lu, Q. 2011. Applying Regression Models to Query-focused Multi-document Summarization. *Information Processing and Management (IPM)*, Article under review (revision submitted).

18. **Zhang, R.**, Li, W., and Lu, Q. 2010, Sentence Ordering with Event-Enriched Semantics and Two-Layered Clustering for Multi-Document News Summarization. In *COLING 2010: Poster Volume*, pages 1489–1497.

19. **Zhang, R.**, Ouyang, Y., and Li, W. 2010, PolyU's Experimentation with Guided Summarization. In *Proceedings of TAC 2010*.

20. Ouyang, Y., Li, W. and **Zhang, R.**. 2010, Keyphrase Extraction Based on Core Word for Identification and Word Expansion. In Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, pages 142–145.

21. Ouyang, Y., Li, W., **Zhang, R.**, and Lu, Q. 2010. A Study on Position Information in Document Summarization. In *COLING 2010*, pages 919–927.

Acknowledgements

There is a long list of people I am grateful to for the completion of this doctoral dissertation. Three years ago, I came to Hong Kong with little more than a linguistics background and an inquisitive mind. Dr. Wenjie Li accepted my application, my background, and my limited knowledge about Computational Linguistics, making a decision that will possibly change the life of me and my family. Without her appreciation of my potential in the first place, I would never have the chance of writing down these words.

Ever since day one, Dr. Li as my supervisor has shown immeasurable trust in me, powering me to reach new heights. She has guided me to successfully sail through the whole process of writing this dissertation, giving comments, suggestions, support, and encouragement when they are most needed. In life, Dr. Li is a good friend and mentor, sharing with me frustration at each rejection and delight in each acceptance.

The Department of Computing of the Hong Kong Polytechnic University has an amazing faculty. Professor Qin Lu, my co-supervisor, has also shown deep trust in me. During seminar sessions, I have benefited considerably from her instructions and suggestions. Many ideas and experimental designs in this dissertation have been significantly improved after discussions with Dr. Grace Ngai, Dr. Korris Chung, Dr. James Liu, and Professor Jane You.

I must thank my wife for all the support and sacrifice for all those years. She

shouldered all the family burdens and played both mother and father during my absence from family duty. Her understanding and encouraging words are the greatest relief in my darkest and loneliest moments. I also thank my six-year-old daughter for tolerating and loving a long absent daddy.

The colleagues and visitors in the Chinese Computing and Natural Language Processing Lab have offered constant support and help throughout the years. I thank you all: Dr. You Ouyang, Mr. Dehong Gao, Dr. Yuexian Hou, Dr. Xiaoyan Cai, Dr. Jun Xu, Ms. Shasha Li, Ms. Yaoyun Zhang, Mr. Jian Xu, Dr. Jintao Tang, and Mr. Oscar Lai.

Several overseas mentors deserve my sincere gratitude. Professor Graeme Hirst, a renowned computational linguist and a great mentor to me and my wife, has been supporting and encouraging my work for all the years. Dr. Micha Elsner has given me invaluable suggestions on coherence modeling when he tutored me on ACL 2011. Professor Lenhart Schubert, an expert on event logics, has helped to shape up some of my burgeoning ideas in this paper.

Last but not the least, my thanks go to the friends I have made in Hong Kong for making my life in Hong Kong much easier and more enjoyable – Dr. Chenyu Hong, Mr. Ji, Mr. Dejun Zheng, Mr. Shibiao Wan, and Mr. Junyi Chai.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

"Queen: More matter with less art."

William Shakespeare, *Hamlet, Act II, Scene ii*

"Language is not merely a set of unrelated sounds … it is a total coherent system of these integrating with each other …"

Kenneth L. Pike, *Linguistic Concepts*

Over half a century ago, Luhn (1958) brought into our cognizance the automatic technology of text summarization. Nurtured in an age of information explosion and fueled by people's increasing demand of efficient information consumption, automatic text summarization ("text summarization" for short hereafter if no confusion arises) has since blossomed into a full-fledged field among Natural Language Processing (NLP) applications.

The topic of this dissertation is coherence-targeted text summarization, a familiar task with a much needed focus – producing coherent and readable summaries for human readers. Different from most other works on coherence in summarization, this work regards coherence as a joint effect under the force of **a number of textual factors** and subject to **multi-disciplinary accounts**. In this opening chapter, I will explain in more detail what those words mean and give an overview of my work. But before that, let's take a brief tour in the realm of text

summarization.

1.1 Text Summarization: A Brief Tour

By definition, text summarization is “the process of distilling the most important information from a text to produce an abridged version for a particular task and user” (Mani and Maybury, 1999). It was born out of the need for an efficient way to obtain the most important information from a large body of documents without reading all of them. With a history of over 50 years, it proves to be one of the most vigorously explored frontiers in NLP applications. What was once believed to be a human-privileged creative task is now extensively automated in modern computing labs and commercial packages.

1.1.1 Taxonomies of Text Summarization

Over the years, researchers have come up with terms and theories to better calibrate this task. Taxonomically, text summarization can be classified with various criteria. According to information coverage, there are **extractive** and **abstractive** summaries. The former (also called extracts) are made up of sentences or phrases verbatim from the source documents, and the latter (also called abstracts) contain novel sentences via content reformulation or paraphrasing. According to function, we can divide summaries into **indicative**, **informative**, and **critical** types. Indicative summaries indicate the content of a document with no further details; informative summaries provide such details

and can act as an abridged surrogate for the source document; critical summaries represent the summarizer's attitude and opinion about the source document and are thus highly human-privileged. Taking a utilitarian criterion, we identify **generic** summaries and **query-focused** summaries. Generic summaries don't address a particular user need whereas query-focused summaries are produced in response to a user need or query. The last criterion is the number of source documents, which can distinguish **single-document summarization** from **multi-document summarization**. Table 1.1 lists the major types of summarization according to the different criteria. I should point out that most current researches are biased toward extractive, informative, query-focused, multi-document summarization.

Criterion	Information Coverage	Function	Use	Source Documents
Types	*Extractive* *Abstractive*	*Indicative* *Informative* *Critical*	*Generic* *Query-focused*	*Single-document* *Multi-document*

Table 1.1: Taxonomies of Text Summarization

1.1.2 Components of Text Summarization

Technically, there are different models for the general architecture of text summarization. A classical one is attributed to Mani (2001: 14), who identifies

analysis, **transformation**, and **synthesis** as the three fundamental phases in a "high-level architecture of a summarizer". Both analysis and synthesis address some "internal representation" of a text through deep semantic and logical parsing. Transformation, however, concerns the condensation of information from the source and is thus regarded as the essential phase of summarization. A terminological variant of this tripartite model is adopted by Jones (1999), consisting of **interpretation** (from source text to source representation), **transformation** (from source representation to summary representation), and **generation** (from summary representation to summary text). A more extract-oriented model is assumed by Hovy (2005), who establishes **topic identification**, **interpretation**, and **summary generation** as three distinct stages of summarization. Topic identification corresponds to the selection of the most salient units (e.g., sentences), which may suffice for simple (extractive) summarization. Interpretation and summary generation are aimed at higher-quality, abstractive, human-like output. Interpretation means the transformation of words to concepts by simulating the human understanding. Summary generation is aimed at reducing dysfluencies and improving readability. A similar approach is taken by Jurafsky and Martin (2009: 824), who identify three stages of summarization: **content selection**, **information ordering**, and **sentence realization**. Content selection and sentence realization are basically Hovy's (2005) topic identification and summary generation, with a narrower

focus on sentence extraction. The intermediary information ordering concerns the ordering of the selected sentences in the output.

1.1.3 Evaluation Methods

Given all the above components, the research on text summarization is not complete without evaluation of the output for content/informativeness and coherence/readability or their combination (Dang and Owczarzak, 2008). Summarization evaluation methods include **intrinsic** evaluation and **extrinsic** evaluation. Intrinsic methods evaluate a system for the quality of its output by doing cross-summary comparisons, so that the system-produced summary is evaluated against other system-produced summaries, simple baselines, or human-produced summaries. Since human-produced summaries lack agreement (Rath et al., 1961), automatic summaries can be compared against chosen sets (e.g., intersection, union) of multiple human-produced summaries (Salton et al., 1997). The lead baseline, which is very hard to beat in single-document newswire summarization, is widely used (Brandow et al., 1995). Initiated by (Lin and Hovy, 2003), the content-oriented intrinsic evaluation has been fully automated and implemented as ROUGE (Lin 2004) and BE (Hovy et al., 2005) in the DUC/TAC[1] competitive tasks. Coherence or overall quality-oriented intrinsic evaluation can be done by human judges according to the Pyramid

[1] Organized by National Institute of Standards and Technology (NIST), the Document Understanding Conference (DUC, 2001-2007) and the successive Text Analysis Conference (TAC, since 2008) summarization track is a major competitive event in the text summarization community. Visit http://www-nlpir.nist.gov/projects/duc/pubs.htm) and http://www.nist.gov/tac/ for more information.

Method (Passonneau et al., 2005).

Extrinsic methods evaluate a system by means of external tasks and makes cross-species (e.g., summary vs. source document) comparisons. A classic example is reported in (Morris et al., 1992), which compares summaries and the original documents in solving GMAT reading comprehension questions. Other adopters of extrinsic evaluation methods include Firmin and Chrzanowski (1999), Mani et al. (2002), etc.

1.1.4 Text Summarization Systems

In this section, we will document major summarization systems since 1980. All of them are denoted by acronyms and most are not proprietary (cf. Microsoft Word's AutoSummarize).

The first batch of systems, spanning the time period 1980–1990, typically incorporates text understanding and knowledge engineering techniques, which is motivated by theories about human cognition in summarization (Endres-Niggemeyer, 1998: 310–312). Representative systems include **FRUMP** (Dejong, 1982) using event schemata and sketchy scripts to organize its domain knowledge, **SUSY** (Fum et al., 1982) using a Natural Language Understanding (NLU) engine to summarize scientific texts, **SCISOR** (Jacobs and Rau, 1990) using syntactic and semantic analysis to summarize multiple documents, **TOPIC** (Hahn, 1990) using knowledge base concepts and ontology to generate indicative

summaries, and **PAULINE** (Hovy, 1988) using semantic representations to adapt summaries to specific user needs. Due to intensive knowledge engineering and human labor, most of those systems are not presently sustained and some (like SUSY) were only partly implemented.

The next decade (1990–2000) saw the birth of a new generation of summarizers. Though some of them inherit the knowledge processing legacy, most demonstrate people's inclination to treat text summarization as a knowledge-independent NLP task. **SIMPR** (Gibbs, 1993) is an indicative-summarizing system, producing indexes by incorporating both morpho-syntactic constraints and knowledge-based generation rules. McKeown et al. (1995) report two systems generating abstracts of domain-specific documents — **STREAK** and **PLANDOC** — summarizing basketball game results and telephone network planning activity respectively, using Natural Language Generation (NLG) techniques. **SumGen** (Maybury, 1999) summarizes from structured data and consists of three main modules — content selection, aggregation, and presentation. **SUMMON** (McKeown and Radev, 1995) is a well-known multi-document summarizing system built on NLU models and templates, consisting of a content planner and a linguistic component. The most representative system in this period is **SUMMARIST** (Hovy and Lin, 1999), a modulated, comprehensive system that deals with both extraction and abstraction. Its four major modules are preprocessing, topic identification, topic

interpretation/concept fusion, and summary generation.

Since the beginning of this century, increased interest in automatic summarization and public competitive events (DUC) has stimulated the growth of new systems. With each participant since DUC 2001 counted as a distinct system, the total will be in the hundreds. Here I will only discuss some representative, well-known, or publicly available systems. The first is **SumUM** (Saggion and Lapalme, 2002), which is targeted at technical documents by addressing the need of abstracts and integrates indicative and informative summarization. Lin and Hovy (2002) built a multi-document version of their single-document SUMMARIST — **NeATS** — which distinguishes itself at DUC 2001. Another sophisticated system that made its debut at DUC is **GISTexter** (Harabagiu and Lacatusu, 2002), which uses Information Extraction (IE) techniques to generate both single-document and multi-document summaries. A discourse-level summarizer during the period is **PALSUMM** (Polanyi et al., 2004), which extracts sentences based on discourse structure and produces summaries preserving the language style of source documents. A well-known public domain and open source platform for multi-document summarization is **MEAD**[2] (Radev et al., 2004). It has implemented a number of summarization algorithms and provided popular classifiers for supervised content selection. Another publicly available and mass audience-oriented system is Columbia

[2] http://www.summarization.com/mead

University's **Newsblaster** [3] (McKeown et al., 2002), a multi-document summarizer for newswire articles. It works by crawling the Web for news articles, clustering them on specific topics and events and producing multi-document summaries for each event cluster.

Table 1.2 lists the three major stages in the development of summarization systems, along with representative examples.

Period	1980 – 1990	1990 – 2000	2000 – now
Technical Highlights	Text understanding, Intensive knowledge engineering	NLU, NLG, Less knowledge engineering	Multi-document oriented, IE, Discourse structure based, open source
Examples	*FRUMP, SUSY, SCISOR, TOPIC, PAULINE*	*SIMPR, STREAK, PLANDOC, SumGen, SUMMON, SUMMARIST*	*SumUM, NeATS, GISTexter, PALSUMM, MEAD, Newsblaster*

Table 1.2: Systems of Text Summarization

1.2 Research Motivation

In the growing community of text summarization, research efforts are

[3] http://newsblaster.cs.columbia.edu/

traditionally lopsided towards the information contained in summaries, not the presentation of the information, although the latter is arguably no less important. This dissertation is exclusively devoted to one crucial aspect of summary presentation – **coherence**.

The concern with coherence is motivated by the ultimate purpose of automatic text summarization – to provide human readers, not machines, with a sufficiently abridged summary of a long document or document set to facilitate efficient information processing. In this sense, the summary serves as a surrogate for the original document(s) in terms of informativeness and expressiveness. Informatively, the summary is expected to maximally reproduce the original document's essential information in a reduced space. Expressively, it is expected to convey the information in an intelligible and coherent way to human readers. In practice, a coherent summary is as preferable to an incoherent summary as a sentence-based summary is to a keyword-based "summary".

A deciding factor for the expressiveness of a summary is coherence, i.e., how well textual components such as sentences are connected to each other and stand together in the whole text. Failure to address coherence will defeat the purpose of summarization because coherence is also interrelated with informativeness. An incoherent summary, e.g., with unresolved anaphors or a disordered structure, will thwart the communication of the content to a human reader, no matter how informatively faithful it is to the original document.

As summaries are written texts intended for human readers with cognitive abilities, my quest is further motivated by concerns from linguistics and cognitive psychology.

Linguists have long regarded textual coherence as an indispensable quality for the proper functioning of language (Halliday and Hasan, 1976). According to text linguistics and discourse study, coherence takes effect on two levels: global and local, which involve different discourse and mental processes (Tapiero, 2007). They are the underlying threads that interweave textual pieces into a systematic fabric and drive various models to account for text comprehension.

Cognitive psychologists explain coherence as a reader's cognitive need for coming to terms with words or concepts activated during the whole process of text comprehension. Cognitive models of text comprehension and coherence (van den Broek et al., 1996; Kintsch, 1998, 2001) employ similar human memory organizations to account for the effect of successful comprehension – one that is coherent.

Motivated by the purpose of text summarization per se and findings from linguistics and cognitive psychology, I will pursue the pivotal role played by coherence in text summarization.

1.3 Research Overview

A distinctive feature of this research is that coherence is not taken for

granted or to be defined in passing, as is the practice in many coherence-related summarization works. Situated in different theoretical backgrounds, coherence has different implications for model design or algorithm development. Therefore I will attempt a multi-dimensional approach to explore the multi-faceted and multi-disciplinary nature of coherence and its role in summarization. In this work, I will choose three major dimensions of coherence:

1. **Shallow content**-driven coherence
2. **Deep content**-driven coherence
3. **Cognitive model**-driven coherence

Taking the shallow-content dimension, coherence is defined in a micro-textual scope, i.e., cohesion patterns between words, repetition and continuation patterns among discourse entities, or sentences with overlapping words. Shallow content-driven coherence has predominantly local features. It is also semantically lean because no deep understanding of the text content is needed. Many existing summarization works with a focus on coherence are engaged with this kind of coherence.

Beyond entities, words, sentences, etc., my research on coherence extends to a macro-textual scope, taking into account rich textual components that are usually domain-specific (e.g., organization of news aspects or story plot units) or

genre-specific (e.g., organization of speech acts in written speech). Bearing more global features, deep content-driven coherence is semantically rich and indicative of deep text understanding. This kind of coherence distinguishes my work from most others.

Accommodating cognitive models of text comprehension and coherence, my work reaches out to an extra-textual scope. Coherence is not only derived from the text being processed, but also rooted in human cognitive mechanisms – mental lexicon, word association patterns, semantic network in long-term memory, etc. that are sedimented from years of linguistic contact. To capture cognitive model-driven coherence, algorithms need to simulate the cognitive process of text comprehension as a concerted effort between different compartments in human memory. This kind of coherence is seldom reported in the summarization community.

The above descriptions, however, do not imply that coherence-targeted summarization is a brand-new summarization variant or that it cannot be reconciled with the existent approaches and models. My primary goal is to instill coherence into the general process and integrate coherence-based techniques with the major techniques of text summarization.

The shallow content-driven coherence is compatible with almost all existing summarization approaches because it can direct the arrangement of selected summary contents (usually sentences) to optimize the readability of the output.

The role of coherence in this sense is thus **information ordering**. I will show that information ordering improves coherence among the selected sentences for multi-document as well as single-document summarization by developing algorithms that leverage lexical cohesion, entity repetition, sentence overlap, and event relation.

The deep content-driven coherence is more deeply integrated into the general process of summarization, affecting not only information ordering but also **sentence selection**. Better yet, drawing on IE techniques, I can extract coherent contents that are not necessarily sentences and generate readable **abstractive** summaries using NLG techniques. I will establish two models to show the role of coherence in this sense – a semantic content model leveraging the news aspects for the domain of news report and a pragmatic content model leveraging the speech acts for the domain of social media messaging.

The cognitive model-driven coherence assumes a central role in the process of summarization. **Content selection** applies to sentences as well as **propositions**, the latter being the fundamental unit of cognitive processing. Therefore extractive summarization on the propositional level is possible. To model coherence with a cognitive background, I will simulate the long-term human memory by building a semantic network from a large corpus like Wiki and design algorithms to account for the information flow among different compartments of human memory. I will then apply the computational model to

news reports and fairy tales.

Table 1.3 sketches the three-dimensional approach taken in my research, including the three major kinds of coherence with their theoretical backgrounds, working scopes, coherence processing units, and roles in summarization.

Type of coherence	**Theoretical background**	**Working scope**	**Processing unit**	**Role in Summarization**
Shallow content-driven	Lexical semantics, Discourse analysis	Micro-textual	Words, Sentences, Discourse units	Information ordering
Deep content-driven	Semantics, Pragmatics	Macro-textual	Domain-specific content units, speech acts	Content selection (and ordering), Abstraction
Cognitive model-driven	Cognitive psychology	Extra-textual	Propositions	Content selection, (Proposition-level) Extraction

Table 1.3: A Three-dimensional Approach to Coherence

According to this architecture, my work spans both extractive and abstractive summarization, covers various summarization components, and addresses different domains and genres of text (news report, story, social media

message), showing how coherence can be captured in text summarization.

1.4 Research Contributions

My work is not the first of its kind on summarization with an emphasis on coherence. Most of such works, however, treat coherence as a self-evident quality of summary. Little thought has been given to the multi-faceted nature of coherence and its implications for developing models and algorithms of summarization. By contrast, I regard coherence as an analyzable concept and a key player in the whole process of summarization. Subject to different interpretations, coherence has different implications for developing summarization models. My work proves that formulating coherence as such is beneficial for improving the state of the art and breaking new fertile ground of text summarization. This is the prominent contribution of my work.

In terms of the three dimensions I use to calibrate coherence, the specific technical contributions are listed in the following.

- For shallow content-driven coherence,

1. Using lexical cohesion and sentence overlaps, I find that **sentence reordering creates significantly better single-document summaries**, a conclusion that overthrows a long-held assumption.

2. Combining entity relation and verb semantics, I propose **a novel approach to ordering sentences for multi-document summaries by using**

event information, which significantly outperforms event-agnostic models.

- For deep content-driven coherence,

3. In the news domain, I identify content units (news aspects) as coherence contributors and develop **a probabilistic model that unifies sentence selection and information ordering** to maximize the coherence between such content units, outperforming similar models unaware of such content units.

4. In the social media message (Twitter) domain, I introduce speech acts – pragmatic content units – as coherence contributors and develop **an abstractive system by finding coherence in and summarizing over speech acts and speech act-dominated phrases**. This innovative system defeats all known competitors.

- For cognitive model-driven coherence,

5. Believing coherence is not merely an effect on a text being read, I **model coherence with an extra-textual dimension on cognitive accounts**. The model **simulates the human aspect of coherence** and outperforms more traditional competitors on both news reports and fairy tales.

6. I adopt **propositions as selection units to better capture the coherence** in this dimension and propose a novel approach to **proposition-level extractive summarization**. This approach proves to be superior to the sentence-level extractive counterpart.

1.5 Structure of Dissertation

For clarity, let's now chart a map of this dissertation.

Chapter 1, this chapter, first sets the background of this research in the whole landscape of text summarization and then outlines the current research, explaining the motivations and presenting the contributions.

Chapter 2 surveys over half a century's literature on text summarization. After presenting the mainstream techniques developed with little or no concern of coherence, I shift the focus to works devoted to coherence, covering its independent development outside the sphere of text summarization and its accommodation in summarization. To complete the survey, I also discuss works on coherence evaluation.

Chapters 3 to 5 constitute the technical body of the dissertation. Chapter 3 discusses shallow content-driven coherence in summarization, where I develop coherence models for both single-document and multi-document summarization.

Chapter 4 is devoted to deep content-driven coherence in summarization, where I concretize deep content as news aspect and speech acts for generating coherent summaries on two distinct domains: news report and social media messaging.

Chapter 5 is engaged with cognitive model-driven coherence in summarization, where I approach coherence from the perspectives of cognitive psychology and model coherence using concepts, theories, and models whereof. I

present evaluation results on both news reports and fairy tales.

Chapter 6, the last chapter, pieces together the major findings and technical results in the broad picture of coherence in summarization. I reiterate the major contributions in this work and also point out future extensions.

Chapter 2: Literature Review

Since its inception in the late 1950s, automatic text summarization has been actively pursued for more than half a century and proved to be one of the most vigorously explored frontiers in NLP applications. The past decades have witnessed the mushrooming of theories, models, algorithms, implemented systems, as well as our enhanced understanding of text summarization per se, including general and coherence-focused approaches. Work in this area has so flourished that the turn of the century saw two compendiums of the state-of-the-art models and techniques: (Mani and Maybury, 1999) and (Mani, 2001). On the other hand, coherence enjoys a much longer history of academic efforts in various disciplinary backgrounds, especially linguistics and cognitive psychology. Linguists and cognitive psychologists have established theoretical models to account for coherence in the language system or human cognition, which are the precursors of computational models of coherence.

This chapter is intended to draw a large picture of text summarization with coherence-related works in the foreground. Specifically, 2.1 surveys general-purpose summarization techniques, many of which can be extended with an additional concern of coherence; 2.2 introduces the major theories and models of coherence that inspire the multi-dimensional approaches in my work; 2.3 reviews various coherence models in text summarization to complete the picture.

2.1 General-purpose Summarization Techniques

In this section, I will address the major summarization techniques developed over the decades. Most of them are not built on coherence models or oriented for coherence, serving mainly the purpose of informativeness. It is in this sense that they are considered traditional or "general-purpose". Nevertheless, they jointly lay a solid foundation for coherence-based summarization by providing widely applicable frameworks, models, and algorithms with possible extensions to coherence concerns. It is for this reason that I will first review a number of works that generally inform text summarization researchers, coherence-minded or not, by dwelling on four major camps of summarization – shallow text-based, discourse-based, graph-based, and machine learning-based.

2.2.1 Shallow Text-based Approaches

Such approaches make use of shallow text features, such as word frequency, length, position, text layout, keywords, etc., to find important units (usually sentences) for summaries. The summarization algorithms are usually heuristically or empirically guided. The earliest such approach is reported by Luhn (1958), who measures sentence summary-worthiness by word frequency only, assuming that the summary sentences must contain the most frequent words in a text.

Edmundson (1969) extends Luhn's work by considering cue phrases, title, and location in addition to high-frequency words. The findings that the

combination of cue-title-location gives the best performance and that location is the best individual feature are often quoted as the most substantial achievements made by shallow feature studies.

Pollock and Zamora (1975) apply the shallow feature-based approach to chemical abstracts, but similar studies are rarely reported after (Edmundson, 1969). However, an important amendment is made by (Lin and Hovy, 1997), where the finding about location ("Position Hypothesis") is rigorously tested. The authors use the Ziff-Davis corpus, composed of document with keywords and abstracts, and evaluate the position-based extract.

Works making use of word relations such as synonymy, hyponymy, and meronymy are an extension of those relying only on word repetition, a shallow text feature. The availability of machine-readable dictionaries and thesauri like WordNet (Fellbaum, 1998) facilitates the growth of word relation-based summarization.

Based on previous studies on lexical relations (Morris and Hirst, 1991), Barzilay and Elhadad (1997) explore summarization using lexical chains, a useful tool to measure the connectedness between sentences with reference to lexical relations. The efficiency of the lexical chain-based method is later improved by Silber and McCoy (2000), who use meta-chains, a special data structure, to achieve a linear core runtime.

Adopting the simple word frequency and a redundancy control mechanism resembling Maximal Marginal Relevance (MMR, Carbonell and Goldstein,

1998), Nenkova and Vanderwende (2005) build a simple summarizer called SumBasic, which is one of the top systems on DUC 2004. Its extended version, SumFocus (Vanderwende et al., 2007) adjusted for query-focused summarization with query expansion, continues the success. This shows how far a simple method based on shallow text features can go. In fact, word frequency alone, as is shown by Nenkova et al. (2006), can be very effective so that a simple frequency-based summarizer using a good composition function or combined with word position information (Yih et al., 2007) can generate summaries comparable to state-of-the-art systems.

2.2.2 Discourse-based Approaches

Most of the approaches introduced above regard the source text as a collection of sentences and operationalize their core algorithms on the sentence or word level. An alternative family of approaches, however, take the whole discourse in their view and extract discourse units on this level. There are two strains in this family: one that studies the coherence relations between discourse units, which will be covered in 2.3, and the other that is simply based on structural characteristics of a discourse, which will be introduced now.

Teufel and Moens (1999) explore discourse-level summarization by studying the "argumentative structure" of science research papers. They identify 7 rhetorical roles (Background, Topic/Aboutness, Related Work, Purpose/Problem, Solution/Method, Result, Conclusion/Claim) as global rhetorical features to

extract sentences. Similarly, Blair-Goldensohn and McKeown (2006) use "rhetorical-semantic" relation (Contrast and Cause) to generate query-focused summaries. A similar approach is adopted by Cristea et al. (2005).

Unresolved anaphora, both nominal and pronominal, is a discourse-level problem. Orăsan (2007) and Steinberger et al. (2007) show that reliable anaphora resolution enhances summarization. But conflicting observations (Mitkov et al., 2007) exist.

The possibility of using paragraphs, instead of sentences, as extraction units is tested by Salton et al. (1997), who utilize text structuring and segmentation. A paragraph relationship graph is established for a text, based on which topic-bearing paragraphs can be identified and extracted with "bushy" or "depth-first" algorithms.

A more comprehensive application-oriented endeavor is reported by Strzalkowski et al. (1999). The authors exploit the Discourse Macro Structure (DMS), such as the background-main story structure in most news-style documents. Like Salton et al. (1997), they work on the paragraph level. They also rely on shallow (including DMS) features to score paragraphs, such as titles, cue words, location etc.

The PALSUMM (Polanyi et al., 2004) introduced in 1.1.4 is an implemented system that works on the syntactic and semantic structure of the discourse. Thione et al. (2004) discuss the discourse-level syntactic structure used in it, such as coordination, subordination and n-aries.

The last group of this camp consists of models tailored for specific domains of discourse, such as the aforementioned technical domain (Teufel and Moens, 1999), legal domain (Grover et al., 2003; Farzinder and Lapalme, 2004), and medical domain (McKeown et al., 1998; Elhadad and McKeown, 2001).

2.2.3 Graph-based Approaches

The graph-based technology has found increasing applications in NLP and Information Retrieval fields (Mihalcea and Radev, 2011). Graph-based approaches to textual summarization have also flourished in the recent decade.

Mani and Bloedorn (1999) report a complicated query-based multi-document summarization system that is built on a standard "analysis-refinement-synthesis" architecture. In the analysis stage, documents are represented as graphs with words as nodes and word attributes and relations as edges. In the refinement stage, a spreading activation algorithm is used to reweight the nodes based on the user's query.

Developed on the idea of sentence centrality, the LexRank algorithm proposed by Erkan and Radev (2004) makes use of text graph and the PageRank algorithm. Instead of computing the conventional centroids, it makes use of eigenvector centrality operated on a connectivity matrix of the graph representation of sentences. A similar and equally representative model is Mihalcea's (2004, 2006) TextRank, which is simply built on sentence similarities computed from term overlaps. The model uses a PageRank-style (Brin and Page,

1998) algorithm to rank sentences for extractive summarization.

In Wei et al. (2009), a graph model based on the LexRank model is established that takes into account both generic summarization and query-oriented summarization. When applying a PageRank-style algorithm, the authors consider document factors when calculating sentence similarities. Also adopting a PageRank-style algorithm, Wan and Yang (2008) use “topics”, cast as sentence clusters, in addition to the commonly used documents, sentences or words, as nodes in the text graph.

Adopting the mutual reinforcement paradigm, Zha (2002) adopts a bipartite graph to capture the interactions between words and sentences in order to rank them simultaneously. A similar bipartite graph scheme is used by Cai et al. (2010), who simultaneously rank and cluster sentences in order to discover summary-worthy sentences.

The graph-based approaches can be adapted to many genres of text other than news, such as books (Mihalcea and Ceylan, 2007), opinions (Ganesan et al., 2010), and biomedical articles (Morales et al., 2008).

2.2.4 Machine Learning-based Approaches

Most of the aforementioned approaches are unsupervised, i.e., without human summaries as reference to sentence selection. If human summaries are available, supervised summarization is feasible, which can leverage machine learning techniques.

The classical machine learning-based summarization work is by Kupiec et al. (1995), known as KPC. The authors use a set of five features (sentence length, fixed-phrase/cue phrase, paragraph (location), thematic word (frequency), uppercase word) to train a Bayesian classifier.

A direct inheritor of KPC is (Myaeng and Jang, 1999), which applies a similar approach to summarizing Korean texts, with two noteworthy modifications: 1) using a text component identification model to filter sentences before ranking and selecting them; 2) limiting the KPC approach to individual features and then computing the final score for each sentence with the Dempster-Shafter combination rule.

Aone et al. (1999) experiment extensively with different feature combinations and calculation methods. Like KPC, they find summaries that are based on machine learning significantly better than those that are not.

Recently, researchers experiment with alternative learning methods by using untypical learning models for NLP such as Particle Swarm Optimization (PSO, Binwahlan et al., 2009), regression models (Schilder and Kondadadi, 2008; Ouyang et al., 2011), feature selection (Wong et al., 2008) and non-summary training data (Fuentes et al., 2007).

In addition to sentence selection, recently sentence compression – an important sentence generation technology – has relied on machine learning by using parallel document / summary corpora. Exemplary methods are maximum entropy (Riezler et al., 2003), noisy channel model (Knight and Marcu, 2000),

large-margin learning (McDonald, 2006), and Integer Linear Programming (Clarke and Lapata, 2007). Joint learning models that address both extraction and compression have also been proposed, such as (Martin and Smith, 2009) and (Berg-Kirkpatrick et al., 2011).

2.2 Theories and Models of Coherence

Early study of coherence in linguistics is started by Halliday and Hasan (1976), for whom coherence is a textual effect achieved by linguistic devices of cohesion, including reference, substitution, ellipsis, conjunction and lexical cohesion. Hoey (1991) extends the study of cohesion to the study of patterns of lexis in text.

The view of coherence as an effect of cohesive devices is challenged by text linguists, who hold that coherence is the “continuity of senses” and “mutual access and relevance within a configuration of concepts and relations” that involves readers (De Beaugrande and Dressler, 1996). Similar viewpoints, such as treating coherence as the result of the interaction between text and reader, are also held by Blum-Kulka (1986).

In cognitive psychology, a large body of research focuses on text comprehension, of which coherence is treated as a dynamic component because a coherent representation from the text underlies successful text understanding (van Dijk and Kintsch, 1983; Tapiero, 2007). To capture coherence in this flavor, many models have been developed, such as the Construction-Integration Model

(Kintsch, 1998), the Structure Building Framework (Gernsbacher, 1990), and the Landscape Model (van den Broek et al., 1996).

Many coherence models in NLP and text summarization make a distinction between **global coherence** and **local coherence**. Global coherence characterizes the global pattern of textual units (sentences or paragraphs) in meaningful relations to each other, thus the whole text is usually represented as a tree or graph. There are various accounts for coherence relations in this sense. In an early work, Hobbs (1985) puts forth a group of 10 coherence relations: (occasion, evaluation, parallel, elaboration, background, explanation, contrast, violated expectation, generalization, exemplification). A similar taxonomy is made by Kehler (2002), which is philosophically justifiable and linguistically explicatory. Coherence relations are also recast as "rhetorical relations" in the seminal paper by Mann and Thompson (1988) and lay the foundation of the Rhetorical Structure Theory (RST), an extensively used model in coherence-based NLP. In the realm of discourse study, Unger (2006) observes that global coherence can be accounted for by the ostensive-inferential account of relevance theory (Sperber and Wilson, 1995).

Unlike global coherence, local coherence is concerned with how information flows smoothly from one sentence to the next. Therefore most researches in this camp focus on adjacent sentence pairs and their coherence patterns are manifested on an entity level. The Centering Theory (CT) proposed by Grosz et al. (1995) is a theoretical prototype in the local coherence literature. Though it

was originally intended to deal with the linguistic problem of anaphora resolution (Beaver, 2004), it finds extensive applications in text generation and text summarization.

2.3 Coherence Modeling in Summarization

As many researchers in text summarization turn their attention to the language quality in the output, coherence plays an increasingly important role. Most of the works are focused on modeling coherence as understood in (text) linguistics. In this section, I will first review models on different levels – global coherence models (2.3.1), local coherence models (2.3.2) and hybrid models of coherence (2.3.3) and then in 2.3.4, I will introduce efforts at coherence evaluation.

2.3.1 Global Coherence Models

Most models of global coherence are based on RST to capture the discourse-level coherence patterns. The extensive use of RST to text summarization is usually credited to Marcu (1997, 1999, 2000). He shows that guided by rhetorical relations between clauses, it is possible to parse a discourse. In (Marcu, 1997), he implements a robust rhetorical parser by a manually built corpus and a rhetorical parsing algorithm. According to (Marcu, 1999), the salience of textual units is determined by the depth of their tree nodes, which in turn is determined by their nucleus/satellite status constrained by the RST

rhetorical relations. Further theoretical proofs and parsing details are provided in (Marcu, 2000).

Despite the success, Marcu's RST tree model is also criticized. For example, Wolf and Gibson (2004, 2006) find fault with the binary tree in RST, which they contend to be inadequate due to its structural constraints. Instead, they advocate a "chain graph structure" that can represent crossed dependencies and multiple-parent nodes and is thus descriptively more adequate than RST trees.

Another criticism is made by Knott et al. (2001), who argue against a problematic rhetorical relation in RST – (object-attribute) elaboration. The authors prove that elaboration is on the entity, instead of proposition, level. Therefore, they propose supplementing RST with entity-based coherence, a contribution that local coherence models can make.

Using a content model based on HMM, Barzilay and Lee (2004) interpret global coherence as a domain-specific topical structure. According to their content model, each HMM state corresponds to a topic from which sentences are generated. In effect, the content model captures the coherence pattern as shift between topic states.

2.3.2 Local Coherence Models

Most models of local coherence are based on CT or the linguistic account of lexical cohesion to capture the relationship between adjacent textual units (usually sentences). As a direct application, CT's constraints and rules (Brennan

et al., 1987) can be used to generate metrics for local coherence. Karamanis and his colleagues (Karamanis, 2001; Karamanis et al., 2004a, 2004b; Karamanis and Mellish, 2005) experiment extensively with various CT-derived metrics for sentence ordering, a subtask of summarization.

Hasler (2004) directly applies the CT's transitions (Continue, Retain, Smooth Shift, Rough Shift) to text summarization. The author undertakes two tasks about text extracts and finds that those transitions are unable to distinguish human extracts from machine extracts. CT's limitation is also discussed by Poesio et al.'s (2004) parametric research, which discovers that many real documents do not follow the CT constraints and rules. The authors observe that CT provides at most an account of entity coherence as part of local coherence. Orăsan (2003) develops a CT-based local coherence algorithm for sentence extraction by using evolutionary programming. Sentences are ranked and selected on the basis of content and context. However, the author shows that CT plays a limited role in producing good summaries.

The idea of **entity coherence**, which is related to CT transitions, gives rise to a wave of new research interests. Barzilay and Lapata (2005, 2008) propose an entity grid model to capture local coherence. Using entity grids, they are able to compute the entity transitions in adjacent sentences with transition-based vectors. Coherence assessment is thus recast as a ranking task. The model is tested with summary coherence evaluation and the results show that a linguistically rich version (including coreference, syntax, and salience) of the model gives the best

performance.

Filippova and Strobe (2007) extend the entity-grid model from coreference to semantic relatedness. They experiment on German newspaper texts and find lowered performance as compared with Barzilay and Lapata's (2005) experiment on English news texts. They find that: 1) coreference information is important; 2) entities are distributed unevenly throughout a text; 3) syntactic information helps little, if not at all.

Another effort to extend the entity-grid model is made by Nahnsen (2009), who resorts to a number of shallow features: group similarity, WordNet relations, temporal orderings, and longer range relations. She finds that "group similarity + WordNet relations + Longer range relations" gives the best performance, though not as good as the "coreference + syntax + salience" in (Barzilay and Lapata, 2008).

In CLASSY, Conroy et al. (2006) rely on lexical overlap to order sentences that achieves local coherence, which instantiates a Traveling Salesman Problem (TSP)-style search method. For the same purpose, Lapata (2003) considers both lexical and syntactic features to calculate local coherence between neighboring sentences using a greedy algorithm.

2.3.3 Hybrid Models of Coherence

Both the global coherence and local coherence models may only reveal some coherence patterns and address some issues involved in text summarization.

As Knott et al. (2001) argue, CT-based local coherence will complement RST-based global coherence in some aspects. That's why many researchers strive to develop hybrid models that combine the strengths of both.

An attempt to integrate lexical cohesion into a global coherence model is made by Alonso i Alemany and Fuentes (2003). They build a hybrid model of text summarization that combines rhetorical relations to account for coherence and lexical chains to account for cohesion.

Soricut and Marcu (2006) develop "utility-trained coherence models" based on HMM. Different from most other hybrid models, their model integrates a number of heterogeneous coherence models, both local ones (word-co-occurrence coherence models and entity-based coherence models) and global ones (HMM-based content models), in a log-linear fashion. Similarly, Elsner et al. (2007) report on a method of coherence-targeted text generation that combines a local coherence model (Barzilay and Lapata, 2005) and a global coherence model (Barzilay and Lee, 2004). The combination gives rise to a probabilistic model that relies on the relaxed entity grid as its local features and a unigram language model on a global (topic-based) level.

Cristea et al. (1998) report a method to combine both a global coherence model (like RST) and a local one (like CT). They establish the Veins Theory (VT), which extends the arguments of CT to text spans beyond adjacent units, thus addressing global coherence. It starts from the RST tree that identifies the global discourse structure with nuclear/satellite nodes and then calculate the vein

of each leaf node representing discourse unit. Following Marcu's (1997) basic idea, VT can be used to summarize a given unit or sub-tree of a text.

If VT is essentially an RST-based CT model in which the local coherence model dominates, Kibble and Power (2004) present a CT-guided RST model in which the global coherence model dominates. Building on the propositional representations and the established RST rhetorical structure of the text, it explores a CT-guided text generation scheme that integrates text planning, sentence planning, and pronominalization.

2.3.4 Coherence Evaluation

Most summarization evaluation methods, intrinsic and extrinsic alike, are content-targeted (see 1.1.3), which makes automatic evaluation possible. Coherence evaluation, on the other hand, is rather subjective, so summary coherence is usually done manually, a practice adopted by DUC/TAC since DUC 2005 (Dang, 2005).

But in the case of sentence ordering – an instantiation of coherence realization – automatic evaluation is possible. Given a gold standard ordering, Kendall's τ (Lapata 2003, 2006) is proven to be the best metric for evaluating alternative orderings. The entity-grid model (Barzilay and Lapata 2005, 2008) also provides a method for automatic evaluation of summary coherence.

Lapata and Barzilay (2005) experiment with various models, including syntactic models and semantic ones, for automatic evaluation of coherence. The

syntactic model is based on (Barzilay and Lapata, 2005) that captures the local coherence with entity transitions. The semantic models do not concern syntactic structure or even word order. The experimental results show that individually, the models that are most highly correlated with human coherence assessment are the entity grid, the LSA (Foltz et al., 1998), and two WordNet-based models (Hirst and St-Onge, 1998; Jiang and Conrath, 1997). Collectively, the combination of the entity grid, word-overlap, LSA, Hirst and St-Onge, and Lesk (1986) models are the best.

Chapter 3: Shallow Content-driven Coherence in Summarization

In computational linguistics, many accounts and models of coherence are driven by shallow content or literal textual features, which are structurally represented by **words** or **sentences**. Word-level coherence is an effect from word cohesion patterns, such as repetition, synonymy (*tornado* and *twister*), hyponymy (*dog* and *animal*), meronymy (*chair* and *furniture*), etc. Sentence-level coherence results from the coherence between words in sentences. The more coherent word pairs in two sentences, the more closely the sentences are related. Placing such sentences in close proximity leads to good coherence.

Based on words and word relations, we can also derive **entities** and **events** that provide more semantic dimensions to the shallow content of a text. As a result, coherent sentences can also be defined as sentences with closely related entities and/or events.

As will be explicated in 3.1, shallow content-driven coherence based on word, entity, or event relations is typically used for information ordering to improve the coherence in the output summaries. Section 3.2 deals with information ordering for single-document summarization, using word and entity relations; 3.3 adopts a more complicated model for a similar task in multi-document summarization, using entity and event relations; 3.4 summarizes work in this chapter.

3.1 Shallow Content-driven Coherence and Information Ordering

As mentioned in 1.1.2, summary generation and especially information ordering is an important component of textual summarization, which is often intended to enhance summary readability. In the case of extractive summarization, for which sentences are usually extracted in their entirety, summarization is almost equivalent to **sentence extraction + sentence ordering**. Sentences are extracted to cover as much important information as possible and then ordered to make the output as coherent as possible. The following 3 sentences are from the DUC 2002 dataset, which are extracted by human annotators to compose a summary for a set of 6 documents (code: “d061”). Repeated words and entities have been highlighted to facilitate the explanation below.

(3.1) *Tropical **Storm Gilbert** formed in the eastern Caribbean and strengthened into a hurricane Saturday night*.

(3.2) ***Gilbert** reached **Jamaica** after skirting southern Puerto Rico, Haiti and the **Dominican Republic**.*

(3.3) *The **storm** killed 19 people in **Jamaica** and five in the **Dominican Republic** before moving west to Mexico.*

(3.4) *Prime Minister Edward Seaga of **Jamaica** said Wednesday the **storm** destroyed an estimated 100,000 of **Jamaica's** 500,000 homes when it throttled*

the island Monday.

On the word level, (3.1) and (3.2) share “Gilbert”; (3.2) and (3.3) share “Jamaica”, “Dominican” and “Republic”; (3.3) and (3.4) share “storm” and “Jamaica”. Placing the four sentences in the order of {(3.1), (3.2), (3.3), (3.4)} results in good coherence by leveraging such word repetitions. A similar account can be made on the entity level, with the named entities “Gilbert” shared by (3.1) and (3.2), “Jamaica”/ “Dominican Republic” shared by (3.2) and (3.3), and “Jamaica” shared by (3.3) and (3.4). Changing the order of the sentences may result in a less coherent passage. For example, {(3.2), (3.1), (3.3), (3.4)} is less coherent because the named entity “Dominican Republic” is not in adjacent sentences. Moreover, the coherence between (3.3) and (3.4) is not only attributed to the repetition of “storm” and “Jamaica” but also to the relationship between two events – storm killing people and storm destroying homes.

Most works on information ordering for summarization operate on the sentence level and most are motivated by shallow content-driven coherence.

Word overlap is the most obvious shallow content information. The sentences can be ordered in such a way that adjacent sentences have the greatest word overlap on average, which is implemented by Conroy et al. (2006). Accounting for word overlap as well as other lexical relations, Barzilay et al. (2002) combine chronological ordering with lexical cohesion information. Lapata (2003) considers both lexical and syntactic features in calculating

coherence between neighboring sentences using a greedy algorithm.

Entity relation is another kind of shallow content information. Inspired by the Centering Theory (Grosz et al., 1995), Barzilay and Lapata (2005, 2008) propose an entity grid model. Syntactic roles played by entities and transitions between these syntactic roles underlie the coherence patterns between sentences. An entity-parsed corpus is used to train a model that prefers the sentence orderings that comply with the optimal entity transition patterns.

In addition to coherence, sentence timestamps and occurrence order in the source text, if available, are simpler criteria that can be followed independently or combined with coherence criteria in ordering, such as Bollegala et al.'s (2006) "agglomerative ordering" approach.

My work described in the following sections complements those above from two aspects: 1) I will apply information ordering to single-document summarization, which is widely believed to be a trivial task; 2) I will use event information, a less explored shallow content unit, to order sentences in multi-document summarization.

3.2 Information Ordering for Single-document Summarization

It is noticeable that most efforts at information ordering for summarization are aimed at multi-document summarization. For single-document summarization, it is usually taken for granted that the original text order suffices

for the ordering of the extracted sentences. Therefore, information ordering for single-document summarization is traditionally considered to be trivial, which is questionable because no theory proves the sufficiency of text order. On some occasions, e.g., a news article adopting the "Wall Street Journal Formula" (Rich and Harper, 2007) where conceptually related sentences are placed at the beginning and the end, sentence conceptual relatedness does not necessarily correlate with spatial proximity and thus selected sentences for a summary may need to be rearranged for better readability.

Therefore I regard information ordering for single-document summarization as an open issue, as it has long been recognized as an actual strategy taken by human summarizers (Jing, 1998; Jing and McKeown, 2000) and acknowledged early in work on sentence ordering for multi-document summarization (Barzilay et al., 2002).

In the following, I will propose an integrated ordering model for single-document summarization based on sentence grouping, which is inspired by human writer's arrangement of sentences to improve the local and global coherence between sentences. Two grouping methods are discussed in 3.2.1 and a greedy ordering algorithm is presented in 3.2.2. Section 3.2.3 shows experimental results that prove the text order for single-document summarization sentence ordering is not optimal.

3.2.1 Grouping-based Ordering

Human writers and summarizers organize sentences by blocks. Sentences within a block are conceptually close to each other and adjacent sentences cohere with each other. Note that "blocks" are sometimes synonymous with "paragraphs" for documents in general. Local coherence is thus realized within blocks. On the other hand, blocks are not randomly ordered. Two blocks are put next to each other if their contents are close enough. So text-level, or global coherence is realized among blocks. Figure 3.1 is an illustration of a block-style outline for a news report on a hurricane.

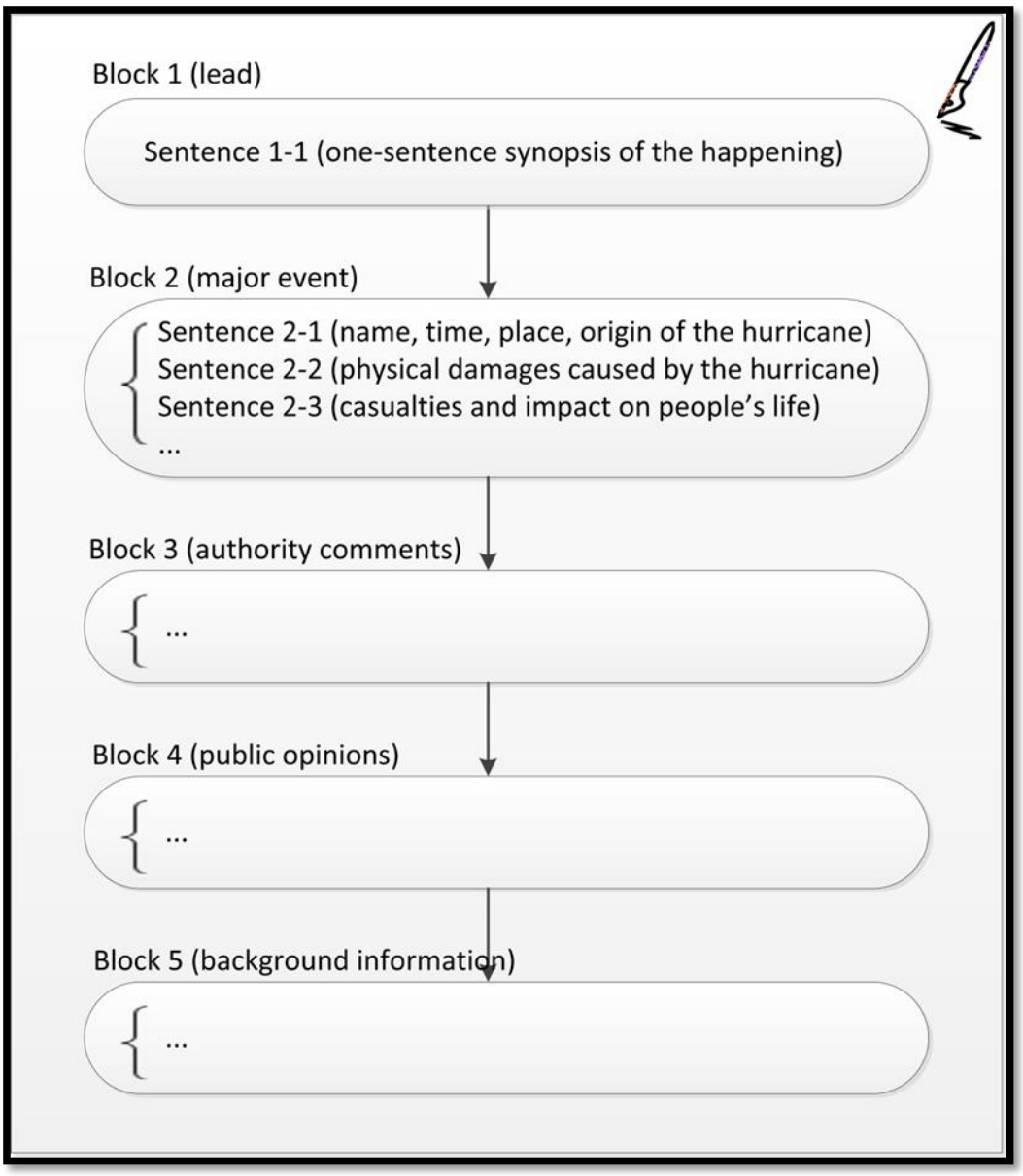

Figure 3.1: Sample Block-Style Writing

My ordering framework is designed to capture both local and global coherence understood in this sense. Globally, we identify related groups among sentences and find their relative order. Locally, we keep sentences similar or related in content close to each other within one group.

As summary sentences are isolated from their original context, I retain the

shallow content information by representing sentences as concept vectors. In the simplest case, the "concept" is equivalent to content word. A drawback of this practice is that it considers every content word equally contributive to the sentence content, which is not always true. For example, in the news domain, entities realized as NPs are more important than other concepts. To represent sentences as entity vectors, I identify both **common entities** (as the head nouns of NPs) and **named entities**. Two common entities are equivalent if their noun stems are identical or synonymous. Named entities are usually equated by identity. But in order to improve accuracy, I also consider: 1) structural subsumption (one is part of another); 2) hypernymy and holonymy (the named entities are in a superordinate-subordinate or part-whole relation).

Now with summary sentence S_i and m entities e_{ik} ($k = 1 \ldots m$), $S_i = (wf(e_{i1}), wf(e_{i2}), \ldots, wf(e_{im}))$, where $wf(e_{ik}) = w_k \times f(e_{ik})$, where $f(e_{ik})$ is the frequency of e_{ik} and w_k is the weight of e_{ik}. We define $w_k = 1$ if e_{ik} is a common entity and $w_k = 2$ if e_{ik} is a named entity. Other things being equal, two sentences sharing a mention of named entities are thematically closer than two sentences sharing a mention of common entities.

To meet the global need of identifying sentence groups, I develop two grouping algorithms by applying a graph-based operation and clustering.

3.2.1.1 Connected Component Finding (CC)

This algorithm treats grouping sentences as finding connected components

(CC) in a text graph $TG = (V, E)$, where V represents the set of sentences and E the sentence relations weighted by cosine similarity. Edges with weight $< t$, a threshold, are removed because they represent poor sentence coherence.

The resultant graph may be disconnected, in which we find all of its connected components, using depth-first search. The connected components are the sentence groups we are looking for. Note that this method cannot guarantee that every two sentences in such a group are directly linked, but it does guarantee that there exists a path between every sentence pair.

3.2.1.2 Modified K-means Clustering (MKM)

Observing that the CC method finds only *coherent groups*, not necessarily *groups of coherent sentences*, I develop a second algorithm using clustering. A good choice is K-means as it is efficient and outperforms agglomerative clustering methods in NLP applications (Steibach et al., 2000), but the difficulty with the conventional K-means is how to decide K, the number of clusters.

My solution is modified K-means (MKM) based on (Wilpon and Rabiner, 1985). Let's denote cluster i by CL_i and cluster similarity by $Sim(CL_i) = \underset{S_{im}, S_{in} \in CL_i}{Min} (Sim(S_{im}, S_{in}))$, where $Sim(S_{im}, S_{in})$ is the cosine similarity of S_{im} and S_{in}. Figure 3.2 illustrates the algorithm of MKM.

CL_1 = all the sentence vectors;

Do the 1-means clustering by assigning all the vectors to CL_1;

While at least 1 cluster has at least 2 sentences and $Min(Sim(CL_i)) < t$, do:

If $Sim(S_m, S_n) = Min(Sim(CL_i))$, create two new centroids as S_m and S_n;

Do the conventional K-means clustering until clusters stabilize;

Figure 3.2: MKM Algorithm

The above algorithm stops iterating when each cluster contains all above-threshold-similarity sentence pairs or only one sentence. Unlike CC, MKM results in more strongly connected groups, or groups of coherent sentences.

3.2.2 Ordering Methods

After the sentences are grouped, ordering is conducted on two levels: group and sentence. Composed of closely related sentences, groups simulate the blocks in Figure 3.1. As is illustrated there, a coherent passage is arranged on two successive stages to realize global and local coherence. Therefore, ordering is done first on the sentence group level and then on the (intra-group) sentence level. A formal representation of our goal is as follows. Assuming D is a set of sentence blocks or groups $\{G_i\}$ (i = 1, 2, …) and each G_i is a set of sentences $\{S_{ij}\}$ (j = 1, 2, …), the goal of ordering is to maximize the following H.

$$H = \sum_{i=2}^{|D|} (Sim(G_{i-1}, G_i) \sum_{j=2}^{|G_i|} Sim(S_{ij-1}, S_{ij}))$$

In my work, $Sim(G_{i-1}, G_i)$ and $Sim(S_{ij-1}, S_{ij})$ denote similarities (*Sim*) between sentence groups (G_i) and intra-group sentences, and both are redefined as the commonly accepted cosine similarity. To maximize *H*, I propose two approaches to both group-level ordering and sentence-level ordering. For group-level ordering,

1) Group order is decided by the group-representing sentence (g_i) order in the text: $g_i \prec g_j \Rightarrow G_i \prec G_j$, where $\prec$ means "precedes".

2) Group order is decided in a greedy fashion in order to maximize the connectedness between adjacent groups, thus enhancing local coherence. Each time a group is selected to achieve maximum similarity with the ordered groups and the first ordered group (G_1) is selected to achieve maximum similarity with all the other groups.

$$G_1 = \arg\max_{G} \sum_{G' \neq G} Sim(G, G'), \text{ and } \quad G_i = \arg\max_{G \in \{\text{unordered groups}\}} \sum_{j=1}^{i-1} Sim(G_j, G) \quad (i > 1)$$

where *Sim*(*G*, *G*') is the average sentence cosine similarity between *G* and *G*'.

Within the ordered groups, sentence-level ordering is aimed to enhance local coherence by placing conceptually close sentences next to each other. Similarly, I propose two approaches.

1) Sentences are arranged by the text order.

2) Sentence order is greedily decided. Similar to the decision of group order, with ordered sentence S_{pi} in group G_p:

$$S_{p1} = \arg\max_{S \in G_p} \sum_{S' \neq S} Sim(S, S'), \text{ and } \quad S_{pi} = \arg\max_{S \in \{\text{unordered sentences in } G_p\}} \sum_{j=1}^{i-1} Sim(S_{pj}, S) \, (i > 1)$$

Note that the text order is used as a common heuristic used by many single-document summarizers.

3.2.3 Experimental Results

Currently, I am not aware of previous work that has empirically compared alternative ways of sentence ordering for single-document summarization. The experimental results reported below may shed some new light on this "trivial" issue.

3.2.3.1 Experimental Design

I prepared 3 datasets of 60 documents each, the first (D400) consisting of documents of about 400 words from the Document Understanding Conference (DUC) 01/02 datasets; the second (D1k) consisting of documents of about 1000 words manually selected from popular English journals such as *The Wall Street Journal*, *The Washington Post*, etc; the third (D2k) consisting of documents of about 2000 words from the DUC 01/02 dataset. Then I generated 100-word summaries for D400 and 200-word summaries for D1k and D2k. Since sentence selection is not our focus, the 180 summaries were all extracts produced by a simple but robust summarizer built on term frequency and sentence position (Aone et al., 1999).

Three human annotators were employed to each provide reference orderings for the 180 summaries and mark paragraph (of at least 2 sentences) boundaries,

which will be used by one of the evaluation metrics described below.

In my implementation of the grouping-based ordering, the CC grouping threshold $t = Avg(Sim(S_m, S_n)) \times c$, the average sentence similarity in a group multiplied by a coefficient empirically decided on separate held-out datasets of 20 documents for each length category. The "group-representing sentence" is the textually earliest sentence in the group. I experimented with both CC and MKM to generate sentence groups and all the proposed methods in 3.2.2 for group-level and sentence-level orderings, resulting in 8 combinations as test orderings, each coded in the format of "Grouping (CC/MKM) / Group ordering (T/G) / Sentence ordering (T/G)", where T and G represent the text order approach and the greedy selection approach respectively. For example, "CC/T/G" means grouping with CC, group ordering with text order, and sentence ordering with the greedy approach.

I evaluate the test orderings against the 3 reference orderings and compute the average result (Madnani et al., 2007) by using 3 different metrics.

The first metric is Kendall's τ (Lapata, 2003, 2006), which has been reliably used in ordering evaluations (Bollegala et al., 2006; Madnani et al., 2007). It measures ordering differences in terms of the number of adjacent sentence inversions necessary to convert a test ordering to the reference ordering.

$$\tau = 1 - \frac{4m}{N(N-1)}$$

In this formula, m represents the number of inversions described above and N is the total number of sentences.

The second metric is the Average Continuity (*AC*) proposed by Bollegala et al. (2006), which captures the intuition that the quality of sentence orderings can be estimated by the number of correctly arranged continuous sentences.

$$AC = exp(1/(k-1)\sum_{n=2}^{k}\log(P_n + \alpha))$$

In this formula, k is the maximum number of continuous sentences, α is a small value to suppress zeroes in case $P_n = 1$. P_n, the proportion of continuous sentences of length n in an ordering, is defined as $m/(N - n + 1)$ where m is the number of continuous sentences of length n in both the test and reference orderings and N is the total number of sentences. Following (Bollegala et al., 2006), we set $k = Min(4, N)$ and $\alpha = 0.01$.

I also go a step further by considering only the continuous sentences in a paragraph marked by human annotators, because paragraphs are local meaning units perceived by human readers and the order of continuous sentences in a paragraph is more important than the order of continuous sentences across paragraph boundaries. So in-paragraph sentence continuity is a better estimation for the quality of sentence orderings. This is my third metric: Paragraph-level Average Continuity (*P-AC*).

$$P\text{-}AC = exp(1/(k-1)\sum_{n=2}^{k}\log(PP_n + \alpha))$$

Here $PP_n = m'/(N - n + 1)$, where m' is the number of continuous sentences of length n in both the test ordering and a paragraph of the reference ordering. All the other parameters are as defined in *AC* and P_n.

3.2.3.2 Results

The following tables show the results measured by each metric. For comparison, we also include a "Baseline" that uses the text order. For each dataset, two-tailed paired t-test is conducted between the top scorer and all the other orderings and statistical significance ($p < 0.05$) is indicated by *.

	τ	***AC***	***P-AC***
Baseline	0.6573*	0.4452*	0.0630
CC/T/T	**0.7286**	**0.5688**	**0.0749**
CC/T/G	0.7149	0.5449	0.0714
CC/G/T	0.7094	0.5449	0.0703
CC/G/G	0.6986	0.5320	0.0689
MKM/T/T	0.6735	0.4670*	0.0685
MKM/T/G	0.6722	0.4452*	0.0674
MKM/G/T	0.6710	0.4452*	0.0660
MKM/G/G	0.6588*	0.4683*	0.0682

Table 3.1: D400 Evaluation

	τ	*AC*	*P-AC*
Baseline	0.3276	0.0867*	0.0428*
CC/T/T	0.3324	0.0979	0.0463*
CC/T/G	0.3276	0.0923	0.0436*
CC/G/T	0.3282	0.0944	0.0479*
CC/G/G	0.3220	0.0893*	0.0428*
MKM/T/T	**0.3390**	**0.1152**	**0.0602**
MKM/T/G	0.3381	0.1130	0.0588
MKM/G/T	0.3375	0.1124	0.0576
MKM/G/G	0.3379	0.1124	0.0581

Table 3.2: D1k Evaluation

	τ	*AC*	*P-AC*
Baseline	0.3125*	0.1622	0.0213
CC/T/T	**0.3389**	**0.1683**	**0.0235**
CC/T/G	0.3281	**0.1683**	0.0229
CC/G/T	0.3274	0.1665	0.0226
CC/G/G	0.3279	0.1672	0.0226
MKM/T/T	0.3125*	0.1634	0.0216
MKM/T/G	0.3125*	0.1628	0.0215
MKM/G/T	0.3125*	0.1630	0.0216
MKM/G/G	0.3122*	0.1628	0.0215

Table 3.3: D2k Evaluation

In general, our grouping-based ordering scheme outperforms the baseline for news articles of various lengths and statistically significant improvement can be observed on each dataset. This result casts serious doubt on the widely accepted practice of taking the text order for single-document summary generation.

The three evaluation metrics produce consistent results although they are based on different observations. The *P-AC* scores are much lower than their *AC* counterparts because of its strict paragraph constraint.

Interestingly, applying the text order posterior to sentence grouping for group-level and sentence-level ordering leads to consistently good performance, as the top scorers on each dataset are almost all "__/T/T". This suggests that the textual realization of coherence can be sought in the source document if possible, after the selected sentences are rearranged. It is in this sense that the general intuition about the text order is justified. It also suggests that tightly knit paragraphs (groups), where the sentences are closely connected, play a crucial role in creating a coherent flow. Shuffling those paragraphs may not affect the final coherence.

The grouping method does make a difference. While CC works best for the short and long datasets (D400 and D2k), MKM is more effective for the medium-sized D1k. Whether the difference is simply due to length or linguistic/stylistic subtleties is an interesting topic for further in-depth study.

3.3 Information Ordering for Multi-document Summarization

If information ordering for single-document summarization is often believed to be trivial, that for multi-document summarization is not so. It is because a multi-document summary probably consists of sentences from different source documents and there is no one “text order” to guide their arrangement in the output summary. Therefore, information ordering for multi-document summarization is nontrivial for most researchers.

Among the works oriented for coherence, the use of word and entity relations as shallow content to compute sentence coherence is widely adopted. It is not surprising that few efforts have been made at levels higher than entity or word in measuring sentence coherence, considering the fact that the traditional Vector Space Model (VSM) underlying many MDS summarization systems represents sentences simply as word occurrence or frequency vectors. Less explored is some higher-level content unit like **event**, which will play an important role in my novel approach to information ordering for multi-document summarization.

The use of event to better capture sentence coherence derives from the observation that what links sentences may be “packages” of words/entities instead of individual words/entities. A good package is event, which relates words/entities in a meaningful way and stretches the possibility of shallow content-based analysis. For example, the appearance of the same word “fire” in

two adjacent sentences does not necessarily make them coherent. But if the first sentence includes “police investigation into a fire case” and the second sentence discusses the “arrest of the person who started the fire”, they are certainly coherent. Intuitively the coherence exists between the “investigation” event and the “arrest” event both involving “fire”.

Event has proved useful in content selection for multi-document summarization (Filatova and Hatzivassiloglou, 2004; Li et al., 2006), but it is rarely used for information ordering. In the following, I will show that event information is instrumental for this task. The overall architecture of my work is shown in Figure 3.3.

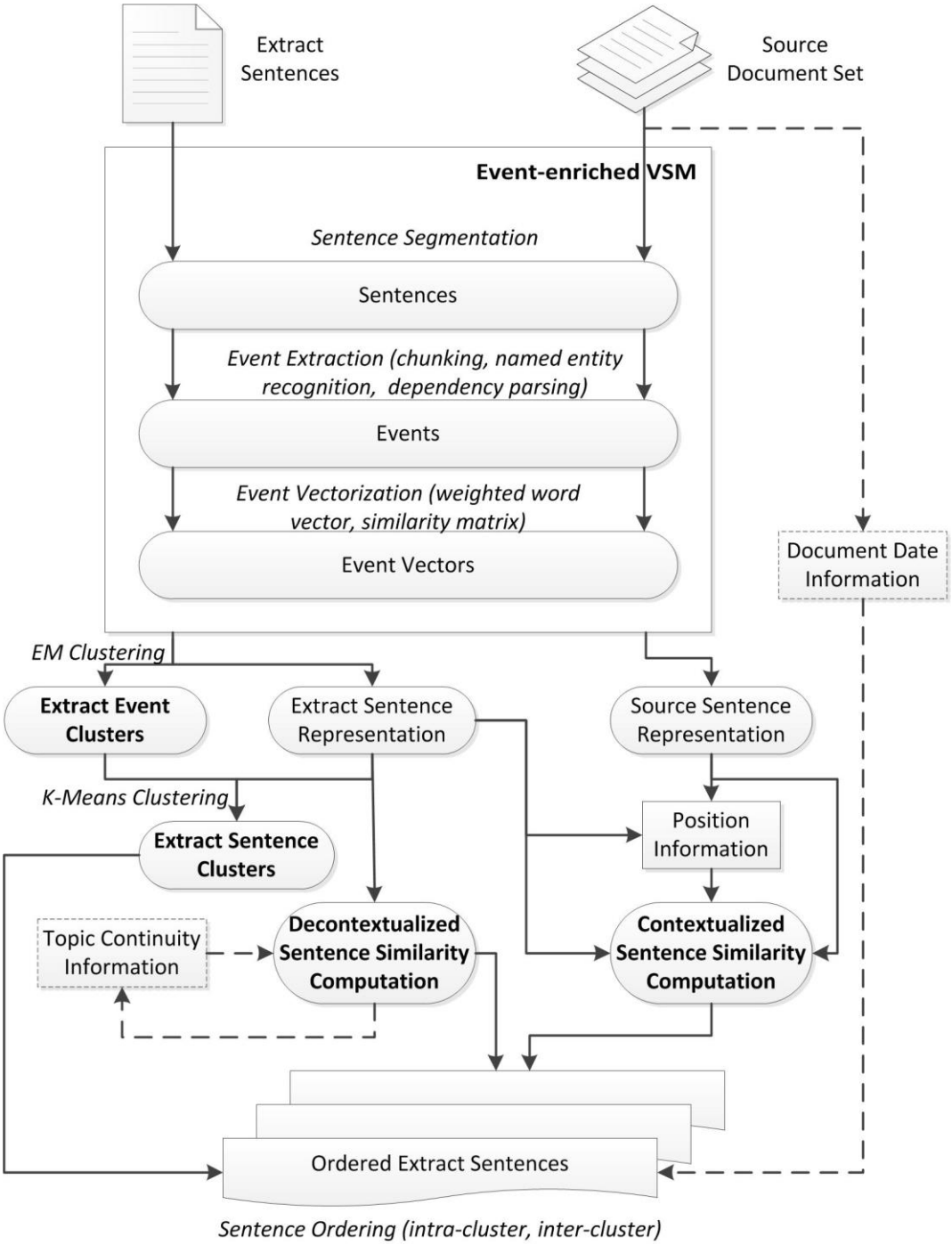

Figure 3.3: Architecture of the Event-driven Ordering Model

Event information is used to construct the "event-enriched VSM" highlighted in Figure 3.3, as the model converts sentences to events and event-structured vectorial representations. The details are provided in 3.3.1. Then

a novel two-layered clustering approach is adopted in 3.3.2 to produce "extract event clusters" and "extract sentence clusters", the latter to be input to a sentence ordering algorithm. The other highlighted parts, "decontextualized sentence similarity computation" and "contextualized sentence similarity computation", lie at the core of the ordering algorithm and will be discussed in 3.3.3. The basic idea about "context" is to process not only an individual sentence, but also a sentence from a specific position in its source document, which explains why "source document set" as well as the to-be-ordered "extract sentences" should be processed by the same model. The dashed-line parts, such as "topic continuity information" and "document date information", are optional components to refine the ordering algorithm. Section 3.3.4 provides evaluation results, including the effects of all the optional components.

3.3.1 Event as Shallow Content Unit

I will first give a more formal definition of event in the following discussion. Conceptually, an event is an occurrence, happening, activity, or episode associated with participants, time, place, and manner. Structurally, an event is a composite shallow content unit that encompasses **event terms** and **event entities**, which are all **event elements**. The event term corresponds to the activity or episode that is central to an event and each event entity denotes a participant, time, place, or manner that constitutes the event. An event E has one event term $Term(E)$ and a set of event entities $Entity(E)$, i. e.,

$$E = [Term(E), Entity(E)]$$

The previous example (3.3), repeated here as (3.5), has one event about storm killing people and the other about storm moving to Mexico. In terms of events, (3.5) can be represented as (3.6).

(3.5) *The storm killed 19 people in Jamaica and five in the Dominican Republic before moving west to Mexico.*

(3.6) {[*killed*, [*storm*, *people*, *Jamaica*, *Dominican Republic*]],

[*moving*, [*storm*, *Jamaica*, *Dominican Republic*, *west*, *Mexico*]]}

(3.7) [*Dominican*, *Mexico*, *Jamaica*, *Republic*, *five*, *kill*, *move*, *people*, *storm*, *west*]

For comparison, I have also shown the word-based representation of (3.5) as (3.7), after stemming and normalization. It is noteworthy that both representations are based on shallow content, but the composite structure of event in (3.6) helps to organize the sentence words in a sensible way and build some semantics into the representation. In the following, I will show how we can extract the events as shown in (3.6) using external resources and shallow content analysis.

3.3.1.1 Event Extraction and Vectorization

Event terms are typically action verbs that denote actions or activities. I

make use of WordNet 3.0's lexical file information and restrict action verbs to the 13 types in Table 3.4.

WordNet File Number	WordNet File Name	Examples
29	verb.body	*cry, breathe*
30	verb.change	*change, intensify*
31	verb.cognition	*analyze, doubt*
32	verb.communication	*ask, order*
33	verb.competition	*compete, combat*
34	verb.consumption	*drink, consume*
35	verb.contact	*clash, hit*
36	verb.creation	*paint, perform*
39	verb.perception	*hear, feel*
38	verb.motion	*fly, swim*
40	verb.possession	*transfer, claim*
41	verb.social	*overthrow, segregate*
44	verb.weather	*rain, thunder*

Table 3.4: Action Verbs in WordNet 3.0

From this set I remove light verbs, which contribute little to the semantic content of phrases they often participate in. Examples are "take" (as in "take a walk"), "make" (as in "make a decision"), and "give" (as in "give a speech")

(Tan et al., 2006). In English, light verbs make up a limited set.

Deverbal nouns are the other category of event terms, which are grammatically nouns but functionally like verbs. Some of the recognized deverbal nouns are also used as verbs (e.g., "damage", "flood"), but some are truly nouns derived from verbs (e.g., "investigation", "establishment"). I also make use of WordNet 3.0's lexical file information and restrict deverbal nouns to the 3 types in Table 3.5.

WordNet File Number	WordNet File Name	Examples
04	noun.act	*action, investigation*
11	noun.event	*upheaval, destruction*
22	noun.process	*corrosion, deposition*

Table 3.5: Deverbal Nouns in WordNet 3.0

Event entities include named entities and common entities. Unlike the triplets (two named entities and one connector) in (Filatova and Hatzivassiloglou, 2003), an event in our model can have an unlimited number of event entities, as is often the real case.

Event extraction begins with sentence segmentation, shallow parsing with the lexical resource WordNet, and named entity recognition (NER) using a trained model provided in NLTK[4] (Bird et al., 2009), analyzing each sentence *S*

[4] The Natural Language Toolkit (NLTK) is a natural language processing module for the Python language. Also see http://www.nltk.org/.

into ordered lists[5] of event terms $\{t_1, t_2, \ldots\}$ with segments among them, as shown in Figure 3.4. If a noun is decided to be an event term, it cannot be (the head noun of) an entity.

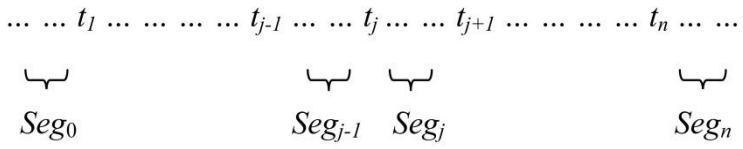

Figure 3.4: Segments among Event Terms

For each t_j in Figure 3.4, the corresponding event E_j is extracted by taking t_j and the ordered event entities in its nearest entity-containing Seg_p and Seg_q, i.e., $E_j = [Et_j, Entity(Seg_p) \cup Entity(Seg_q)]$ where $p = \text{argmax}_{0 \le i \le j-1} Entity(Seg_i) \neq \varnothing$ and $q = \text{argmin}_{j+1 \le i \le n} Entity(Seg_i) \neq \varnothing$ if such p and q exist. This method can produce the result (3.8) for the sentence (3.5). Compared with (3.6), the "moving" event loses the important entity "storm" and has the unnecessary "people".

(3.8) {[*killed*, [*storm*, *people*, *Jamaica*, *Dominican Republic*]],

[*moving*, [*people*, *Jamaica*, *Dominican Republic*, *west*, *Mexico*]]}

This problem can be partly resolved by recognizing the dependency of "storm" on "moving". For this purpose, I use the state-of-the-art Stanford parser

[5] They are not sets because they are ordered and redundant members are allowed. However, I still use set operators for convenience's sake where confusion does not arise.

2.0.1 (Klein and Manning, 2003) to produce dependency information such as:

nsubj(*killed, storm*),

dobj(*killed, people*),

prepc_before(*killed, moving*),

which enables us to attach the missing subject, "storm", to the "moving" event in 3.8) by following the dependency links. For each sentence, I first extract its events by using the simple method described above. Then I use the Stanford parser to obtain all its dependency pairs in the form of *dependency_relation*(*verb, noun*), *dependency_relation*(*noun, noun*), or *dependency_relation*(*verb, verb*). The algorithm in Figure 3.5 shows how we refine event extraction by using the dependency pairs. First all the directly dependent nouns that are not included are added to the event. Then all the indirectly dependent nouns (via a dependency relation between verbs sharing the same subject: *nsubj* or *nsubjpass*) are also added.

Input: sentence events to be refined, dependency pairs

Output: refined sentence events

```
For each sentence event do
    For each dependency pair do
      If dependency pair is dependency_relation(verb, noun) and verb is the
      event term do
          Add new noun to event entities
      Elseif dependency pair is dependency_relation(noun1, noun2) and noun1
      is the event term do
          Add new noun2 to event entities
      Elseif dependency pair is dependency_relation(verb1, verb2) and verb2
      is the event term do
          For all nouns found in nsubj(verb1, noun) and nsubjpass(verb1,
              noun):
              Add new noun to event entities
          Endfor
        Endif
    Endfor
Endfor
```

Figure 3.5: Algorithm for Refining Sentence Event Extraction

In practice, the dependency information is useful in retrieving some subjects, objects, or adverbials that could have been missed. Its limitation is that it only deals with "false negatives", or event entities that fail to be retrieved by the shallow method such as "storm" in the "moving" event, but not "false positives", or event entities that are erroneously retrieved by the shallow method such as "people" in the "moving" event. I believe the false negatives leading to incomplete events are more problematic than the false positives leading to impure events, but will also deal with the latter in future work.

The conversion of events such as those in (3.6) to vectors is similar to sentence vectorization in the traditional VSM. The only subtlety is that unlike bag-of-word sentences, events are structured, with event terms and entities on different conceptual levels. My strategy is to "flatten" such a structure by organizing all corpus-wide event elements into a concatenation of all event terms and all event entities. Given m distinct event terms and n distinct event entities, each event can be converted to an $m+n$-dimension vector with ternary values {0, 1, 2}. For event terms and common event entities, 1 and 0 denote their existence or non-existence. For named entities, non-existence is denoted by 0 but existence by 2. The term-entity flattening is important for constructing a similarity matrix to compute event similarity, as I will explain in the following.

3.3.1.2 Event Similarity

Computing event similarity seems more complicated than computing word

similarity or bag-of-word sentence similarity because the relations between event terms and event entities have to be considered differently. That is why early research tends to treat it as two tasks, computing term (or verb) similarity and entity similarity before selecting one of them or linearly combining both to obtain a score for event similarity. The divide-and-combine approach has two flaws. First, term and entity similarities are computed using different measures and different lexical resources. For example, Liu et al. (2007) use VerbOcean (Chklovski and Pantel, 2004) information to compute term (verb) similarity; Li et al. (2006) use term overlap and WordNet information to compute term and entity similarities; Zhang et al. (2010) use term overlap, VerbOcean, and WordNet information to compute term and entity similarities. But it is not clear how compatible different sources or measures are, which is also associated with the second flaw: deciding a parameter to combine scores from different processes is usually laborious and error-prone.

In the following I will adopt a better method. The basic idea is to multiply the event vector $\vec{E}$ with a similarity matrix W to get a new vector $\vec{E}'$, after a similar technique taken by Stevenson and Greenwood (2005) to compute pattern similarity. With $m+n$ event elements $e_1, \ldots, e_m, e_{m+1}, \ldots, e_{m+n}$ (m terms + n entities), W is an $(m+n)\times(m+n)$ matrix with each w_{ij} denoting the similarity between e_i and e_j. As event terms and entities are situated at different conceptual levels, their similarity is assigned 0. Specifically,

$$w_{ij} = \begin{cases} Sim_{ET}(e_i, e_j) & 1 \le i, j \le m \\ Sim_{EE}(e_i, e_j) & m+1 \le i, j \le m+n \\ 0 & \text{otherwise} \end{cases}$$

WordNet is used to populate *W*. The similarity between two event terms, $Sim_{ET}(e_i, e_j)$, is defined as the maximum Jiang-Conrath similarity (Jiang and Conrath, 1997), $Sim_{JCN}(e_i, e_j)$, between their WordNet senses.

$$Sim_{ET}(e_i, e_j) = Sim_{JCN}(e_i, e_j) = \max_{\substack{s \in senses(e_i), \\ s' \in senses(e_j)}} 1/(IC(s) + IC(s') - 2 \times IC(lcs(s, s')))$$

IC is the Information Content from corpus statistics and *lcs*(*s*, *s*') is the least common subsumer or most specific ancestor node of senses *s* and *s*'. Note that we must transform any deverbal noun to its associated verb (homonym or derived) because WordNet does not support the similarity computation between a verb and a noun. I choose this measure because it proves to be superior to some other WordNet-based measures (Budanitsky and Hirst, 2006).

Computing the similarity between two event entities, $Sim_{EE}(e_i, e_j)$, is slightly more complicated. Because of WordNet's limited coverage of proper nouns, the Jiang-Conrath similarity may not apply to two named entities (and returns 0). Therefore I also compute a word overlap score $Sim_{WO}(e_i, e_j)$ between two named entities as follows.

$$Sim_{WO}(e_i, e_j) = \frac{|Word(e_i) \cap Word(e_j)|}{|Word(e_i) \cup Word(e_j)|}$$

Word(*e*) is the set of all words in *e*. This score captures the surface similarity between two named entities, which is often sufficient for comparing terms with

restricted senses. Then for two named entities e_i and e_j, I take the maximum of $Sim_{WO}(e_i, e_j)$ and $Sim_{JCN}(e_i, e_j)$. If one of them is a common entity, I use $Sim_{JCN}(e_i, e_j)$.

After W is established, computing the similarity between two event vectors $\vec{E}_i$ and $\vec{E}_j$, $Sim_E(\vec{E}_i, \vec{E}_j)$, is simple.

$$Sim_E(\vec{E}_i, \vec{E}_j) = Sim_{COS}(\vec{E}_i W, \vec{E}_j W) = \frac{\vec{E}_i W \cdot \vec{E}_j W}{\left\|\vec{E}_i W\right\| \left\|\vec{E}_j W\right\|}$$

Sim_{COS} is the cosine similarity. It is easy to see that if $W = I$, this similarity measure reduces to the standard cosine.

3.3.2 Two-layered Event and Sentence Clustering

I now move from events to their enclosing sentences. In event-enriched VSM, sentence vectorization based on events is not as straightforward as on entities or terms. If represented as a bag of entities or terms, a sentence can be directly represented as a word (entity or term) vector. But in our model, a sentence is expressed as a set of events and only indirectly related to words, so a direct approach is infeasible. Therefore I propose a novel *two-layered clustering* for sentence vectorization. The basic idea is clustering events at the first layer and then using event clusters as a feature to vectorize and cluster sentences at the second layer. Dimensionality reduction is also used to improve the clustering quality.

3.3.2.1 Event Clustering

A hard clustering of events, such as K-means, will result in binary values in sentence vectors and data sparseness. Because of the internal structure of events, hard clustering of events is also inappropriate. For example, if EC_1 clusters events all with event terms similar to *et** (an event term), and EC_2 clusters events all with event entity sets similar to *es** (an event set), how should event {*et**, *es**} be clustered? Assigning it to either EC_1 or EC_2 is inappropriate as it is partially similar to both. For those reasons, I believe soft clustering is more appropriate.

A well-studied soft clustering technique is the Expectation-Maximization (EM) algorithm which finds maximum likelihood estimates of hidden variables in a statistical model by iteratively computing the expectation of the log-likelihood by using the currently estimated variables (E-step) and computing new parameters that maximize the expected log-likelihood (M-step). Let's assume a Gaussian mixture model for the q event vectors $\boldsymbol{V}_1$, $\boldsymbol{V}_2$, …, $\boldsymbol{V}_q$, with hidden variables $\boldsymbol{H}_i$, initial means $\boldsymbol{M}_i$, priors $\boldsymbol{\pi}_i$, and covariance matrix $\boldsymbol{C}_i$. In the E-step, to compute the expectation of log-likelihood, we calculate the hidden variables $\boldsymbol{H}_i^t$ as the conditional distribution of each $\boldsymbol{V}_t$.

$$\boldsymbol{H}_i^t = \frac{\boldsymbol{\pi}_i \left(\boldsymbol{C}_i^{det}\right)^{-1/2} \exp(-\frac{1}{2} * \left(\boldsymbol{V}_t - \boldsymbol{M}_i\right)^T \boldsymbol{C}_i^{-1} (\boldsymbol{V}_t - \boldsymbol{M}_i))}{\sum_j \boldsymbol{\pi}_j \left(\boldsymbol{C}_j^{det}\right)^{-1/2} \exp(-\frac{1}{2} * \left(\boldsymbol{V}_t - \boldsymbol{M}_j\right)^T \boldsymbol{C}_j^{-1} (\boldsymbol{V}_t - \boldsymbol{M}_j))}$$

where $\boldsymbol{C}_i^{det}$, $\left(\boldsymbol{V}_t - \boldsymbol{M}_i\right)^T$, $\boldsymbol{C}_i^{-1}$ denote the determinant of $\boldsymbol{C}_i$, transpose of $\boldsymbol{V}_t - \boldsymbol{M}_i$, and inverse of $\boldsymbol{C}_i$.

The M-step re-estimates the new priors, means, and covariance matrix in

order to maximize the log-likelihood.

$$\boldsymbol{M}_i^{'} = \frac{\sum_t \boldsymbol{H}_i^t \boldsymbol{V}_t}{\sum_t \boldsymbol{H}_i^t}, \quad \boldsymbol{\pi}_i^{'} = \frac{\sum_t \boldsymbol{H}_i^t}{q}, \quad \boldsymbol{C}_i^{'} = \frac{\sum_t \boldsymbol{H}_i^t (\boldsymbol{V}_t - \boldsymbol{M}_i^{'})(\boldsymbol{V}_t - \boldsymbol{M}_i^{'})^T}{\sum_t \boldsymbol{H}_i^t}$$

The two steps are iterated until the log-likelihood ***LL*** converges within a threshold = 10^{-6}.

$$\boldsymbol{LL} = \log(\sum_t \sum_j \boldsymbol{\pi}_j \left(\boldsymbol{C}_j^{det}\right)^{-1/2} \exp(-\frac{1}{2} * \left(\boldsymbol{V}_t - \boldsymbol{M}_j\right)^T \boldsymbol{C}_j^{-1} \left(\boldsymbol{V}_t - \boldsymbol{M}_j\right)))$$

The performance of the EM algorithm is sensitive to the initial means, which are pre-computed by the standard K-means algorithm repeated 100 times with random initial means for optimal output and the number of clusters *K* is empirically determined to be $\sqrt{q}$. The initial priors are set to be (1.0, 1.0, …, 1.0) and the initial covariance matrix is an identity matrix of dimension *n*.

3.3.2.2 Sentence Clustering

An outcome of the EM clustering of events is that each sentence event is assigned a probability distribution over all event clusters. Next, I vectorize a sentence by summing up the probabilities of its constituent event vectors over all event clusters (*EC*s) and obtaining an *EC*-by-sentence (S_u) matrix $\boldsymbol{S} = [s_{ij}]$, as shown in Figure 3.6, where $s_{ij} = \sum_{E_r \in S_j} P(\left(\overrightarrow{E_r} | EC_i\right))$ and $\overrightarrow{E_r}$ is the corresponding vector of event E_r.

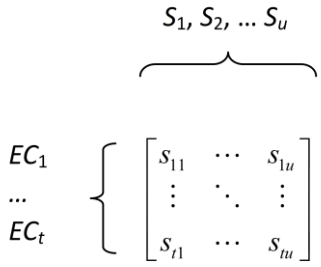

Figure 3.6: *EC*-by-Sentence Matrix

At the sentence layer, hard clustering is sufficient because we need definitive, not probabilistic, membership information for the next step – sentence ordering. For this purpose I use the K-means algorithm and *K* is empirically determined to be $\sqrt{u}$.

3.3.2.3 Dimensionality Reduction

With high dimensionality (in the hundreds) and the similarity matrix method that renders 1/3 to 2/3 of them zeroes, the event vectors demonstrate pronounced sparseness, which may affect the quality of clustering. A solution to this problem in an effort to leverage the latent "event topics" among the event elements is the Latent Semantic Analysis (LSA, Landauer and Dumais, 1997), which finds a low-rank approximation to an original matrix by doing Singular Value Decomposition (SVD). I apply LSA-style dimensionality reduction to the event element-by-event matrix $\boldsymbol{E}$ by doing SVD.

$\boldsymbol{E} = \boldsymbol{A}\,\boldsymbol{\Sigma}\,\boldsymbol{B}^T$

$\boldsymbol{A}$ is the $(m+n)\times(m+n)$ event element matrix consisting of the orthogonal

eigenvectors of $\boldsymbol{EE}^T$; $\boldsymbol{B}$ is the $q \times q$ event matrix consisting of the orthogonal eigenvectors of $\boldsymbol{E}^T\boldsymbol{E}$; $\boldsymbol{\Sigma}$ is a $(m+n) \times q$ matrix with singular values of $\boldsymbol{E}$ in descending order on its diagonal and zeroes elsewhere.

Next I obtain $\boldsymbol{\Sigma_h}$ by keeping the *h* largest singular values of $\boldsymbol{\Sigma}$ and transform $\boldsymbol{E}$ to $\boldsymbol{E_h}$, which contains more compact information about the event elements, before applying the EM clustering to $\boldsymbol{E_h}$.

$$\boldsymbol{E_h} = \boldsymbol{A}\,\boldsymbol{\Sigma_h}\,\boldsymbol{B}^T$$

A problem is the selection of *h*, which affects the performance of dimensionality reduction. In this work, I adopt a utility-based metric to find the best *h** that maximizes intra-cluster similarity and minimizes inter-cluster similarity. For that purpose, I use a probability-adapted version of the attested Davies-Bouldin index (*DB*, Davies and Bouldin, 1979).

$$DB(h) = \frac{1}{n}\sum_{i=1}^{n} \max_{j \neq i}\left(\frac{\phi_i^h + \phi_j^h}{\varphi^h(C_i^h, C_j^h)}\right)$$

ϕ_i^h is the average cosine distance (*COS*) of all vectors *v* in event cluster EC_i^h to their cluster centroid C_i^h, relative to *h*-dimensionality reduction. Note that as a result of the EM clustering, *v* belongs to EC_i^h because of a large probability not necessarily equal to 1. Therefore I also take that probability $P(v \mid EC_i^h)$ into account, as well as the centroid probability $P(C_i^h \mid EC_i^h)$, defined as the mean of vector probabilities.

$$\phi_i^h = \frac{\sum_{v \in EC_i^h} COS(v, C_i^h) * P(v \mid EC_i^h)}{\left|EC_i^h\right|} P(C_i^h \mid EC_i^h) \quad ,$$

$$P(C_i^h \mid EC_i^h) = \frac{\sum_{v \in EC_i^h} P(v \mid EC_i^h)}{\left|EC_i^h\right|}$$

$\varphi^h(C_i^h, C_j^h)$ is the cosine distance of event cluster centroids C_i^h and C_j^h, relative to *h*-dimensionality reduction. Centroid probabilities are considered.

$$\varphi^h(C_i^h, C_j^h) = COS(C_i^h, C_j^h) * P(C_i^h \mid EC_i^h) * P(C_j^h \mid EC_j^h)$$

Since *DB*(*h*) is a distance measure, a smaller value corresponds to better clustering quality. So our goal is to find the optimal *h** that minimizes it.

$$h^* = \arg\min_h DB(h)$$

The same dimensionality reduction applies to sentence clustering as possible performance gain is expected from the discovery of latent *EC* "topics". The best dimensionality is determined in a way as described above. But in the case of hard clustering, no probabilities are needed.

3.3.3 Cluster-based Sentence Ordering

As explained in my work on sentence ordering for single-document summarization (3.2.1 and 3.2.2), sentence ordering is guided by the block-style writing (Figure 3.1) addressing both global and local coherence. Our goal is to maximize a function *H* that takes into account both sentence cluster (block) similarity and intra-cluster sentence similarity.

Since the extracted sentences do not necessarily come from the same source document, the computation of sentence similarity should not only be done between extract sentences but also between extract sentences and their original

contexts to accommodate the source difference. It amounts to the distinction between *decontextualized* and *contextualized* sentence similarity computation to be discussed in the following.

3.3.3.1 Decontextualized and Contextualized Sentence Similarity

With regard to extract sentences S_i and S_j, I compute two types of similarity in order to quantify the degree they can stand next to each other in a linear text. The first is to only look at the sentences by themselves, or to treat them as isolated and decontextualized. The decontextualized sentence similarity $Sim_{-C}(S_i, S_j)$ is defined as the maximum event similarity between their events.

$$Sim_{-C}(S_i, S_j) = \max_{\substack{e \in Event(S_i) \\ e' \in Event(S_j)}} Sim_E(\vec{e}, \vec{e'})$$

Event(*S*) is the set of events contained in *S*. This measure suffices for truly decontextualized sentences, but the fact is that the two extract sentences do not come from nowhere. If we are to decide how well S_2 succeeds S_1 in the new extract context, we should also seek clues from their source context, which is inspired by the "sentence precedence" by Okazaki et al. (2004). Reasonably, the relative sentence positions in the source document should be coherence-optimized. That is why I also define contextualized sentence similarity $Sim_{+C}(S_i, S_j)$, which measures to what degree S_i and S_j resemble each other's relevant source context. More formally, let $LC(S_i)$ and $RC(S_i)$ be the left source context and right source context of S_i respectively and suppose S_i and S_j are to be arranged in that order in the new extract,

$$Sim_{+C}\left(S_i, S_j\right) = \frac{Sim_{-C}\left(S_i, LC\left(S_j\right)\right) + Sim_{-C}\left(S_j, RC\left(S_i\right)\right)}{\left|LC\left(S_j\right)\right| + \left|RC\left(S_i\right)\right|}$$

In this work, I simply take $LC(S_i)$ and $RC(S_i)$ to be the left adjacent sentence and right adjacent sentence of S_i in the source document, but expanding the context window to more than one sentence is also feasible. As the denominator, the total number of sentences in $LC(S_j)$ and $RC(S_i)$ is used for normalization, which is necessary even if both are one sentence long because either may be empty.

The final score for the similarity of S_i and S_j, $Sim_S(S_i, S_j)$, is the product of $Sim_{-C}(S_i, S_j)$ and $Sim_{+C}(S_i, S_j)$.

$$Sim_S\left(S_i, S_j\right) = Sim_{-C}\left(S_i, S_j\right) \times Sim_{+C}\left(S_i, S_j\right)$$

3.3.3.2 Intra-cluster and Inter-cluster Ordering

The ordering algorithm is composed of intra-clustering ordering and inter-cluster ordering, analogous to sentence-level ordering and group-level ordering in 3.2.2. But first, we have to decide the leading sentence to start an extract *P*.

Let's define the document-leading extract sentence set L_{Doc} to be the set of all the extract sentences that appear earliest in a document that contributes to the extract, and the time-leading sentence set L_{Time} to be the set of all the extract sentences in documents that have the earliest publication date. Using the heuristic of time and textual precedence, I first generate a set of possible leading sentences $L = \{L_i\}$ as the intersection of L_{Doc} and L_{Time}. Note that $|L_{Doc}|$ = the number of

documents in the document set that are used in the extract, L_{Time} is in fact a sentence collection of time-leading documents used in the extract, and $L_{Doc} \cap L_{Time} \neq \emptyset$.

If L is a singleton, finding the leading sentence S_L is trivial. If not (when more than one document are published on the same earliest date), S_L is decided to be the sentence in L most similar to all the other sentences in the extract so that it qualifies as a good topic sentence.

$$S_L = \arg\max_{L_i \in L} \sum_{L' \in P \setminus \{L_i\}} Sim_S(L_i, L')$$

After the leading sentence is determined, the leading cluster is the one it belongs to. Intra-clustering ordering now starts with this cluster. I adopt a greedy algorithm similar to that used in 3.2.2, which selects a sentence from the unordered sentence set that best coheres with the sentence just ordered. After all the sentences in the current sentence cluster are ordered, I select the next sentence cluster and do the intra-cluster ordering again. This process is iterated until all the sentences in the extract are ordered. Now the remaining question is: how do we determine the next best sentence cluster?

Let's consider the similarity of sentence clusters. Given a processed sentence cluster SC_i, the next best sentence cluster SC_{i+1} among candidate SC_j's is the one that maximizes the cluster similarity $Sim_{CLU}(SC_i, SC_j)$. Since clusters are collections of sentences, their similarity should be measured in terms of all cross-cluster sentence similarities. That is,

$$Sim_{CLU}\left(SC_i, SC_j\right) = \frac{\sum_{S \in SC_i, S' \in SC_j} Sim_S(S, S')}{\left|SC_i \times SC_j\right|}$$

Based on this, the following equation summarizes the inter-cluster ordering method among the set of all clusters *U*.

$$SC_{i+1} = \arg\max_{SC_j \in U \setminus \{SC_i\}} Sim_{CLU}\left(SC_i, SC_j\right)$$

Starting from the second chosen sentence cluster, I choose the first sentence in the current cluster with reference to the last sentence in the previous ordered cluster and apply inter-cluster ordering. The process is iterated until all the extract sentences are in place.

3.3.3.3 Coherence-Enhancing Factors

Drawing on previous works on ordering for summarization and linguistic accounts of coherence, I also consider factors that can possibly enhance coherence: **topic continuity** and **document date** information.

According to the event structure (3.3.1), all entity information is included in events and entity-based topic continuity possibly affects overall textual coherence. To better capture the transition between entities and the flow of topic, I consider a topic-continuity score $tc(S_i, S_j)$ in the spirit of the Centering Theory (CT). According to CT, a list of forward-looking centers (*CF*) can be created from the entities in a sentence. The *CF*s are ranked by their grammatical relations (Subject – Object – Other) and the highest ranked *CF* is called the preferred center (*CP*). One of the *CF*s is a backward-looking center (*CB*), which is the

realized highest-ranked entity from the previous sentence. If the topic continuity and transition are measured in terms of entity change, local coherence can be captured by the relations of *CB* and *CP* in adjacent sentences.

Table 3.6 is adapted from (Taboada and Wiesemann, 2010), which lists all the possible centering transitions and the corresponding $tc(S_m, S_{m+1})$. CB_m is the backward-looking center of S_m; CB_{m+1} and CP_{m+1} are the backward-looking center and preferred center of S_{m+1}. All of the transition types are borrowed from (Taboada and Wiesemann, 2010) except for HALF-ESTABLISH, which fails to be recognized by them.

<table>
<tr><th></th><th colspan="2">$CB_m = \emptyset$</th><th colspan="3">$CB_m \neq \emptyset$</th></tr>
<tr><td>$CB_{m+1} = \emptyset$</td><td colspan="2">NULL
$tc(S_m, S_{m+1}) = 0$</td><td colspan="3">ZERO
$tc(S_m, S_{m+1}) = 0$</td></tr>
<tr><td rowspan="3">$CB_{m+1} \neq \emptyset$</td><td rowspan="2">$CB_{m+1} = CP_{m+1}$</td><td rowspan="2">ESTABLISH
$tc(S_m, S_{m+1}) = 0.2$</td><td rowspan="2">$CB_{m+1} = CP_{m+1}$</td><td>$CB_{m+1} = CB_m$</td><td>CONTINUE
$tc(S_m, S_{m+1}) = 0.2$</td></tr>
<tr><td>$CB_{m+1} \neq CB_m$</td><td>SMOOTH SHIFT
$tc(S_m, S_{m+1}) = 0.1$</td></tr>
<tr><td>$CB_{m+1} \neq P_{m+1}$</td><td>HALF-ESTABLISH
$tc(S_m, S_{m+1}) = 0.1$</td><td>$CB_{m+1} \neq CP_{m+1}$</td><td>$CB_{m+1} = CB_m$</td><td>RETAIN
$tc(S_m, S_{m+1}) =$</td></tr>
</table>

					0.1
				$CB_{m+1} \neq CB_m$	ROUGH SHIFT $tc(S_m, S_{m+1}) = 0$

Table 3.6: Centering Transitions and Topic-Continuity Scores

To implement the CT-based measure, I make two simplifications. First, "realization" of *CB* is taken to be entity repetition. Discourse anaphors (pronouns referring to out-of-sentence entities) are not restored to their co-referring NPs. Second, I approximate the *CF* list with an event entity list ranked by the text order, which does not necessarily agree with the Subject – Object – Other (S-O-X) precedence (Strube and Hahn, 1999). To ensure that this is not a serious problem for our task, I manually checked the grammatical structures of all the extract sentences in the experimental dataset and judged to what degree the text order of event entities matches the S-O-X precedence. Table 3.7 shows the result.

Sentences with event entity order matching S-O-X precedence	Sentences without discourse anaphors	Sentences with events	Total Sentences
503 (77.4%, 73.6%)	635 (92.8%)	650 (95.2%)	683

Table 3.7: Dataset Sentence Statistics

The statistics show that for 73.6% of all the extract sentences, the above approximation is accurate. If we discount the sentences without events, the proportion reaches up to 77.4%. I also counted the sentences without discourse anaphors and found 92.8% of all the sentences extracted by humans fit into this category. Nevertheless, deep parsing and coreference resolution are expected to improve the identification of centering transitions.

Returning to the computation of sentence similarity, the topic-continuity score is used as a bonus score to the decontextualized similarity because topic continuity only applies to the newly constructed extract.

$$Sim_{-C}(S_i, S_j)' = Sim_{-C}(S_i, S_j) \times (1 + tc(S_i, S_j))$$

As my experimental dataset consists of news articles, the second coherence enhancer is based on the widely accepted chronological order (Barzilay et al., 2002). Intuitively, a sentence from a news article published earlier should precede one from a news article published later because the former is probably about earlier events. Therefore I use a time penalty, $tp(S_i, S_j)$, to discount the score. Suppose S_j is ordered after S_i,

$$tp(S_i, S_j) = \begin{cases} 0.8 & S_i\text{'s document date is later than } S_j\text{'s document date} \\ 1 & \text{otherwise} \end{cases}$$

The following equation shows the coherence-enhanced method to compute sentence similarity.

$$Sim_S(S_i, S_j)' = Sim_{-C}(S_i, S_j)' \times Sim_{+C}(S_i, S_j) \times tp(S_i, S_j)$$

$$= Sim_{-C}(S_i, S_j) \times Sim_{+C}(S_i, S_j) \times (1 + tc(S_i, S_j)) \times tp(S_i, S_j)$$

Note that those coherence-enhancing factors are only optional to the ordering algorithm and their functions will be evaluated in the experiments.

3.3.4 Experimental Results

In this section, I report the experimental results of applying the event-enriched VSM and ordering algorithm to the DUC 02 dataset consisting of newswire articles, both objectively and subjectively.

3.3.4.1 Automatic Evaluation

I use the dataset of the DUC 02 multi-document summarization task because model (human) extracts are available. For each document set, 2 model extracts are provided each for the 200-word and 400-word length categories. I use 1 randomly chosen model extract per document set per length category.

I intended to use all 59 document sets of DUC 02 but found that for some length categories, the two provided model extracts contain material not from the news body, but from other sections such as “title”, “lead”, or even “byline”. They are incompatible with our design tailored for news body extracts and therefore I use only those extracts with all sentences selected from the <TEXT><\TEXT> sections of the XML files. As a result, I collect 42 200-word extracts and 39 400-word extracts, which make up the experimental dataset.

- **Experimental Design**

I intend to evaluate the validity of event coherence-based ordering as against entity-based ordering and traditional bag-of-words (BOW) ordering as well as the role played by performance boosters, including topic continuity, time penalty, and LSA-style dimensionality reduction. Therefore I produce 3 sets of 4 peer orderings (different orderings of the same extracted sentences) based on event coherence and entity coherence respectively. Each set consists of a version with all the three performance boosters (EventAll, EntityAll, BOW_All) and three versions corresponding to the absence of one of the performance boosters (EventNoTC, …, EntityNoTC, …, BOW_NoTC, …). For the entity coherence-based orderings, sentences are converted to entity vectors before being multiplied by an entity-only similarity matrix. For the BOW orderings, sentences are directly converted to word vectors using all non-stopwords. For both entity coherence-based and BOW orderings, sentence clustering is done by one-layered K-means based on cosine distance and the ordering details are the same as event coherence-based orderings.

In addition, I use a random ordering and a baseline ordering. The baseline only uses chronological and text order. Extracted sentences are first ordered by the publication date of their source documents and sentences from the same documents are then textually ordered. Table 3.8 lists the 14 peer orderings to be evaluated.

1	Random
2	Baseline (time order + textual order)
3	EventAll (event coherence-based, using all three performance boosters)
4	EventNoTC (event coherence-based, using all but topic continuity)
5	EventNoTP (event coherence-based, using all but time penalty)
6	EventNoLSA (event coherence-based, using all but dimensionality reduction)
7	EntityAll (entity coherence-based, using all three performance boosters)
8	EntityNoTC (entity coherence-based, using all but topic continuity)
9	EntityNoTP (entity coherence-based, using all but time penalty)
10	EntityNoLSA (entity coherence-based, using all but dimensionality reduction)
11	BOW_All (BOW-based, using all three performance boosters)
12	BOW_NoTC (BOW-based, using all but topic continuity
13	BOW_NoTP (BOW-based, using all but time penalty)
14	BOW_NoLSA (BOW-based, using all but dimensionality reduction)

Table 3.8: The Peer Orderings

The evaluation metrics are similar to the ones used for single-document summarization ordering (3.2.3). But since the human extracts do not contain multiple paragraphs, only Kendall's τ and Average Continuity (*AC*) are used.

- **Result**

For each of the peer orderings, I calculate its average τ and *AC* scores for a length category. I also test the statistical significance between the top scorer in each length/metric category (boldfaced in Table 3.9) and all the other versions in the same category, marked by * ($p < .05$) and ** ($p < .01$) on a two-tailed t-test.

	200w		400w	
	Kendall's τ	*AC*	Kendall's τ	*AC*
Random	0.014**	0.009**	–0.019**	0.004**
Baseline	0.387*	0.151*	0.259**	0.151*
EventAll	**0.429**	0.227	**0.416**	**0.235**
EventNoTC	0.391*	0.171*	0.347*	0.189*
EventNoTP	0.425	**0.230**	0.383*	0.227
EventNoLSA	0.388*	0.175*	0.363*	0.170*
EntityAll	0.405*	0.221	0.399*	0.206*
EntityNoTC	0.389*	0.160*	0.341*	0.182*
EntityNoTP	0.410	0.197*	0.377*	0.207*
EntityNoLSA	0.385*	0.170*	0.359*	0.169*
BOW_All	0.407*	0.199	0.391*	0.201*
BOW_NoTC	0.386*	0.158*	0.332*	0.177*
BOW_NoTP	0.402*	0.214	0.374*	0.204*
BOW_NoLSA	0.382*	0.152*	0.348*	0.165*

Table 3.9: Kendall's τ and *AC* for All the Peer Orderings

Nearly all versions of coherence-based orderings, whether BOW, entity or event, outperform the baseline that only considers time and text order, showing that content coherence is an important guidance for human extract generation. In addition, all event versions significantly outperform their entity and BOW counterparts. It clearly shows that events are high-level content units that incorporate all of the document-level entities. Ordering on event information thus subsumes ordering on entity information and the extra information introduced by event structure leads to better result. The improvement of the event versions over their BOW counterparts demonstrates that enriching the traditional VSM with event semantics leads to improvement on the quality of output summary.

Among the three performance enhancers, the LSA-style dimensionality reduction and topic continuity are more useful than time penalty. For dimensionality reduction applied to event coherence ordering, its absence lowers the performance up to 27.7% in the case of 400W/*AC*. The use of topic continuity is also profitable because the centering transition effectively captures the coherence pattern between adjacent sentences. Without it, the performance degrades by as much as 27.6% in the case of 200W/*AC* of EntityAll vs. EntityNoCT. My explanation is that the quality of entity coherence orderings is more sensitive to the entity-based topic continuity. This result is generally consistent with many other CT-inspired works (e.g., Barzilay and Lapata, 2008). What is at issue is the effect of time information. Introducing this factor does not always enhance performance and sometimes lowers it, so that the top scorer in

the 200W/*AC* category is EventNoTP instead of EventAll. There are two possible accounts. First, document time often deviates from sentence time as a sentence in an early document is not necessarily about early events. Performance will be harmed if such deviation introduces much noise. Second, the time effect is proportional to the size of extract as removing it hurts longer extracts more than short extracts. Therefore chronological clues are more valuable for ordering more sentences.

The ordering algorithm achieves better result with long extracts than with short extracts. Understandably, the importance of order and coherence grows with text length.

3.3.4.2 Human Rating

Using the same dataset, I also recruited human judges in a coherence rating task in order to measure how different orderings lead to different degrees of textual coherence from the human perspective.

- **Experimental Design**

To explore whether the ordering algorithm is sensitive to the extraction method, this time sentences are extracted automatically. I build a simple summarizer based on SumBasic (Nenkova and Vanderwende, 2005) with word position information (Ouyang et al., 2010) and produce a 400-word extract for each of the document sets. This time, all the 59 document sets are used since I can ensure all the extracted sentences are from the news body.

After the extracts are generated, I asked a human annotator, a native speaker of English, to order a randomly shuffled collection of extracted sentences for each document set.

Because human rating is highly labor-intensive, I controlled the size of test sets by using 5 ordering versions for each document set: one baseline (based on time and text order), one human ordering, one event coherence-based ordering, one entity coherence-based ordering, and one BOW-based ordering. The last three orderings use all the three performance enhancers.

Three human judges were employed to rate the different orderings according to their degree of coherence. I asked each of them to rate the 5 orderings for each of the 59 document sets. None of the judges was the annotator and all of them are native English speakers with teaching experience in English writing. Their teaching experience, which involves comparing and grading student works similar to the design of this task, contributes to the reliability of the test result.

Following (Barzilay et al., 2002) and (Bollegala et al., 2006), I instructed the judges to rank the orderings for each set as having *low*, *medium*, or *high* coherence, along a scale from being least coherent to most coherent. The orderings were randomly organized in each of the 59 groups so that the judges could not detect any pattern. The judges were also instructed to pay attention to only textual coherence and ignore any other problem with spelling, punctuation, grammar, style, etc. Some coherence rating samples were provided as warm-up.

- **Results**

In Figures 3.7 to 3.9, I show the results of human rating by each of the judges (A, B, C) for the same set of all the orderings.

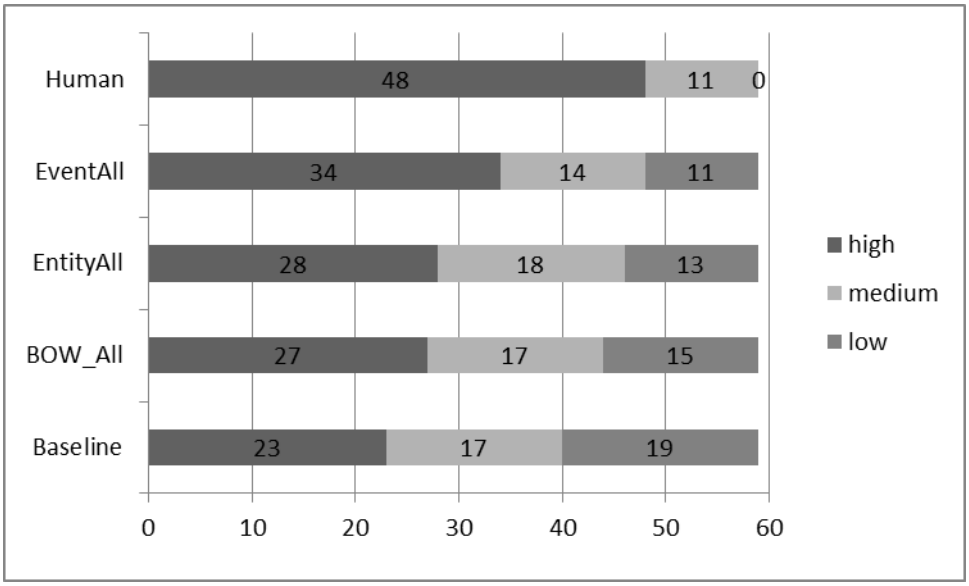

Figure 3.7: Judge A's Rating of the Orderings

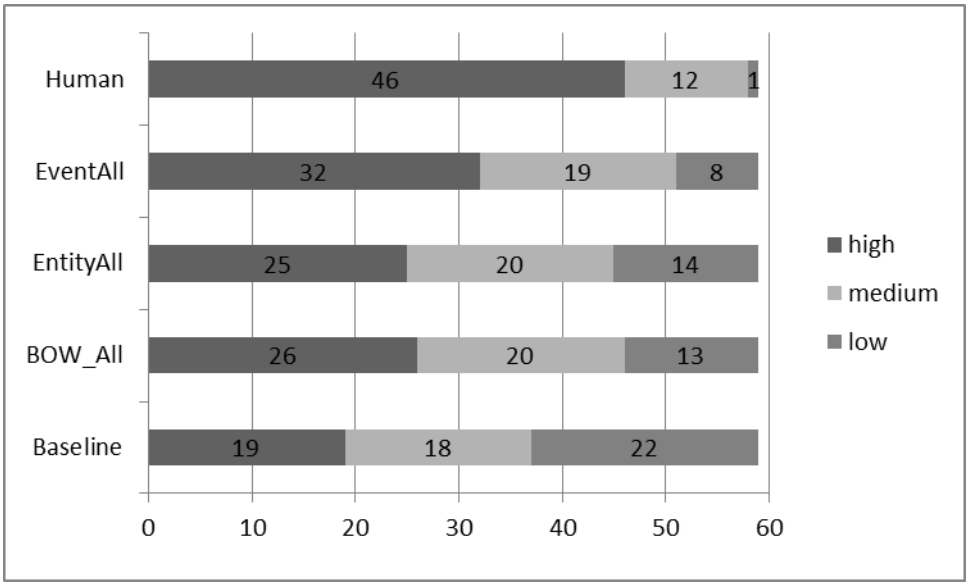

Figure 3.8: Judge B's Rating of the Orderings

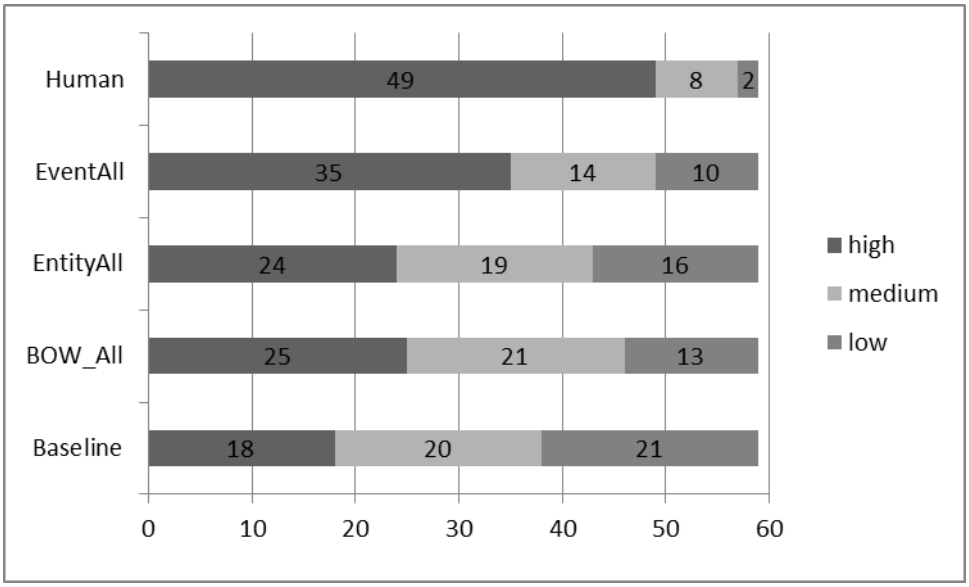

Figure 3.9: Judge C's Rating of the Orderings

I first assess inter-judge agreement by calculating Kendall's *W*, which ranges from 0 (indicating no agreement among the judges) to 1 (indicating total agreement among them). In our case, Kendall's $W = 0.889$, indicating high agreement. Table 3.10 shows the aggregate rating percentages of all types of ordering.

	High	**Medium**	**Low**
Human	80.8%	17.5%	1.7%
EventAll	57.1%	26.6%	16.4%
EntityAll	43.5%	32.2%	24.3%
BOW_All	44.0%	32.8%	23.2%
Baseline	33.9%	31.1%	35.0%

Table 3.10: Aggregate Rating Percentages

Overall, an obvious gap still exists between human orderings and automatic orderings, but nearly 60% of the event coherence-based orderings achieve high coherence, which is quite encouraging. By comparison, entity-based orderings produce 13% less high-coherence orderings, but 5% and 8% more medium-coherence and low-coherence orderings. The BOW orderings perform similarly, which is consistent with the results in Table 3.9. The baseline achieves the lowest performance and produces more low-coherence orderings than high-coherence ones. The superiority of EventAll over EntityAll and BOW_All is also consistent with the result of the automatic evaluation. Since the extracted sentences and human-ordered extracts are different from those used in the first experiment, I claim for our task that sentence ordering is not sensitive to extraction methods.

3.3.4.3 Qualitative Evaluation

For a qualitative evaluation that provides a more intuitive understanding of the effect of my method, I select the 200-word extract d080ae from the dataset used in the first experiment and list all its sentences in Figure 3.10. The event terms are boldfaced and the named entities are underlined.

1) Thursday's **acquittals** in the McMartin Pre-School **molestation** case **outraged** parents who **said** prosecutors **botched** it, while those on the defense side **proclaimed** a triumph of justice over hysteria and hype.
2) Originally, there were seven defendants, including Raymond Buckey's sister, Peggy Ann Buckey, and Virginia McMartin, the founder of the school, mother of Mrs. Buckey and grandmother of Raymond Buckey.
3) Seven jurors who **spoke** with reporters in a joint news conference after **acquitting** Raymond Buckey and his mother, Peggy McMartin Buckey, on 52 **molestation** charges Thursday said they felt some children who **testified** may have been **molested** _ but not at the family-run McMartin Pre-School.
4) ``The children were never allowed to **say** in their own words what happened to them," **said** juror John Breese.
5) Ray Buckey and his mother, Peggy McMartin Buckey, were found not guilty Thursday of **molesting** children at the family-run McMartin Pre-School in Manhattan Beach, a **verdict** which brought to a close the longest and costliest criminal trial in history .
6) As it becomes apparent that McMartin cases will **stretch** out for years to come, parents and the former criminal defendants alike are trying to **resign** themselves to the inevitability that the matter may be one they can never leave behind.

Figure 3.10: Extract Sentences of d80ae, 200-word

The human extract is (5, 2, 3, 4, 1, 6). The Baseline, BOW_All, EntityAll, and EventAll versions are (1, 2, 3, 4, 5, 6), (3, 5, 2, 4, 6, 1), (3, 5, 2, 4, 6, 1), and (3, 5, 2, 4, 1, 6), corresponding to Kendall's τ of 0.07, 0.60, 0.60, and 0.73 against the human reference. It turns out that the BOW_All and EntityAll methods produce the same result in this case.

On a closer examination, the news extract is about the acquitting of child molesters. Among the six extracted sentences, 1), 3), and 5) contain the central events of "acquitting" and "molesting". However, important event entities, especially named entities (people's names and the school name in this case), are densely distributed in 3) and 5) but not in 1). It is also true that 3) and 5) have the most (non-stop) words that occur in the other sentences on a bag-of-words account. Either 3) or 5) qualifies as the leading sentence, but 1) is about the "outraging" of parents and "proclaiming" of the defense side as consequences of the "acquitting" event. Only the Baseline chooses it as the leading sentence. On the other hand, all the other three automatic methods choose 3) but a human extractor chooses 5). The human choice is more appropriate as 3) slightly shifts the focus to the "jurors", instead of the central events. We observe that this shows the limitation of the shallow recognition method: EventAll prefers 3) to 5) because of the explicit mention of "acquitting", "molestation", and "molested" in 3) and the explicit mention of only "molesting" in 5). In fact, the "acquitting" event is also reported by 5), but in an implicit ("were found not guilty") way that humans excel at. EntityAll and BOW_All make the same choice because of the

high density of named entities and heavy word overlaps.

The difference between EventAll and EntityAll/BOW_All lies in the order of 1) and 6) and it is on this point that event information, especially event terms, makes a difference. After 4) is in place, which sentence follows it better, 1) or 6)? In terms of entity similarity, neither 1) nor 6) is close to 4); in terms of event similarity, 1) is a better choice because it contains one "said" event and one "proclaimed" event that are related to the "say" and "said" events in 4) whereas 6) only contains one "stretch" event and one "resign" event that are hardly related to the saying events. The BOW method could have captured the overlap of "say" and "said" but unfortunately, such words are filtered out as stopwords. By contrast, common verbs such as "say" are always considered as event terms according to our method. Even if all verbs were excluded from the stopword list, the BOW method would still fail to link the "proclaimed" event with the "say" / "said" event in a semantic way as our event method does. Therefore, only EventAll makes a choice closer to the human's. When entities or bags of words offer little help as in this case, an event coherence-based scheme can be useful. I examined the sentence clusters before the ordering and found that EventAll clusters 1) and 4) together and leaves 6) a cluster by itself. Both EntityAll and BOW_All, however, put 1), 4), 6) in one cluster. This shows that event information can be more helpful than bag of words or entity information and the two-layered clustering scheme is effective.

3.4 Chapter Summary

This chapter discusses techniques to model shallow content-driven coherence in summarization. Most existing works on summary coherence focus on shallow content, especially words or entities. Coherence driven by such shallow content is usually used in a post-extractive fashion in the whole process of summarization – information ordering.

With respect to coherence-oriented information ordering, my works make two breakthroughs. First, I have applied sentence ordering to single-document summarization, a taken-for-granted topic, and come up with surprising results. It turns out that the default text order does not necessarily lead to optimal coherence in the output summary. Second, I have used event, a composite shallow content unit, to represent sentences with a semantically enriched VSM, leading to improvement in the coherence of multi-document summaries. The empirical evidences show that a sentence coherence model based on event information outperforms one based on entity or word information.

Whether it is for single-document or multi-document summarization, the ordering algorithms are designed to model the human style of writing. The model naturally accommodates textual coherence on the local and global level and suits a shallow content-based analysis.

Chapter 4: Deep Content-driven Coherence in Summarization

Not surprisingly, the human summarization process begins with "document exploration" (Endres-Niggemeyer, 1998) instead of word or entity counting. Exploration means human summarizers need to understand a text before deciding on the essential information to be selected for a summary. Usually the selected pieces of information are closely related, making up a coherent summary. Computational approaches inspired by this process are markedly different from those relying merely on statistical evidence of shallow content. As far as text understanding is brought into the picture, we evolve to deep content text analysis and coherence modeling driven by deep content.

Contrasting with shallow content realized as words, entities, or independent events, deep content can be realized as higher-level meaning constructs such as a chronological sequence of related events, logical development from cause to effect, genre-specific textual aspects, topics or threads in communicative text. As deep content analysis amounts to some extent of text understanding, a summary composed of related information from deep content is expected to be highly coherent and readable.

In this chapter, I will present work on two representative kinds of deep content – **textual aspects of news stories** and **speech acts of Twitter posts**. My work will address two easily accessible data sources for NLP, newswire and

Twitter text, with novel approaches. As will be shown, deep content-driven coherence concerns not only sentence ordering but also sentence selection, and abstractive as well as extractive summarization is feasible.

In 4.1, I will situate deep-content driven coherence in a large picture by providing illustrative examples amenable to logical, semantic, rhetorical, or pragmatic accounts. Section 4.2 tries to model coherence based on textual aspects for newswire articles, where an HMM model will be built to capture aspect-based coherence; 4.3 models coherence in Twitter summaries by utilizing speech acts and speech act-based threads; finally, section 4.4 summarizes this chapter.

4.1 Deep Content-driven Coherence and Text Understanding

I will start our exploration by explaining why a shallow-content account of coherence is sometimes not sufficient and how a deep-content account can capture certain instances of coherence that are beyond a shallow-content account.

It is observed that coherent writings usually employ shallow content devices such as word repetitions or lexical cohesion patterns (synonyms, hyponyms, etc.) as in example (4.1), where “village” is repeated. But the reverse is not necessarily true, or in other words, word repetitions or lexical cohesion patterns do not guarantee a coherent text.

(4.1) *By dawn they had arrived at a small* ***village****. People in the* ***village*** *came out to welcome them.*

(4.2) *By dawn they had arrived at a small* ***village****. It was rumored that an Indian* ***village*** *was haunted by snakes and other beasts.*

The word "village" is also repeated in the two consecutive sentences in (4.2), but they hardly make a coherent passage. We may even doubt whether the sentences are about the same village. On the other hand, the absence of such shallow content devices does not mean the absence of coherence.

(4.3) *By dawn they had arrived at a small village. The team was three hours ahead of schedule.*

(4.4) *On Monday, an aircraft reportedly crashed in the mountainous area. Rescue teams found no survivors.*

(4.5) – *What about the new iPad?*

– *The screen is good.*

– *Not very expensive.*

The above examples are obviously coherent, all of which defy a shallow-content account. The two sentences in (4.3) share no common or similar words, but they are causally related as their (the team's) arrival by dawn is the reason of being three hours ahead of schedule. An apt account here is to resort to

discourse relations or **rhetorical relations** (Marcu 1997, 2000) that go beyond shallow content.

A discourse analysis can also be made for (4.4), where the fact that rescue teams found no survivors is a *continuance* of the aircraft crash. But instead of using the underspecified *continuance*, we can analyze the **genre-specific aspects**.

Note that in the following the "genre-specific aspects" are also called "aspects" for short. They should not be confused with grammatical aspects, verb aspects, etc. in other contexts because in this work, "aspects" are invariably semantic and specific to a certain genre.

Apparently, (4.4) comes from a news report about an accident. An accident report typically includes *WHAT* ("an aircraft reportedly crashed"), *WHEN* ("on Monday") and *WHERE* ("in the mountainous area"), plus other details like *REACTIONS* to the accident ("rescue teams"), which are news aspects specific to the accident report. Different genres or types of text have different aspects, and treated as a whole, they make up scripts (Schank and Abelson, 1977) or frames (Fillmore, 1985) in our mental representation. If the organization of the aspects in a text accords with their stereotypical mental representations, the text is perceived to be coherent. For an accident report, *WHAT*, *WHEN*, and *WHERE* are stereotypically presented together at the beginning, which are then followed by *REACTIONS* or *COUNTERMEASURES*, among other possible aspects. Since the aspects in (4.4) are organized in this way, it is perceived to be coherent.

The coherence in the conversational discourse of (4.5) is attributed to a

different reason. In the lack of shallow content links, we understand that both good screen and inexpensiveness are about “the new iPad”, a thread initiated by the first utterance. On the one hand, the last two utterances are answers to the question in the first utterance so that the three utterances are well connected to each other. On the other hand, the first utterance performs a *request* or *question* and the last two perform *comments* on the same thread. Request, question, and comment are instances of **speech acts** that can be used to organize information about the same thread in a coherent way, such as “people asked about … and commented on …”

Rhetorical relations, genre-specific aspects, and speech acts are all instances of deep content inaccessible to shallow parsing methods. Therefore the first challenge of using them is automatically recognizing them. To take the challenge, we need to build dedicated corpora and apply statistical parsing (for rhetorical relations) or text classification (for genre-specific aspects and speech acts) techniques.

Rhetorical relations and their application to (coherent) summarization have been well studied by Marcu (1997, 1999, 2000) on expository texts, where he leverage discourse markers such as “therefore” and “as a result”. But little work has been done on other kinds of deep content, such as genre-specific aspects or speech acts and their roles in coherence-targeted summarization.

In the following, I will introduce my original work on those kinds of deep content – from their automatic recognition to their use in coherence modeling –

which is expected to shed new light on the scope and modeling of coherence in text summarization.

4.2 Coherence Modeling Based on Genre-specific Aspects

In 2010, the summarization track of TAC initialized a new task, aspect-guided summarization, to "encourage a deeper linguistic (semantic) analysis"[6]. Since TAC uses newswire articles as source documents, the aspects are actually news aspects. Specifically, the TAC 2010 organizers define a total of 30 aspects for 5 news categories, covering various key elements of a news story, such as *WHEN* and *WHERE* for "Attacks" stories, *IMPORTANCE* and *THREATS* for "Health and Safety" stories, and *CHARGES* and *PLEAD* for "Investigations and Trials" stories. A complete list of all the TAC-defined news categories and aspects (as well as their brief explanations) is shown in Table 4.1. These aspects are our focus in the following discussions.

Category	Aspect	Explanation
D1. *Accidents and Natural Disasters*	D1.1 *WHAT*	what happened
	D1.2 *WHEN*	date, time, other temporal placement markers
	D1.3 *WHERE*	physical location
	D1.4 *WHY*	reasons for accident/disaster
	D1.5 *WHO_AFFECTED*	casualties (death, injury), or individuals otherwise negatively affected by the accident/disaster
	D1.6 *DAMAGES*	damages caused by the accident/disaster
	D1.7 *COUNTERMEASURES*	countermeasures, rescue efforts, prevention efforts, other reactions to

[6] http://www.nist.gov/tac/2010/Summarization

		the accident/disaster
D2. *Attacks (Criminal / Terrorist)*	D2.1 *WHAT*	what happened
	D2.2 *WHEN*	date, time, other temporal placement markers
	D2.3 *WHERE*	physical location
	D2.4 *PERPETRATORS*	individuals or groups responsible for the attack
	D2.5 *WHY*	reasons for the attack
	D2.6 *WHO_AFFECTED*	casualties (death, injury), or individuals otherwise negatively affected by the attack
	D2.7 *DAMAGES*	damages caused by the attack
	D2.8 *COUNTERMEASURES*	countermeasures, rescue efforts, prevention efforts, other reactions to the attack (e.g. police investigations)
D3. *Health and Safety*	D3.1 *WHAT*	what is the issue
	D3.2 *WHO_AFFECTED*	who is affected by the health/safety issue
	D3.3 *HOW*	how they are affected
	D3.4 *WHY*	why the health/safety issue occurs
	D3.5 *COUNTERMEASURES*	countermeasures, prevention efforts
D4. *Endangered Resources*	D4.1 *WHAT*	description of resource
	D4.2 *IMPORTANCE*	importance of resource
	D4.3 *THREATS*	threats to the resource
	D4.4 *COUNTERMEASURES*	countermeasures, prevention efforts
D5. *Investigations and Trials (Criminal/Legal/Other)*	D5.1 *WHO*	who is a defendant or under investigation
	D5.2 *WHO_INV*	who is investigating, prosecuting, or judging
	D5.3 *WHY*	general reasons for the investigation/trial
	D5.4 *CHARGES*	specific charges to the defendant
	D5.5 *PLEAD*	defendant's reaction to charges, including admission of guilt, denial of charges, or explanations
	D5.6 *SENTENCE*	sentence or other consequences to defendant

Table 4.1: TAC-defined Aspects and Their Explanations

Despite TAC's original motive, many participating systems simply ignored

aspects (Owczarzak and Dang, 2011) or fitted aspects to generic topic models. The topic model approach, such as (Li et al., 2011), assumes that aspects are hidden topics in a document and models or "recognizes" them in an unsupervised way, using the Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) or its variants. A major limitation of this approach is that intuitively, aspects are predefined components of document content instead of topics, and the "aspects" recognized in an unsupervised way hardly match the predefined ones.

A more suitable approach should be supervised and is thus able to address the specific aspects. Teufel and Moens (2002) summarize scientific articles by extracting sentences of certain "rhetorical statuses" – research aspects. Ji et al. (2011) extract facts about entities, events, and relations – generic aspects – to generate query-focused summaries. Genest and Lapalme (2010) apply IE techniques and abstractive summarization to meet the agenda of TAC 2010 but its performance is below average. My work is more akin to (Teufel and Moens, 2002) in that aspect-bearing sentences are extracted to compose summaries.

As intended by TAC organizers, using aspect information encourages deep content analysis and meets highly specific user need. But we are faced with two major problems.

- How do we find aspects in the text?
- Once they are found, how do we use aspects to generate coherent summaries?

To fully solve the first problem, we need to do extensive IE. Some aspects, such as *time* and *place*, can be found by shallow parsing, (regular expression) pattern matching, or Named Entity Recognition (NER), but others, like *Cause* and *Importance*, are beyond traditional IE because they are not literally expressed and have no fixed pattern. Although taking an IE approach holds the promise of abstractive summarization, under the status quo little achievement has been made (Genest and Lapalme, 2010).

My strategy is to scale down the problem by positioning the target at the sentence level. In other words, I am looking for aspect-bearing sentences, instead of aspects themselves (which may be words, phrases, sentences, or paragraphs). This amounts to a less costly task – text (or sentence) classification. Doing so is also compatible with the sentence extraction framework. In order to generate coherent summaries, I will model aspect-based coherence by adapting a Hidden Markov Model (HMM) to guide the arrangement of extracted sentences in the output summary.

The technical details of this work will be unfolded in the three sections that follow: sentence-level aspect recognition (4.2.1), HMM-based coherence modeling (4.2.2), and a summarization approach that utilizes recognized aspects and aspect-level coherence (4.2.3). Section 4.2.4 presents experimental results that validate the effectiveness of the proposed approach.

4.2.1 Sentence-level Aspect Recognition

In my framework, both information selection and ordering are based on sentences. As a prerequisite, aspect-bearing sentences need to be recognized, which is cast as a sentence classification problem. In the following, I will focus on extending the usual word features with meta-phrase features to characterize aspects, especially non-literal aspects. Then I will discuss two difficulties for the problem – multi-label classification and limited training data – and suggest solutions.

4.2.1.1 Feature Extraction

As is the usual practice in text classification, words are used to build the feature space. But since aspects are not necessarily associated with literal content, I also employ a new type of features: meta-phrase features.

I define a meta-phrase as a 2-tuple (m_1, m_2) where m_i is a word/phrase or **word/phrase category**. A word/phrase category is a **syntactic tag**, a **named entity (NE) type**, or the special /NULL/ tag. Syntactic tags represent the logical and syntactic attributes of words in a sentence, including logical constituents (/PRED/ for predicate, /ARG/ for argument) and grammatical roles (e.g., /dobj/ for direct objet, /nn/ for noun modifier). A predicate can be a verb, noun, or adjective and an argument is a noun. The combination of syntactic tags and/or words gives rise to meta-phrases of the **syntactico-semantic pattern**, including the predicate-argument pattern and the argument-modifier pattern. The full list of

syntactic tags is provided in Table 4.2, which shows all the syntactic tags, their abbreviations, and simple examples. The grammatical roles are borrowed from the Stanford Parser dependency relations[7].

	Tag name	Abbreviation	Example
Logical constituents	Predicate	PRED	The lake is **beautiful**.
	Argument	ARG	The **lake** is beautiful.
Grammatical roles	Nominal subject	nsubj	The **company** recalled the drug.
	Controlling subject	xsubj	**Jerry** likes to read books.
	Passive nominal subject	nsubjp	The **victims** were found.
	Agent	agent	12 were injured by the **gunman**.
	Direct object	dobj	Police arrested the **suspect**.
	Indirect object	iobj	The boy gave **her** flowers.
	Noun modifier	nn	**Death** warnings were received.
	Prepositional modifier	prepm	Bell is based in **LA**.
	Adjectival modifier	amod	A **strong** storm arrived.
	Appositional modifier	appos	He is George, an old **friend**.
	Abbreviation modifier	abbrev	I live in New York (**NY**).

Table 4.2: Syntactic Tags Used for Meta-phrase Extraction

NE types represent the semantic attributes of special NPs in a sentence, which are indicative of particular aspects. I use 6 NE types: person (PER), organization (ORG), location (LOC), date (DAT), money (MON), and percentage

[7] http://nlp.stanford.edu/software/dependencies_manual.pdf

(PCT). The combination of NE type and/or NE word/phrase gives rise to meta-phrases of the **name-neighbor pattern**, including the left neighbor-name pattern and the name-right neighbor pattern.

For syntactico-semantic patterns, two related words and their syntactic tags give a total of 4 combinations. Similarly, for name-neighbor patterns, an NE or its type and its left/right neighbor or absence of neighbor (indicated by the /NULL/ tag) gives a total of 4 combinations. Table 4.3 shows example meta-phrases of the two patterns, based on the boldfaced part of the following sentence from our experimental data:

(4.6) *The drugs were withdrawn in September 1997 after* ***a Mayo Clinic study linked fen-phen*** *to potentially fatal heart valve damage.*

Syntactico-semantic patterns	Predicate-argument	*linked fen-phen*	(/PRED/, /dobj/)
			(/PRED/, 'fen-phen')
			('linked', /dobj/)
			('linked', 'fen-phen')
	Argument-modifier	*Clinic study*	(/nn/, /ARG/)
			(/nn/, 'study')
			('Clinic', /ARG/)
			('Clinic', 'study')
Name-neighbor patterns	Left neighbor-name	*a Mayo Clinic*	('a', /ORG/)
			('a', 'Mayo Clinic')
			(/NULL/, /ORG/)
			(/NULL/, 'Mayo Clinic')
	Name-right neighbor	*Mayo Clinic study*	(/ORG/, 'study')
			(/ORG/, /NULL/)
			('Mayo Clinic', 'study')

			(‘Mayo Clinic’, /NULL/)

Table 4.3: Examples of Meta-phrases of the Syntactico-semantic Patterns and the Name-neighbor Patterns

The meta-phrases are designed to capture syntactic relations and NE contexts at different levels of abstraction. Name-neighbor meta-phrase extraction relies on NER; syntactico-semantic meta-phrases are extracted in three scans via dependency parsing.

- Scan for all predicate-argument pairs in the sentence from dependency relations: *nominal subject*, *direct object*, *agent*, etc.;
- Scan for all nominal argument modifiers from dependency relations: *noun modifier*, *appositional modifier*, etc.;
- Scan for all adjectival argument modifiers from the dependency relation of *adjectival modifier*.

In 4.2.4, I will show empirically that meta-phrase features help to better recognize aspects, especially non-literal aspects.

4.2.1.2 Multi-label Classification with Limited Training Data

One sentence may be associated with multiple aspects, as sentence (4.6) contains information about a health issue (*WHAT*) and how it affects people (*HOW*). Aspect recognition on the sentence level is a multi-label classification

problem, which can be transformed to a set of single-label classification problems. Two popular transformation methods are label combination and binary decomposition (Boutell et al. 2004; Tsoumakas and Katakis 2007). The former maps the original k label sets to the 2^k label power sets by transforming all distinct label subsets into single label representations. The latter transforms the original k-label classification into k single-label classifications before aggregating the k classification results to obtain the final result.

A problem with label combination (LC) is that there may not be sufficient training data available for each transformed single-label class, whereas binary decomposition (BD) assumes label independence which does not necessarily hold.

For our task, classification accuracy may suffer from insufficient training data. Besides, a model learned from limited training data may not adapt well to test data due to content differences between training and test data. For example, in the TAC datasets used in our experiments, "health and safety" articles can range from Chinese food safety to protective helmets in the United States.

A promising solution is to use a transductive learner, such as transductive SVM (Vapnik, 1998; Joachim, 1999), which predicts test labels by using the knowledge about test data. So it addresses both training (labeled) data deficiency and model adaptability. Unlike the standard or inductive SVM, transductive SVM is formulated to find an optimal hyperplane to maximize the soft margin between positive and negative objects as well as between training and test data. It

has also been theoretically proved that if properly tuned, transductive SVM generally performs no worse than its inductive counterpart (Wang et al., 2007).

Comparisons of classification methods (BD vs. LC, inductive SVM vs. transductive SVM) will be made in 4.2.4.

4.2.2 HMM-based Coherence Modeling

After aspects are recognized for each sentence, I then model text coherence from a topical perspective. Topics are organizational units that a human writer chooses and arranges to deliver a coherent train of thought. Modeling coherence thus hinges on modeling topic formation and transition. Like (Barzilay and Lee 2004), I use an HMM model with topics as states and sentences as observed sequences. But unlike their model that represents topics on the word level, I use aspects as semantic components of a topic, about which specific words are chosen. Figure 4.1 illustrates the difference between their model and ours with sentence generation mediated by aspects.

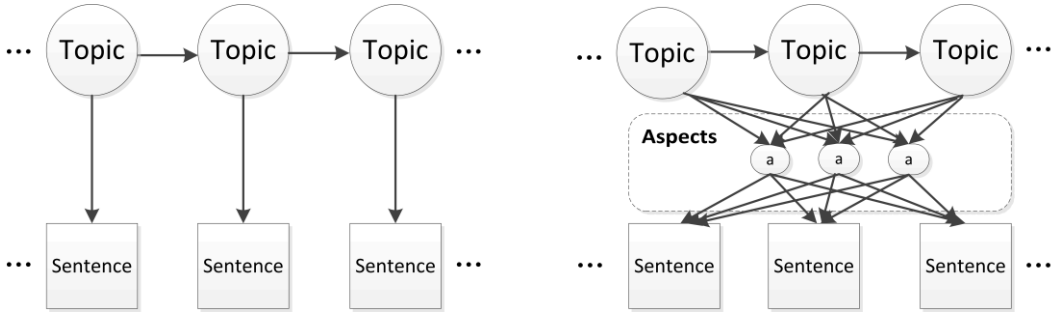

Figure 4.1: Comparison of the HMM Models Without (Left) and With (Right) Aspects

As is shown in the figure, topics and aspects are modeled on different levels and not to be confused. When we write an article, we first plan its structure by focusing on several topics. Take an earthquake report for an example, we may write about its occurrence (topic 1), then the casualties (topic 2), and finally government relief efforts (topic 3). In the next step, each of the topics is fleshed out by a group of related aspects. The Earthquake occurrence (topic 1), for instance, may consist of *WHAT* (aspect 1), *WHEN* (aspect 2), *WHERE* (aspect 3), and even *WHO_AFFECTED* (aspect 4). In this sense, aspects are the building blocks of topics. Clearly, the introduction of aspects contributes to a more intuitive modeling of the human writing process.

The choice of aspects and their order to make up a topic is also guided by content-level coherence. In the following, I list three possible ways to compose the topic of an earthquake occurrence on the aspect level.

4.7) *A massive earthquake measuring 7.8 on the Richter scale* (*WHAT*) *rocked China's Sichuan Province* (*WHERE*) *on May 12* (*WHEN*), *leaving at least 12,000 dead and 26,000 injured* (*WHO_AFFECTED*).

4.8) *A massive earthquake measuring 7.8 on the Richter scale* (*WHAT*) *rocked China's Sichuan Province* (*WHERE*). *Before relief efforts* (*COUNTERMEASURES*) *were made, no one knew the exact cause of the earthquake* (*WHY*).

4.9) *At least 12,000 dead and 26,000 were injured* (*WHO_AFFECTED*) *in China's Sichuan Province* (*WHERE*). *A massive earthquake measuring 7.8 on the Richter scale* (*WHAT*) *occurred on May 12* (*WHEN*)

In terms of content, (4.7) is the most coherent. When a different set of aspects are used (4.8) or the order of the aspects are disrupted (4.9), incoherent results. My model is designed to capture this kind of coherence under the level of topic. In the following, I will give more technical details of the model.

4.2.2.1 Topic Induction

In this model, the topics are represented by sentence clusters. Like (Barzilay and Lee, 2004), I use complete-link hierarchical clustering to cluster sentences. But unlike their work, I vectorize sentences using both words and aspects. The aspects are twice as much weighted as the words, which corresponds to the aspect's conceptual significance. The use of aspect-weighted hierarchical clustering will be justified in 4.2.4.

After obtaining the initial clusters, all small clusters (with cluster size < M) are merged into one cluster because they possibly contain non-essential information. I denote this merged cluster as c_0.

4.2.2.2 HMM Parameter Estimation

Given topics (i.e., clusters) c_0, c_1, … and their corresponding HMM states s_0,

s_1, …, I now estimate the HMM parameters. With no prior knowledge about the topics, I assume uniform distribution for the **state probabilities**. Let's denote aspects as a_i's and words as w_i's. Given a sentence $x = w_1 w_2 ... w_n$ having aspects: $\{a_1, ..., a_m\}$ and state s ($s \neq s_0$), the **emission probability** $P(x|s)$, shorthanded $P_s(x)$, is defined as:

$$P_s(x) = \sum_{i=1}^{m} P_s(x \mid a_i) P_s(a_i) \quad (1)$$

For aspect $a \in A$, the set of all aspects, MLE is used to estimate the raw probability of $P_s{}^*(a)$: ($s \neq s_0$)

$$P_s{}^*(a) = (Count_c(a) + \delta_1) / (\sum_{a'} Count_c(a') + \delta_1 |A|)$$

where $Count_c(a)$ is the count of a in cluster c (corresponding to s) and δ_1 is a smoothing coefficient. Note that some sentences may not have aspects and in this case, we use a special a_0 to represent the "empty aspect" and: ($s \neq s_0$)

$$P_s{}^*(a_0) = \prod_{i=1}^{|A|} (1 - P_s{}^*(a_i))$$

The raw probabilities are normalized so that they sum up to 1: ($s \neq s_0$)

$$P_s(a) = P^*{}_s(a) / \sum_{a'} P^*{}_s(a')$$

$P_{s_0}(a)$ is made complementary to the other $P_s(a)$'s, as in (Barzilay and Lee, 2004):

$$P_{s_0}(a) = (1 - Max_{s' \neq s_0} P_{s'}(a)) / \sum_{a' \in A \cup \{a_0\}} (1 - Max_{s' \neq s_0} P_{s'}(a'))$$

$P_s(x|a)$ in equation (1) can be estimated by taking the aspect-conditioned word generation and a bigram language model:

$$P_s(x \mid a) = P_s(w_{1...} w_n \mid a) \approx \prod_{i=1}^{n} (P_s(w_i \mid a) + P_s(w_i \mid w_{i-1}))$$

$$P_s(w \mid a) = (Count_c(w(a)) + \delta_2) / (Count_c(a) + \delta_2 |V|)$$

where $Count_c(w(a)) = |\{a' : w \in s \wedge s \supset a' \wedge a' \in c\}|$, $s \neq s_0$, and V is the vocabulary.

For $P_{s_0}(w \mid a)$,

$$P_{s_0}(w \mid a) = (1 - Max_{s' \neq s_0} P_{s'}(w \mid a)) / \sum_{w' \in V} (1 - Max_{s' \neq s_0} P_{s'}(w' \mid a'))$$

We use the Bayesian rule for a_0:

$$P_s{}^*(w \mid a_0) = P_s(w) P(a_0 \mid w) / P_s(a_0) = P_s(w) \prod_{i=1}^{p} (1 - \frac{P_s(w \mid a_i) P(a_i)}{P_s(w)}) / P_s(a_0) \tag{2}$$

After normalization,

$$P_s(w \mid a_0) = P^*{}_s(w \mid a_0) / \sum_{w'} P^*{}_s(w' \mid a_0)$$

To calculate $P_s(w)$ in equation (2), for $s \neq s_0$,

$P_s(w) = (Count_c(w) + \delta_3) / (\sum_{w' \in V} Count_c(w') + \delta_3 |V|)$, and for s_0,

$$P_{s_0}(w) = (1 - Max_{s' \neq s_0} P_{s'}(w)) / \sum_{w' \in V} (1 - Max_{s' \neq s_0} P_{s'}(w'))$$

The calculation of $P_s(w)$ is straightforward.

$$P_s(w) = \frac{Count_c(w) + \delta}{\sum_{w' \in V} Count_c(w') + \delta |V|} \quad (s \neq s_0), \quad P_{s_0}(w) = \frac{1 - Max_{s' \neq s_0} P_{s'}(w)}{\sum_{w' \in V} (1 - Max_{s' \neq s_0} P_{s'}(w'))}$$

$$P_s(w' \mid w) = \frac{Count_c(ww') + \delta}{Count_c(w) + \delta |V|} \quad (s \neq s_0),$$

$$P_{s_0}(w' \mid w) = \frac{1 - Max_{s' \neq s_0} P_{s'}(w' \mid w)}{\sum_{u \in V} (1 - Max_{s' \neq s_0} P_{s'}(u \mid w))}$$

Finally, we have

$$P_s{}^*(x \mid a) = \prod_{i=1}^{n} (P_s(w_i \mid a) + P_s(w_i \mid w_{i-1}))$$

After normalization,

$$P_s(x \mid a) = P^*{}_s(x \mid a) / \sum_{x'} P^*{}_s(x' \mid a)$$

The state **transition probabilities** are estimated from two sources: sentences ($P_{sent}(s_j|s_i)$) and aspects ($P_{aspect}(s_j|s_i)$).

$$P_{sent}(s_j \mid s_i) = (SC(c_i, c_j) + \delta_4) / (SC(c_i) + \delta_4 r)$$

$$P_{aspect}(s_j \mid s_i) = (AC(c_i, c_j) + \delta_5) / (\sum_{j=1}^{r} AC(c_i, c_j) + \delta_5 r)$$

In the above, r is the total number of topics (states). $SC(c, c')$ represents the count of documents where a sentence from c immediately precedes a sentence from c'. $SC(c)$ represents the total count of documents with sentences from c. $AC(c, c')$ represents the count of documents where a sentence from c contains an aspect that immediately precedes an aspect contained in a sentence from c'. Aspect precedence is estimated by aspect-bearing sentence precedence.

We can estimate the sentence-based state transitions and the aspect-based state transitions differently because unlike sentences, aspects are not unique in a document. The final transition probability is a linear combination of them:

$$P(s_j \mid s_i) = \lambda_1 P_{sent}(s_j \mid s_i) + (1 - \lambda_1) P_{aspect}(s_j \mid s_i)$$

where λ_1 is a coefficient in 0 .. 1 to be empirically decided.

4.2.2.3 Coherence Accommodation

Human writers do not arrange sentences randomly, so the order of sentences embodies a coherent pattern in terms of aspects and words. For example, in a terrorist attack report, a sentence about the *time* and *place* of the attack presumably precedes a sentence about the *casualties*. For two sentences about *casualties*, the sentence that mentions general facts (“victims”, “died”, “killed”)

may precede the sentence that gives specific names, ages (“40-year-old”), identities (“tourists”), etc.

But the model built so far does not account for sentence order information in the training data. To utilize this important coherence information, after the model is initially trained, I re-cluster the sentences by assigning each one to the topic (state) that most likely emits it, determined by Viterbi decoding. Then the HMM parameters are re-estimated using the new states. We iterate this process until clusters stabilize (Barzilay and Lee, 2004).

The trained HMM model enables us to determine the most coherent sentence ordering. I first permute the sentences and then among all the sentence permutations select one with the highest likelihood from the model, which is computed by the forward algorithm.

4.2.3 Summarization for Aspect-level Coherence

After aspect-bearing sentences are recognized and aspect-level coherence is modeled, I can proceed to do extractive summarization. In the following I introduce an aspect-guided summarizer following the canonical pipeline of sentence selection and sentence ordering.

4.2.3.1 Base Summarizer

The aspect-guided summarizer is built on top of a simple but robust base summarizer (Zhang et al., 2011) that utilizes word frequencies. The following

formula is used to calculate the frequency score of a sentence *s* in document set *D*.

$$freq\_score(s) = \frac{\sum_{w \in s} TF_s(w) \cdot score(w)}{\sum_{w \in s} TF_s(w) \cdot ISF(w)}$$

In this formula, $score(w) = \log TF_D(w)$, and the word *w* is a frequent or document topic word[8]; otherwise *score*(*w*) = 0. *ISF*(*w*) is the inverse sentence frequency of *w* in the document set, defined as $ISF(w) = \log(N_s / SF_D(w))$. $TF_s(w)$ and $TF_D(w)$ are the frequencies of *w* in *s* and *D*; $SF_D(w)$ is the sentence frequency of *w* in *D* and N_s is the total number of sentences in *D*. The ISF-based sentence length is used to give words different weights when counting sentence length. If a word is more dominant in the input document set, it should be considered shorter so that the sentence containing it should be penalized less by length.

Summary sentences are selected iteratively until the length is reached. In each iteration, we select the top ranking sentence *s** and then discount the frequency of all the words in *s** by multiplying $\alpha < 1$. In our experiment, α is set to be 0.9. Redundant sentences (with cosine similarity > 0.7) are discarded.

4.2.3.2 Sentence Selection

To bias sentence selection towards aspects, I integrate the recognized sentential aspect information into the base summarizer.

For a sentence *s*, I first calculate its aspect score:

$$aspect\_score(s) = \sum_{asp \in s} classify\_score(asp),$$

[8] For TAC data, it is a word used in the description of a document set.

where *classify_score*(*asp*) indicates the classification confidence for aspect *asp*. For our current scheme, it is the value calculated by the decision function trained from transductive SVM.

The final score of sentence *s* is a linear combination of its frequency score and aspect score.

$$score(s) = \lambda_2 \times freq\_score(s) + (1-\lambda_2) \times aspect\_score(s),$$

where λ_2 is a coefficient in 0 .. 1, to be decided empirically. The iterative sentence selection algorithm is similar to that described for the base summarizer. The main difference is that after each iteration, not only the word scores but also the aspect scores are updated. For any aspect *asp* in a selected sentence, $classify\_score(asp) = \beta \cdot classify\_score(asp)$, $\beta < 1$. In our experiment, β is set to be 0.9.

4.2.3.3 Coherence-oriented Sentence Ordering

After we select all the sentences that meet the summary length requirement, we order them by considering all possible sentence permutations. Since aspect-guided summaries and source documents obviously differ in aspect density and content structure, I train an aspect-based HMM model with aspect-annotated human summaries. Then I select the best ordering among all sentence permutations as the sequence with the highest likelihood according to the HMM model parameters. This straightforward approach integrates well into the selection-ordering scheme. In 4.2.4, I will show the efficacy of our simple

method, especially for coherence enhancement.

I would like to point out that for multi-document summarization, the summarization strategy in (Barzilay and Lee, 2004), which attempts to correlate summary sentences with source sentences, cannot be adopted because their method only works for single-document summarization. It is simply pointless to train an HMM model with sentences from different documents as if they were from the same document.

4.2.4 Experimental Results

In this part, I will report evaluation results on the proposed approach. As prerequisites for the summarization work, sentence-level aspect recognition and aspect-level coherence ordering will be evaluated individually. Then the summary output will be evaluated on TAC's benchmark dataset.

4.2.4.1 Data Preparation

Because aspect recognition is intended to find summary-worthy sentences in the source documents and coherence modeling is intended to arrange summary sentences in a coherent way, different datasets are used for our purpose.

- **Sentence-level Aspect Recognition**

It is evaluated on the TAC 2010 source documents. Table 4.4 shows the details of our experimental data. I employed a human annotator to label each of the sentences with a TAC-defined list of aspects. For brevity, I will only report

the results on "health and safety" (codename D3), the largest category. The results on the other 4 categories are similar. Table 4.5 lists the 5 aspects with brief explanations for D3, which is part of Table 4.1.

Category	# Documents	# Sentences
D1. Accidents and Natural Disasters	140	2858
D2. Attacks	140	2845
D3. Health and Safety	240	5897
D4. Endangered Resources	200	4007
D5. Investigations and Trials	200	4563
TOTAL	920	20170

Table 4.4: Details of the TAC 2010 Documents

Aspect	Explanation
D3.1 *WHAT*	what is the issue
D3.2 *WHO_AFFECTED*	who is affected by the health/safety issue
D3.3 *HOW*	how they are affected
D3.4 *WHY*	why the health/safety issue occurs
D3.5 *COUNTERMEASURES*	countermeasures, prevention efforts

Table 4.5: Aspects and their Explanations for D3 (Health and Safety)

Note that D3.3 and D3.4 are clearly non-literal aspects. In some cases, D3.5

is also not literally expressed. The following is the aspect-annotated version of example (4.6), a sentence from D3.

(4.10) *The drugs were withdrawn in September 1997 after a Mayo Clinic study linked fen-phen to potentially fatal heart valve damage.* {**D3.1**, **D3.3**, **D3.5**}

- **Coherence Modeling**

It is evaluated using the TAC 2010 and TAC 2011 human summaries. Similar to what I did for aspect recognition, I employed the same annotator to annotate the TAC 2010 and 2011 human summaries with aspects. Table 4.6 shows the details of the datasets.

Category	**TAC 2010**		**TAC 2011**	
	# Summaries	**# Sentences per Summary**	**# Summaries**	**# Sentences per Summary**
D1	28	5.96	36	6.31
D2	28	6.11	36	6.28
D3	48	6.23	40	6.23
D4	40	5.70	32	5.59
D5	40	5.98	32	6.22
TOTAL	184	6.00	176	6.14

Table 4.6: Details of the TAC 2010 and TAC 2011 Human Summaries

The TAC data include initial and update summaries. Since I didn't consider the "update" factor, only initial summaries are used. All the human summaries are 100 words long according to the TAC requirement. The TAC 2010 dataset is used to train the model and the TAC 2011 dataset is used for testing.

- **Summarization Output**

This task is evaluated on the TAC 2011 datasets for initial summarization. TAC 2011 includes 44 document sets and requires one initial summary for each of them. The human annotators recruited by TAC organizers produce 4 different summaries for each of the document sets, which can be used for evaluation.

4.2.4.2 Evaluation of Aspect Recognition

To extract meta-phrase features used for sentence-level aspect recognition, I use the Stanford Parser (Klein and Manning, 2003) to do dependency parsing and extract meta-phrases of the syntactico-semantic pattern. I also use the OpenNLP tools[9] to find named entities for those of the name-neighbor pattern. Features that occur only once are filtered. The classifiers are inductive SVM and transductive SVM, which are implemented by using the SVM^{light} tool[10] with a linear kernel and default settings.

To test the effectiveness of meta-phrase features, I compare 3 features sets (words, meta-phrases, words + meta-phrases) and show the F-measures on all

[9] http://opennlp.sourceforge.net/
[10] http://svmlight.joachims.org/

aspects of D3 in Figure 4.2. The classifier is inductive SVM and the result is based on ten-fold cross validation.

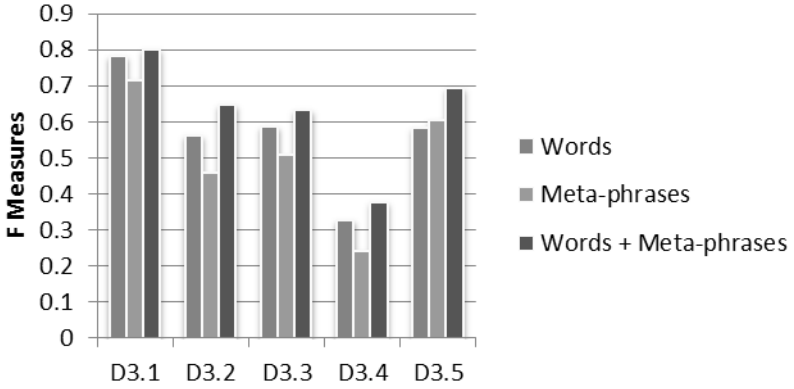

Figure 4.2: F-measures of Different Feature Sets on D3 Aspects

Although outperformed by the word features if used alone, the meta-phrase features consistently help to improve classification performance on all aspects, including those that may not be literally expressed (e.g., D3.3, D3.4, D3.5).

To test the multi-label and transductive learning scheme, I randomly select a small training set of 300 sentences as labeled data and 3000 different sentences to be used as unlabeled data. I vary the size of unlabeled data as (300, 600, 900, …, 3000) so that the labeled/unlabeled ratio ranges from 1:1 to 1:10. Both word and meta-phrase features are used. I compare multi-class transformations (BD vs. LC) and classifiers (inductive SVM vs. transductive SVM). The evaluation metric is macro-average F measure, i.e., the average of F-measures on individual aspects. Figure 4.3 shows the aggregate result.

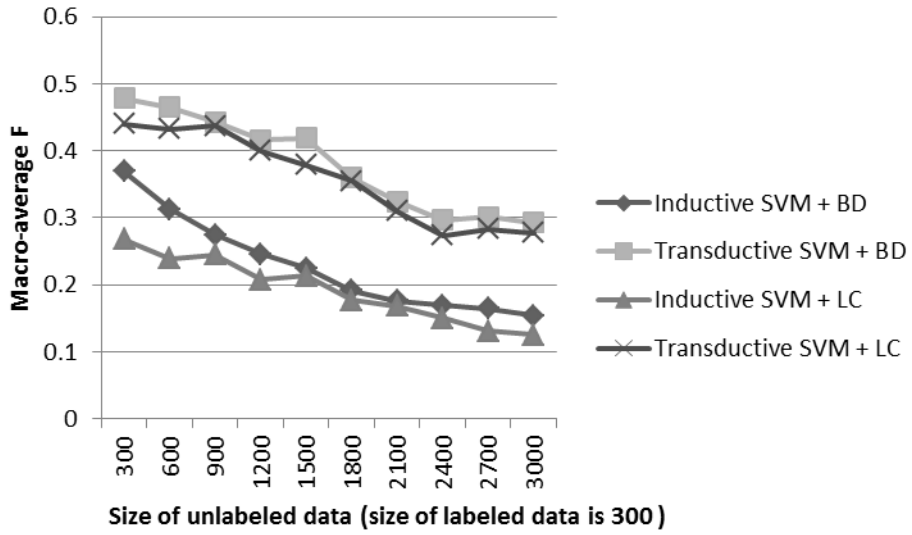

Figure 4.3: Macro-average F on D3 with Different Sizes of Unlabeled Data

Transductive SVM is markedly superior to inductive SVM, so is binary decomposition to label combination. Moreover, the classification performance is more sensitive to a good adaptive (transductive) learner than a good multi-label transformer.

Note that similar results are achieved on the other 4 categories. Such empirical evidence is used to train a model (using word and meta-phrase features, binary decomposition, and transductive SVM) for the following operations.

4.2.4.3 Evaluation of Coherence Modeling

First of all, the HMM-based coherence model depends on the quality of induced topics from sentence clustering. Therefore, I first find the optimal clustering scheme by experiments.

Using all the 184 summaries of the TAC 2010 dataset, I evaluated different

clustering schemes. Specifically, I varied the weight of the role played by aspects: not used (weight = 0), weighted as much as words (weight = 1), weighted twice as much as words (weight = 2). Then 3 popular clustering methods – K-means, spectral clustering, (complete-link) hierarchical clustering – are also compared. Combining aspect weights and clustering methods results in 9 clustering schemes. Each of them is applied to a mixture of all the sentences in a category (D1 to D5) so that the original summaries in that category can be used as "ground truth" clusters for evaluation and cluster number setting (cluster number = summary number).

I adopted the tools in scikit-learn[11], a Python module, with default settings except for the number of clusters. The clustering result is evaluated using the **Rand** index (Rand, 1971), which computes how similar the clustering results are to the ground truth clusters (summaries). Rand index ranges in 0 .. 1 and larger values mean better performance. The results are averaged over the 5 categories and shown in Table 4.7.

	K-means	**Spectral**	**Hierarchical (complete-link)**
Aspect Weight = 0	0.150	0.151	0.161
Aspect Weight = 1	0.130	0.116	0.157
Aspect Weight = 2	0.192	0.138	0.247

Table 4.7: Rand Index of Clustering Schemes

[11] http://scikit-learn.org/stable/index.html

Apparently, the best result is achieved with hierarchical clustering and doubly weighted aspects. This scheme will be used to evaluate the HMM-based model.

Due to aspect differences among categories, I evaluated the HMM-based coherence model on the category level. For each category, I used a random 80% of the TAC 2010 data for training and the rest for development. I tune the HMM model parameters (M, δ_1 to δ_5, and λ_1) as well as the number of topics (states) on the development data. The built model is then tested on the TAC 2011 data.

Barzilay and Lee (2004) have shown the superiority of their HMM model in ordering to a baseline bigram model and a different probabilistic ordering model (Lapata, 2003). Now I report how the aspect-extended HMM model compares with their aspect-agnostic model.

The model's quality in coherence modeling is indicated by how well the model can order a scrambled collection of sentences from a human summary. As the human summary provides the gold standard order of the sentences, I used Kendall's τ, a widely used sequence ordering metric (Lapata, 2006), to measure how similar the model-predicted ordering is to the gold standard. The value of τ ranges in –1 .. 1, with larger values indicating higher similarities. If the ordering is identical to the gold standard, $\tau = 1$; if it is the reversed gold standard, $\tau = 0$. Figure 4.4 shows the results on the 5 categories as averages for all the summaries in the same category.

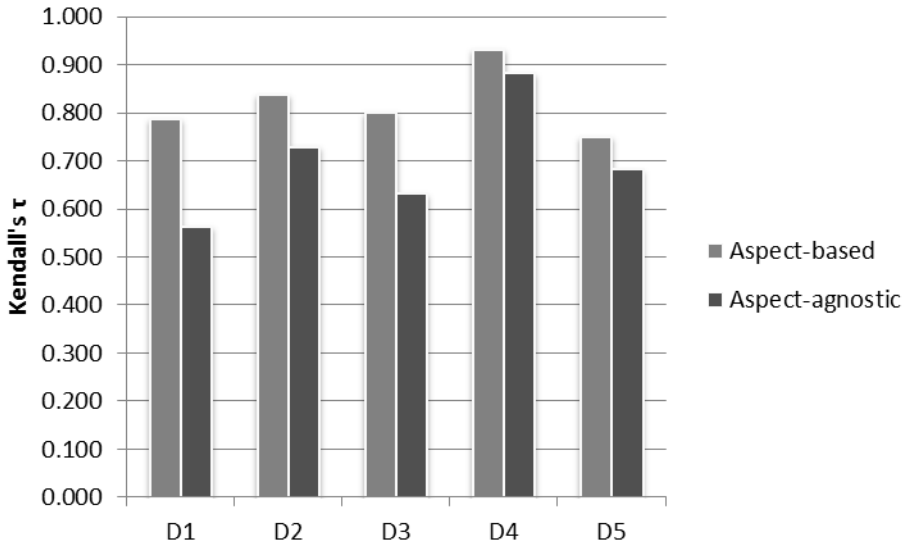

Figure 4.4: Comparison of the HMM models with and without Aspects

The aspect-based model consistently defeats the aspect-agnostic model that relies only on literal information. This shows that aspect information helps the HMM-based model to better capture the pattern of coherent ordering.

4.2.4.4 Evaluation of Summary Output

For summarization, aspect information is used in two stages: selecting sentences to cover salient and aspect-relevant information and ordering the selected sentences to enhance coherence. Aspect recognition and sentence ordering are done with the trained models according to the previous empirical results. Both information coverage and summary coherence are to be evaluated.

Information coverage is evaluated with the standard ROUGE toolkit (Lin and Hovy, 2003) to measure the ngram overlap between automatic summaries and human summaries, which is insensitive to sentence order. The TAC

organizers have released the official results of **ROUGE-2** (bigram overlap) and **ROUGE-SU4** (skip bigram overlap, with up to 4 words as the skip distance). Table 4.8 shows the official results of some of the participating systems, where "Top" is the top ranking participant in TAC 2011 and "Average" is the average over all 50 TAC participants.

	ROUGE-2 (Rank)	**ROUGE-SU4 (Rank)**
Top	0.1337 (1)	0.1636 (1)
Base Summarizer	0.1206 (5)	0.1570 (4)
Base Summarizer + Aspect	**0.1223 (4)**	**0.1581 (3)**
Average	0.0932	0.1266

Table 4.8: ROUGE-2 and ROUGE-SU4 of Summaries on TAC 2011

The base summarizer is a very competitive system (TAC ID: 4) in TAC 2011, ranking 5th and 4th in terms of ROUGE-2 and ROUGE-SU4, but it is outperformed by its aspect-enhanced version ("Base Summarizer + Aspect", TAC ID: 24) ranking 4th and 3rd in terms of ROUGE-2 and ROUGE-SU4. I observe that using recognized aspects helps to include more target information – aspects – in summaries, thus leading to summaries that are closer to human-written summaries. But the improvement is limited, partly because the base summarizer has already included many aspects that happen to contain a lot

of high-frequency words – a point I will illustrate with a sample summary in the following.

To test the effectiveness of aspect in enhancing coherence, I employed two human judges to rate the coherence of summaries on a scale of 5 points, the higher the more coherent. For each of the 44 document sets of TAC 2011, I asked the judges to rate 4 summaries: 3 automatic summaries and 1 randomly selected human summary. The automatic summaries are identical in content coverage as they all result from the same summarizer ("Base Summarizer + Aspect" in Table 4.8). They differ from each other only in sentence ordering[12]: following the selection sequence determined by sentence ranking scores ("Ranking ordering"), using the HMM model without aspect, i.e., Barzilay and Lee's (2004) model ("BL ordering"), using the HMM model with aspect ("Aspect ordering"). Note that "Ranking ordering" is used in our submitted summary version "Base Summarizer + Aspect". "BL ordering" and "Aspect ordering" can be regarded as its coherence-enhanced variants.

For the human rating results, Cohen's Kappa is computed to be 0.71, indicating high inter-judge agreement. Table 4.9 lists the result, with the scores averaged over the two judges.

[12] Strictly speaking, this is not entirely true because I truncate the summaries to meet the exact length limit of 100 words. Due to ordering, if two such summaries end in two different sentences that are made incomplete by truncation, they will be slightly different.

Ranking ordering	BL ordering	Aspect ordering	Human
2.75	3.45	3.73	4.70

Table 4.9: Human Rating Results for Coherence

The differences between the two HMM ordering versions and the "Ranking ordering" or "Human" are very significant ($p < 0.0001$ on a paired two-tailed t-test). The difference between BL ordering and Aspect ordering is also significant ($p = 0.017$), though to a lesser degree. The 3.73 point by "Aspect ordering" proves that aspect-based ordering helps to generate fairly coherent summaries. It is also obvious that a large gap exists between the coherence of human-written summaries and automatic summaries.

For a more intuitive understanding of aspect-based sentence selection and ordering, I provide sample summaries for one of the TAC 2011 document sets (ID: D1110B). This document set consists of 10 news reports about the 2008 earthquake in Sichuan, China, and belongs to category D1 – "accidents and natural disasters" – associated with 7 aspects shown in the upper-left corner of Table 4.10.

A total of 5 different summaries are compared in this table. Among them, S1 is a human summary (ID: D1110-A.M.100.B.H) and S2–S5 are automatic summaries of the same length (100 words). S2 consists of sentences selected without aspect information and ordered by ranking ordering. S3–S5 each consists of sentences selected with aspect information, which are ordered in different

ways: S3 by ranking ordering, S4 by BL ordering, and S5 by Aspect ordering. Note that S2 differs from S3–S5 in terms of sentence set but S3–S5 differs from each other only in terms of sentence order (with slight content change due to truncation, see Footnote 12).

To facilitate the following discussion, I annotate the sentences with sequence numbers so that S3-6 refers to sentence (6) of summary S3. I also annotate each sentence with its aspects from the predefined list. An empty { } means the sentence contains no aspect. Note that the sequence numbers and aspect lists are not part of the summaries themselves.

The human summary is indeed guided by aspects as each constituent sentence contains some aspect to meet the information need[13], as are the automatic summaries with sentences selected using aspect information (S3–S5). In contrast, S2 contains a non-aspect-bearing sentence (S2-3) that is undesirable. The inclusion of S2-3 can be attributed to the system's lack of aspect knowledge, but surprisingly all the other sentences of S2 contain aspects. After scrutinizing the result, I observe that those aspect-bearing sentences are really "happy coincidences" – the desired aspects are embodied by high-frequency words such as "earthquake", "Sichuan", "China", "Monday", and "people". Therefore, using a frequency-based summarizer easily covers some aspect information. This also explains the good performance of the base summarizer and limited improvement by incorporating aspect information (Table 4.8).

[13] Aspects D1.4 (*WHY*) and D1.6 (*DAMAGES*) are not included because they have not appeared in the source documents.

Category	**D1**	**S1** (human)
Aspects	D1.1 *WHAT* D1.2 *WHEN* D1.3 *WHERE* D1.4 *WHY* D1.5 *WHO_AFFECTED* D1.6 *DAMAGES* D1.7 *COUNTERMEASURES*	(1) A massive earthquake measuring 7.8 on the Richter scale rocked China's Sichuan Province on May 12, leaving at least 12,000 dead and 26,000 injured. **{D1.1, D1.2, D1.3, D1.5}** (2) Chinese authorities had detected no warning signs ahead of the quake, China's worst since 1976. **{D1.1, D1.3}** (3) The Chinese have allocated $29 million for disaster relief, and some foreign governments, including those of Germany, Norway, and Belgium have pledged relief funds. **{D1.7}** (4) In addition, the State Ethnic Affairs Commission has allocated $285,000 for aid to ethnic minorities in Sichuan Province. **{D1.3, D1.5, D1.7}** (5) Rain is forecast for the coming days, which could hamper relief efforts. **{D1.7}**

S2 (non-aspect-guided selection, ranking ordering)	**S3** (aspect-guided selection, ranking ordering)
(1) The 7.8-magnitude earthquake struck Sichuan province shortly before 2:30 pm on Monday. **{D1.1, D1.2, D1.3}** (2) The Belgian government pledged on Tuesday an initial relief fund of 250,000 euros to China after a powerful earthquake struck southwestern China on Monday. **{D1.1, D1.2, D1.3, D1.7}** (3) All of those provinces and Chongqing, a special municipality of more than 30 million people, border Sichuan. **{ }** (4) He expressed condolences to the Communist Party, State and people of China and families of the	(1) The 7.8-magnitude earthquake struck Sichuan province shortly before 2:30 pm on Monday. **{D1.1, D1.2, D1.3}** (2) He expressed condolences to the Communist Party, State and people of China and families of the earthquake's victims. **{D1.1, D1.5, D1.7}** (3) China has allocated 200 million yuan for disaster relief work after an earthquake rocked the country's southwest killing more than 8,700 people. **{D1.1, D1.3, D1.5, D1.7}** (4) Vietnam has expressed deep sympathies to China at huge

earthquake's victims. **{D1.1, D1.5, D1.7}** (5) China has allocated 200 million yuan for disaster relief work after an earthquake rocked the country's southwest killing more than 8,700 people. **{D1.1, D1.3, D1.5, D1.7}** (6) Xinhua said 8,533 people had died in **{D1.5}**	losses caused by an earthquake in China's southwestern Sichuan province. **{D1.1, D1.3, D1.7}** (5) Xinhua said 8,533 people had died in Sichuan alone, citing the local government. **{D1.3, D1.5}** (6) Russia's new President Dmitry Medvedev sent condolences and an offer of help to his Chinese counterpart **{D1.7}**
S4 (aspect-guided selection, BL ordering)	**S5** (aspect-guided selection, aspect ordering)
(1) The 7.8-magnitude earthquake struck Sichuan province shortly before 2:30 pm on Monday. **{D1.1, D1.2, D1.3}** (2) He expressed condolences to the Communist Party, State and people of China and families of the earthquake's victims. **{D1.1, D1.5, D1.7}** (3) China has allocated 200 million yuan for disaster relief work after an earthquake rocked the country's southwest killing more than 8,700 people. **{D1.1, D1.3, D1.5, D1.7}** (4) Xinhua said 8,533 people had died in Sichuan alone, citing the local government. **{D1.3, D1.5}** (5) Vietnam has expressed deep sympathies to China at huge losses caused by an earthquake in China's southwestern Sichuan province. **{D1.1, D1.3, D1.7}** (6) Russia's new President Dmitry Medvedev sent condolences and an offer of help to his Chinese counterpart **{D1.7}**	(1) The 7.8-magnitude earthquake struck Sichuan province shortly before 2:30 pm on Monday. **{D1.1, D1.2, D1.3}** (2) Russia's new President Dmitry Medvedev sent condolences and an offer of help to his Chinese counterpart Hu Jintao after Monday's earthquake. **{D1.1, D1.2, D1.7}** (3) He expressed condolences to the Communist Party, State and people of China and families of the earthquake's victims. **{D1.1, D1.5, D1.7}** (4) China has allocated 200 million yuan for disaster relief work after an earthquake rocked the country's southwest killing more than 8,700 people. **{D1.1, D1.3, D1.5, D1.7}** (5) Xinhua said 8,533 people had died in Sichuan alone, citing the local government. **{D1.3, D1.5}** (6) Vietnam has expressed deep sympathies to China at huge losses caused by an earthquake **{D1.1, D1.7}**

Table 4.10: Comparison of Summaries for a Document Set (D1110B)

Now let's focus on the coherence of S3–S5 which, barring the effect of truncation, contain the same set of sentences. The sentences in S3 are ordered according to their rankings in the selection process, i.e., those with high-frequency words as well as high-fidelity aspects are placed first. But there is no guarantee that the sentences so ordered are related to each other in a coherent way. S3-1 and S3-2 are the highest-ranking sentences but they do not connect. The "he" in S3-2 loses its referent and leads to poor coherence. In addition, S3-4 and S3-6 are both about overseas response to China's earthquake and intuitively they should stand next to each other. But the intrusion of S3-5, which reports earthquake casualties, breaks the coherence. This problem disappears in S4, in which S4-5 (= S3-4) and S4-6 (= S3-6) are adjacent to each other. Moreover, S4-3 (= S3-3) and S4-4 (= S3-5) make good neighbors because they both mention earthquake casualties. Intuitively, the HMM content model rearranges the selected sentences in a more coherent way.

When augmented with aspect information, the HMM model can do even better. The only difference between S4 and S5 is in S5-2 (= S4-6). Now it is just before S5-3 (=S4-2) and removes the dangling anaphora of "he", which is resolved as "Russia's new President Dmitry Medvedev". Thus S5 is more coherent than S4.

The human summary is qualitatively superior to the automatic summaries because abstraction, not extraction, is used so that it saves space for more information. For example, S2-3 mentions both China's relief allocation and

foreign countries' responses. But it takes 3 to 4 sentences in the automatic summaries to cover approximately the same amount of information. S2 is also the most coherent, with smooth sentence transitions ("In addition" in S2-4) and clearer aspect-level organization, from the earthquake's happening (*WHAT*, *WHEN*, *WHERE*) to its casualties and countermeasures (*WHO_AFFECTED*, *COUNTERMEASURES*). In comparison, the automatic summaries lack such a clear organization, leading to significantly lower coherence. But our analysis does confirm that aspect information can help improve the coherence of the summary output.

4.3 Coherence Modeling Based on Speech Acts

Deep content-driven coherence does not only apply to news documents or extractive summarization. In this section, I will embark on a new task, **abstractive summarization** of Twitter posts (tweets) by using a kind of linguistically rooted deep content – **speech act**.

With the proliferation of messages on social media such as Twitter, efficient processing of such messages is in urgent need. The summarization technology by nature is an apt answer to the call. I will present my work on Twitter topic summarization, which is formally defined as summarizing a large number of tweets belonging to the same topic. Different from other multi-document summarization tasks such as those for newswire articles, Twitter topic summarization has its unique characteristics and generating coherent Twitter

summaries is a tough challenge. To explicate this point, let's examine the snapshot from Twitter.com in Figure 4.5, which shows several tweets under the topic of *#sincewebeinghonest*. The user accounts have been blotched to protect people's privacy and the tweets have been annotated to facilitate the following explanation.

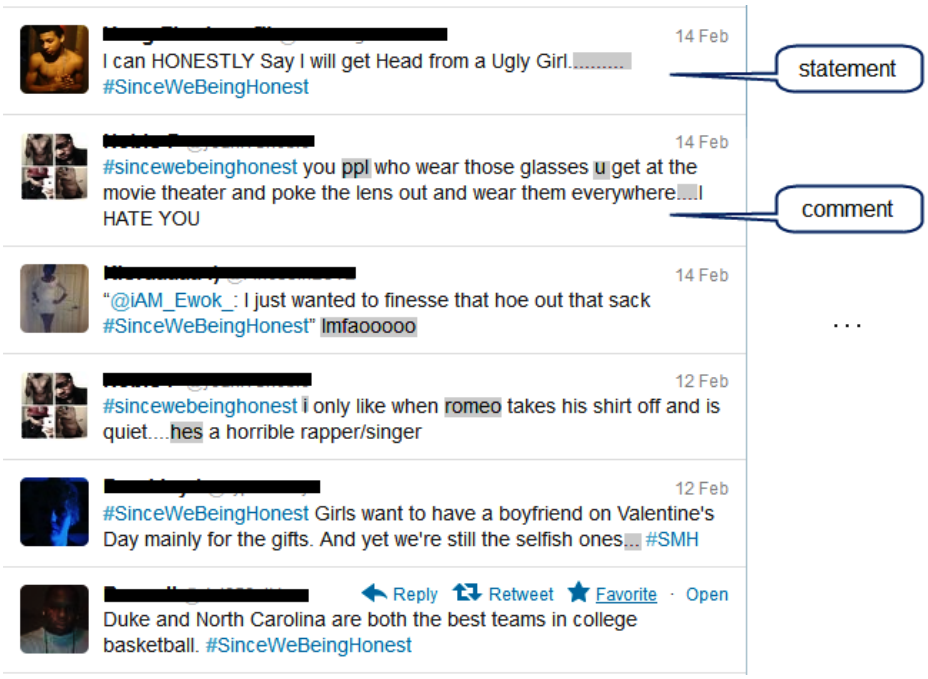

Figure 4.5: A Snapshot of *#sincewebeinghonest* Tweets

First, typical multi-document summarization tasks deal with dozens of documents each with several hundred words or several dozen sentences. By contrast, the tweets under a given topic usually number in the thousands, or tens of thousands, with each tweet being no more than 140 characters long. As is shown in Figure 4.5, a tweet consists of only one or two sentences.

Second, the language on Twitter is highly noisy, rife with nonstandard usage,

spelling and grammar mistakes, mixed symbols and characters, Netspeak expressions, etc., such as ……, *ppl*, *Imfaooooo*, *romeo*, *hes*, which are shaded in Figure 4.5.

Last but not the least, typical multi-document summarization tasks are targeted at closely related documents, but tweets under the same topic are only loosely lumped together, sharing not much in common. The tweets in Figure 4.5, for example, make a miscellany of girl, glasses, pop star, romance, and sports, though all belonging to *#sincewebeinghonest*. It would be infeasible to extract some tweets to make up a coherent summary.

Unfortunately, such Twitter-specific concerns have not been properly addressed in the limited literature on Twitter summarization. Sharifi et al. (2010a) find important phrases to be included in a summary with a graph-based algorithm and later (Sharifi et al., 2010b) develop a simpler "Hybrid TF-IDF" method, which ranks tweet sentences using the TF-IDF scheme and produces even better results. Liu et al. (2011) report a more complicated work using linked webpage content as external sources and extracting tweet sentences with Integer Linear Programming-based optimization. Those approaches are basically extractive summarization methods developed for general-purpose multi-document summarization with little or no adaptation to the special characteristics of Twitter text. According to their experimental reports, the information coverage of the output summaries is low and coherence is not even considered.

Different from those previous efforts, my work on Twitter topic

summarization is designed to overcome the difficulties caused by the Twitter idiosyncrasies and guided by a kind of deep content-driven coherence – speech act.

Rooted in half a century's linguistic study (Austin, 1962), speech acts capture the common grounds of tweets from a communicative perspective. When communicating with tweets, users may share information, ask questions, make suggestions, express sentiments, etc. which are all instances of "speech acts". Each tweet is associated with a type of speech act, like the "statement" and "comment" for the first two tweets in Figure 4.5.

In the following section, I will prove that using speech acts is suitable for coherence-targeted Twitter topic summarization because it enables us 1) to deal with a few clusters of communicatively similar tweets, each cluster being a speech act type, instead of a large medley of tweets; 2) to establish coherent connections between seemingly unrelated words and expressions; 3) to introduce a globally coherent structure by using speech act-based summary templates. In addition to generating coherent summaries with an abstractive summarization approach, speech acts also help to cover more useful information.

Users do not usually report the speech acts they are performing in Twittersphere as well as in face-to-face conversations. So before using speech acts for summarization, I will first discuss how to automatically recognize them in tweets (4.3.1). Then, guided by the recognized speech acts in the tweets, we can proceed to extract key words and phrases from the tweets (4.3.2). Leveraging

the linguistic knowledge of speech acts, I can generate abstractive summaries that integrate the extracted language materials into speech act-based sentence templates (4.3.3). Section 4.3.4 presents experimental results of evaluating the summaries in terms of both informativeness and coherence.

4.3.1 Speech Act Recognition in Twitter

This section presents my work on speech act recognition for Twitter text, as a prerequisite for speech act-guided key word/phrase extraction and summarization.

4.3.1.1 Types of Speech Act in Twitter

The scope of speech act recognition is based on Searle's (1975) popular taxonomy of speech acts: assertives (asserting something's being the case), commissives (committing the speaker to some future action), directives (getting the hearer to do something), declaratives (bringing about a different state of world by uttering something), and expressives (expressing the speaker's psychological state).

Table 4.11 lists the 5 speech act types I use, alongside the corresponding Searle's types and examples from the experimental datasets. A tweet belongs to one of 4 genuine types of speech act – **statement**, **question**, **suggestion**, **comment** – or the **miscellaneous** type. The choice stems from the fact that unlike face-to-face communication, twittering is more in a broadcasting style

than on a personal basis. Statement and comment correspond to Searle's assertives and expressives, which are usually intended to make one's knowledge, thought, and sentiment known. Searle's directives correspond to my question and suggestion, which are distinct speech acts targeted at other tweeters. Both commissives and declaratives are rare, as are other interpersonal speech acts such as "threat" or "thank". So they are all relegated to "miscellaneous".

Searle's Types	**My Types**	**Example Tweets**
Assertive	Statement	*Libya Releases 4 Times Journalists - http://www.photozz.com/?104k*
Directive	Question	*#sincewebeinghonest why u so obsessed with what me n her do?? Don't u got ya own man???? Oh wait.....*
	Suggestion	*RT @NaonkaMixon: I will donate 10 $ to the Red Cross Japan Earthquake fund for every person that retweets this! #PRAYFORJAPAN*
Expressive	Comment	*is enjoying this new season of #CelebrityApprentice.... Nikki Taylor = Yum!!*
Commissive	Miscellaneous	*65. I want to get married to someone i meet in highschool. #100factsaboutme*
Declarative		

Table 4.11: Searle's Speech Act Types and My Speech Act Types with Examples

Assuming one tweet demonstrates only one speech act type, speech act recognition in Twitter is a 5-class single-label classification problem. It is possible that one tweet demonstrates more than one speech act type. But given the short length of tweets, multi-speech act tweets are rare and my simplifying assumption is effective in reducing the complexity of the problem.

4.3.1.2 Feature Set Design

The feature sets used for recognizing the 5 types of speech act include word-based and character-based features.

- **Word-based Features**

There are two major types of 535 word-based features, all of which are binary-valued.

Cue Words and Phrases

Some speech acts are typically indicated by some cue words or phrases, such as *whether* for "question" and *could you please* for "suggestion". There are some manually compiled lexicons for speech act cues (Wierzbicka, 1987), but I refrain from using them for two reasons. First, the cue lexicons are very limited, consisting mostly of verbs. But words of other part of speech (including closed-class words) and phrases may be equally predictive. Second, such lexicons only serve standard English, not Twitter English rife with non-standard spellings, acronyms, and abbreviations. Therefore, I manually compiled a speech act cue lexicon of Twitter English from a dataset of 10K tweets, resulting in 531

ngrams (n = 1, 2, 3) for "statement", "question", "suggestion", or "comment". Table 4.12 shows some examples.

	Examples	**Total**
Unigrams	*know, hurray, omg, pls, why …*	268
Bigrams	*do it, i bet, ima need, you can …*	164
Trigrams	*?!?, heart goes out, rt if you …*	99

Table 4.12: Examples of Cue Words and Phrases

Non-cue Words

Some special words, though not intuitively cuing speech acts, may indirectly signal speech acts. I use four types of such non-cue words explained in the following.

Abbreviations and Acronyms: 1 feature indicates whether such shortened word forms appear. I collected the lexicon from online[14] and published (Crystal, 2006) resources, a total of 1153 words. Examples are *4ever* for "forever" and *tq* for "thank you". The shortened words are then restored to their original forms before I can extract the next two features: opinion words and vulgar words.

Opinion Words: 1 feature indicates whether opinion words appear. To judge opinion words, I used the SentiWordNet (Baccianella et al., 2010) and Wilson Lexicon (Wilson and Wiebe, 2003) widely used for opinion mining. As I am only

[14] http://www.chatslang.com

interested in strong opinion words, I build a lexicon by intersecting highly opinionated words (positive score + negative score ≥ 0.5, note that both positive score and negative score are non-negative) from the SentiWordNet with the "strong" words from the Wilson Lexicon, resulting in a total of 2460 words, like *shallow*, *vague*, *scary*, etc.

Vulgar Words: 1 feature indicates whether vulgar words appear. I used the API from an online resource[15] and collected 341 such words as *c**t and *f**k[16].

Emoticons: 1 feature indicates whether emoticons appear. I collected 276 emoticons from an online resource[17], such as O:) and *-*.

Character-based Features

There are 2 types of 8 character-based features, which indicate the frequency and position of special characters and are either binary- or ternary- valued.

Twitter-specific Symbols: I concentrate on the 3 symbols specific to Twitter: *#*, *@*, and *RT*. # is a hashtag marker often used in a mention of something to be stated about or commented on; @ is a prefix to a tweeter account, which tends to be associated with the more interpersonal speech acts of questions or suggestions; *RT* stands for "retweet" and its presence, especially in the initial position, strongly indicates a statement. Repeated use of them is an even stronger indicator of possible speech act types. Each of those symbols is associated with 2 features: 1 binary-valued feature indicating whether the symbol is in the initial position of a tweet and 1 ternary-valued feature indicating whether the symbol does not

[15] http://www.noswearing.com/dictionary
[16] For ethical concerns, I mask part of the words here and deliberately avoid using them in other examples.
[17] http://www.sharpened.net/emoticons/

appear (0), appears 1 or 2 times (1), or appears more than 2 times (2).

Indicative Punctuations: I single out 2 punctuations: *?* and *!* as the former often indicates a question and the latter is likely to indicate a comment or suggestion. Each of them is associated with 1 ternary-valued feature indicating zero appearance (0), 1 or 2 appearances (1), or 3 or more appearances (2).

The effectiveness of the proposed feature sets will be evaluated against the commonly used word features in 4.3.4.

4.3.2 Speech Act-guided Key Word/Phrase Extraction

The purpose of recognizing speech acts in Twitter text is to sort out the tweeted content for summary-worthy information. Among the 5 recognized speech acts, I focus on only 4 "genuine" types (statement, comment, suggestion, question) and extract key phrases and words from the tweets of major speech act types because they are representative of all communications under the topic. In the current design, I define "major speech act types" to be those covering at least 20% of all the topic tweets.

The introduction of speech acts facilitates a high-level and well-organized view of the tweets, i.e., whether most of them are about facts, opinions, suggestions, or questions. On this level, we can extract particular language expressions to convey the most salient information in a speech act, which would not be feasible with a more traditional framework working with salient terms, phrases, sentences, or tweets in general. Moreover, since the target key words

and phrases are about the same speech act, they are likely to be interconnected, thus enhancing content coherence.

4.3.2.1 Noise-resistant Phrase Extraction

To extract key words and phrases from the tweets of major speech act types, I first compile a stopword list to filter less informative words. Since general stopword lists (Salton, 1971) are targeted at standard English, it is augmented with Netspeak-style acronyms and abbreviations using free resources (Footnote 14). Then I can extract key words as frequent nonstop words. Extracting the key phrases is formulated as finding frequent ngram collocations.

Many approaches to collocation finding are based on statistical tests, such as t-test and chi-square test. I use likelihood ratio, a statistical test that gives the ratio of a non-collocation (word independence) likelihood to a collocation (word dependence) likelihood. It has been shown (Dunning, 1993) that likelihood ratio does not assume a normal distribution as t-test does and it is more appropriate for sparse data (e.g., text ngrams) than chi-square.

Regarding an ngram, for two hypotheses H_0 = the occurrences of the *n* words are independent and H_1 = the occurrences of the *n* words are dependent on each other, I use $L(H)$ to represent the likelihood and calculate $\log(L(H_0) / L(H_1))$. Likelihoods are calculated using n-nomial distribution and ngram probabilities are estimated using MLE. For each topic, I extract 50 top bigram phrases, 50 top trigram phrases, and as many "longer phrases" ($n > 3$) as possible with the

highest likelihood ratios. There are no more than 10 longer phrases in most cases and their length is typically 4, such as *Appeals Programme Illegal Arrest*.

The collocation-based phrase extraction is resistant to Twitter noise because noisy text by nature is accidental and un-conventionalized. Tweeters produce different kinds of noisy text so that a single noisy phrase hardly appears frequently enough to be extracted by my method. I manually checked 100 randomly sampled key phrases and confirmed that all of them are meaningful and noise-free.

4.3.2.2 POS-based Phrase/Word Patterns

Not all the extracted key words and phrases convey the most relevant information to a speech act. For example, statements are about facts, things, people, etc. and suggestions are about actions, activities, etc. Such information can be approximated by part-of-speech (POS) patterns for both words and phrases. Representative POS-based regular expression patterns are listed in the following, along with illustrative examples.

- The statement-relevant word is a noun, or '/N/' (e.g., *school*), phrase is a noun phrase, such as '/Adj/ /N/' (e.g., *high quality*) and '/Adj/ /N/ /N/' (e.g., *sexual abuse charges*).
- The comment-relevant POS patterns are like the statement-relevant ones. But comment phrases must have at least one opinion word (e.g., *good thing*)

judged from SentiWordNet (Baccianella et al., 2010) and the Wilson Lexicon (Wilson and Wiebe, 2003).

- The suggestion-relevant word is a verb, or '/V/' (e.g., *hate*), phrase is verb-centered[18], such as '/Adv/ /V/' (e.g., *truly wish*) and '/V/ /N/ /N/' (e.g., *sell health drugs*).
- The question-relevant word is either a verb or a noun, or (/'N'/ | /'V'/) (e.g., *reason*), phrase is either a noun phrase or a verb-centered phrase, such as '/Adj/ /N/ /N/' (e.g., *dirty ass mirror*).

The POS-based extraction is easy to implement and robust in the face of Twitter's noisy text – for which deep NLP such as syntactic or semantic parsing is not appropriate.

4.3.2.3 Phrase/Word Ranking

Among the speech act-relevant words and phrases (ngrams) I only select the most salient ones for a summary. In my work, "salience" is understood as a cumulative effect from an ngram network, i.e., a salient ngram co-occurs with other salient terms in the same tweet, which in turn boosts the salience of other ngrams it co-occurs with.

Let's construct a graph *G* for all the tweets of a major speech act type, using all the extracted ngrams (*Ng*) as vertices. Two vertices Ng_i and Ng_j are linked by

[18] It is so called to avoid being confused with the "verb phrase" in a syntactic sense, which is actually a kind of verb-centered phrase.

an edge if they co-occur in some tweet and the weight of the edge (w_{ij}) is the number of such co-occurrences. Note that *G* is undirected and I use *NB*(Ng_i) to denote the neighborhood of Ng_i. Then I can define the graph score of Ng_i, *GS*(Ng_i), as:

$$GS(Ng_i) = \frac{1-d}{|Ng|} + d \times \sum_{Ng_j \in NB(Ng_i)} \frac{GS(Ng_j) \times w_{ij}}{\sum_{Ng_k \in NB(Ng_j)} w_{kj}}$$

The calculation is iterated until convergence. As is the usual practice (Brin and Page, 1998), *d* is set to be 0.85. This formulation basically follows the TextRank algorithm (Mihalcea and Tarau, 2004) that can apply to summarization. But their graph vertices are all unigrams from which phrases are later assembled, whereas my *Ng* includes ngrams of different lengths, which are scored in one process.

Although the extracted phrases are noise-resistant, the same is not true about the extracted words as frequent unigram noises do exist. Moreover, phrases are more informative and less ambiguous than words (compare *school life* with *school* or *life*) and longer phrases are more so. Therefore I count the length *Ni* of Ng_i into its salience score *SS*(Ng_i), thus rewarding longer ngrams: $SS(Ng_i) = GS(Ng_i) \times N_i$ and rank all the phrases above all the words. Within all the phrases and all the words, rankings are determined by salience scores.

4.3.3 Abstractive Summarization for Twitter Topics

For a Twitter topic, the salient words/phrases extracted for its major speech act types as well as the topic itself are the building blocks of a summary. The

summary is abstractive in nature as proper words/phrases are to be filled in slots of a template specially designed to accommodate (English) speech acts and ensure summary coherence. In this section, I will first address the missing building block – topic words – which is nontrivial for hashtag topics. Then I will provide details of coherence-oriented template design and propose a novel summarization algorithm for Twitter topics.

4.3.3.1 Topic Processing

A Twitter topic itself is important information that should be included in the summary because it represents the common ground – sometimes the only common ground – shared by all its tweets. For a **regular topic** in words and phrases like *Space Shuttle*, the inclusion of topic words is straightforward and trivial. For a **hashtag topic** as # plus a concatenation of non-delimitated characters like #*justinbieber*, it is less so. I now describe how to split a hashtag into words.

To begin with, let's identify two major types of hashtags, those with mixed-case characters such as *#CyberMonday* and those with all lower-case characters like #*letsbehonest*. The first type resembles the "upper camel casing" naming convention familiar to programmers, which is easy to detect and split with a simple heuristic. To split the second type and sometimes the result after applying the heuristic to a mixed-case hashtag (e.g., #*PrayforRickRoss*), I rely on the mature statistical-based method successfully applied to other similar tasks

such as Chinese word segmentation (Wu and Tseng, 1993). To obtain ngram statistics, I use both unigrams and bigrams from all the tweets used in the experiments (100 regular topics + 100 hashtag topics, with 5000 tweets in each topic), totaling about 2GB text data and 2.3 million unigrams and bigrams.

After removing the #, consider every splitting $f = (w_1, w_2, \ldots w_n)$ of a hashtag by scoring it with ngram statistics: $score(f) = (s_{ug}(f) + \lambda s_{bg}(f)) \times lp(f)$ where $s_{ug} = \sum_{i=1}^{n} \log(P(w_i))$ and $s_{bg} = \sum_{i=1}^{n-1} \log(P(w_i w_{i+1}))$. They represent the unigram-based score and bigram-based score of f and λ determines the relative weight of bigrams. The probabilities $P(w_i)$ and $P(w_i w_{i+1})$ are estimated from the corpus using smoothed MLE. We penalize long words by $lp(f)$, which equals 1 if the average word length of f, $wl(f)$, is no more than r; otherwise $lp(f) = wl(f) / r$.

Suppose a hashtag H has m characters, H_k represents the first k characters of H and $Split(H_k)$ the best splitting of H_k. The hashtag splitting algorithm is based on dynamic programming and shown in Figure 4.6. Its time complexity is $O(m^2)$.

$Split(H_0)$ is empty; $Split(H_1)$ is H's first character itself;
For i = 2 to m
    For j = 0 to i – 1
        Calculate $score(f_j)$ where f_j is formed by $Split(H_j)$ and a "word" as the remaining part of H_i, with H_j removed;
    Choose the highest scoring f_j to be $Split(H_i)$;
Output $Split(H_m)$ i.e., $Split(H)$;

Figure 4.6: Splitting Algorithm for Hashtag Topics

I implemented the splitting algorithm on the 100 hashtags from the experimental dataset (λ = 0.01, $\varepsilon = 10^{-10}$, r = 5). The accuracy is 97%. In the only

3 hashtags not correctly split, one is an acronym hashtag (*#abdc*) that should be treated as a whole, and in the other two, one word is split into two (*lesson* → (*less*, *on*), *justin* → (*just*, *in*)).

4.3.3.2 Coherence-oriented Template Design

With the topic words and the salient words/phrases for each major speech act type, we can generate an abstractive summary by inserting them into proper slots of speech act-guided templates. In the current work, I aim at short (tweet-long) and coherent summaries, which can be conveniently expressed as sentences. So an apt template corresponds to a grammatical sentence, shown in the following.

For "<topic words>**", people** <verb frame> **"**<ngrams>**"{, (and)** <verb frame> **"**<ngrams>**"}*.**

Figure 4.7: Summary Template

In Figure 4.7, boldfaced words and punctuations are template constants and the angle brackets (< >) enclose template slots to be filled; (**and**) means the word **and** is optional; { }* means the enclosed part can appear zero or one or more times. The "topic words" are derived from the topic. For a regular topic, they are a direct copy; for a hashtag topic, they are the split result of the hashtag. The "ngrams" are the salient words/phrases extracted for the major speech act types. A "verb frame" is a verb or verb phrase specific to a particular speech act type.

For the 4 speech acts types used, I choose typical and short expressions from a lexicon of English speech act verbs (Wierzbicka, 1987) listed below.

Speech act type	**Verb frame**
Statement	*state*
Comment	*comment on*
Suggestion	*suggest*
Question	*ask about*

Table 4.13: Verb Frames for the Speech Act Types

It is easy to see that the template sentence is composed of the major speech acts as its predicates. The syntactic structure ensures that related information is well clustered and governed by the same verb frame. On the sentence level, information about different speech acts resides within the boundary of different speech acts, mimicking the style of human writing. This is how I achieve overall coherence with the help of speech acts.

For coherence and grammaticality purposes, the verb frames are designed to agree with the POS patterns of the succeeding words/phrases to form grammatical constructions such as *state NP*. Problems arise with *ask about* because what follows may not be a verb (phrase) and even so the verb may not be gerundive, which is also a problem for *suggest*. To alleviate such problems, I introduce quotation marks (“ ”) around the succeeding words/phrases so that the

sentence generally becomes more readable (compare *... suggest do your homework* and *... suggest "do your homework"*).

4.3.3.3 Summarization Algorithm

Each <verb frame> **"**<ngrams>**"** clause in the template represents the salient information about one speech act. I first decide the specific verb frames according to all the major speech act types and order them in the template according to the number of tweets with the speech acts. For example, if a topic has only two major speech act types: "statement" and "comment" with 2000 and 2500 tweets respectively, the template is "For …, … people comment on …, and state …"

The next step is to derive the ngrams needed for the template. I select the ngrams belonging to different speech acts in a round-robin fashion. Starting from the first speech act type as reflected in the order of the verb frames in the template, let's select the top-ranking ngrams to fill in template slots. After the last speech act type is processed and if the summary length limit is not reached, we loop back to the first speech act type. The detailed algorithm is shown in Figure 4.8.

```
Repeat
For each speech act in the template order
    Select the top-ranking Ng* from all the ngrams extracted for that speech
    act;
        If Ng* is a unigram
            Skip to the next speech act unless all longer ngrams (length ≥ 2)
            for all speech acts have been selected;
        If Ng* is not redundant and summary length permits
            Fill a template slot with Ng*;
        Else
            Remove Ng*;
Until summary length is reached;
```

Figure 4.8: Ngrams Selection Algorithm

The algorithm consistently favors longer ngrams so that the generated summary contains informative and less ambiguous phrases. As in multi-document summarization in general, information redundancy should be avoided (Carbonell and Goldstein, 1998). Ngram redundancy is determined by comparing its words with each of the selected ngrams as well as the topic words. Suppose Ng_0 is selected and Ng_1 is under consideration, we use $W(Ng_0)$ to denote the word set of Ng_0 and decide Ng_1 is redundant if $\frac{|W(Ng_0) \cap W(Ng_1)|}{|W(Ng_0) \cup W(Ng_1)|} \geq \theta$. θ is 0.35 in the experiment.

Note that the template-based approach allows character-level length control. So unlike truncation methods that may leave the last sentence unfinished or last word incomplete, my summarization algorithm guarantees the completeness and readability of the generated summaries.

Now let's look at an example of the hashtag topic "#100factsaboutme". The two major speech act types are statement and comment. Predicted by the trained model, there are more statement tweets than comment tweets, so the template is:

For …, people state … and comment on …

The ngrams collected for each of them are:

Statement: {"fridge door", "high school", "fluent sarcasm", "knowledge success", "gusta el", "middle school", "real life", … }

Comment: {"hate school", "love", "people", "good", "feel", …}

We then select ngrams for the speech acts according to the algorithm in Figure 4.8. First we select "fridge door" because it is the first long ngram for statement, and then we move to comment and select "hate school", its only long ngram. In the second round, we select "high school" for statement, but we skip "love" for comment because it is not a long ngram and we have not run out of all long ngrams for both speech acts. For the same reason, in the next rounds we

select “fluent sarcasm”, “knowledge success” for statement but nothing for comment until we reach the length limit (100 words). The final summary is:

For "100 facts about me", people state "fridge door, high school, Fluent Sarcasm, Knowledge Success, gusta el" and comment on "hate school".

4.3.4 Experimental Results

In this part, I will report the experimental results on two evaluation tasks – speech act recognition and summarization – on different datasets.

4.3.4.1 Data Preparation

According to my experimental design, Twitter speech act recognition is evaluated on a relatively small dataset of tweets annotated with speech acts. Twitter summarization is evaluated on two much larger datasets.

- **Speech Act Recognition in Twitter**

Using the Twitter search API, I collected tweets of 6 randomly chosen trending topics on *Twitter.com* from March 1, 2011 to March 31, 2011. The topics fall into three categories – News, Entity, Long-standing Topic (LST) – that correspond to the three “topic types” (Zhao and Jiang, 2011). I manually annotated all the 8613 tweets as one of Sta (statement), Que (question), Sug (suggestion), Com (comment), or Mis (Miscellaneous). The categories, topics

and tweet numbers are shown in Table 4.13.

Category	Topic	# Tweets
News	*Japan Earthquake*	1742
	Libya Releases	1408
Entity	*Dallas Lovato*	677
	Nikki Taylor	786
LST	*#100factsaboutme*	2000
	#sincewebeinghonest	2000

Table 4.14: Details of Experimental Datasets

Different categories/topics of tweets have different speech act distributions. Figure 4.9 illustrates the speech act distributions in all the 6 topics used. Obviously, statements and comments take the majority. Generally speaking, entity topics are dominated by comments and news topics by statements. Special cases also exist, such as “Japan Earthquake” containing a considerable proportion of suggestions (e.g., about what people can do to help victims). The imbalanced distribution of speech act types in Twitter topics justifies the design of our summarization algorithm.

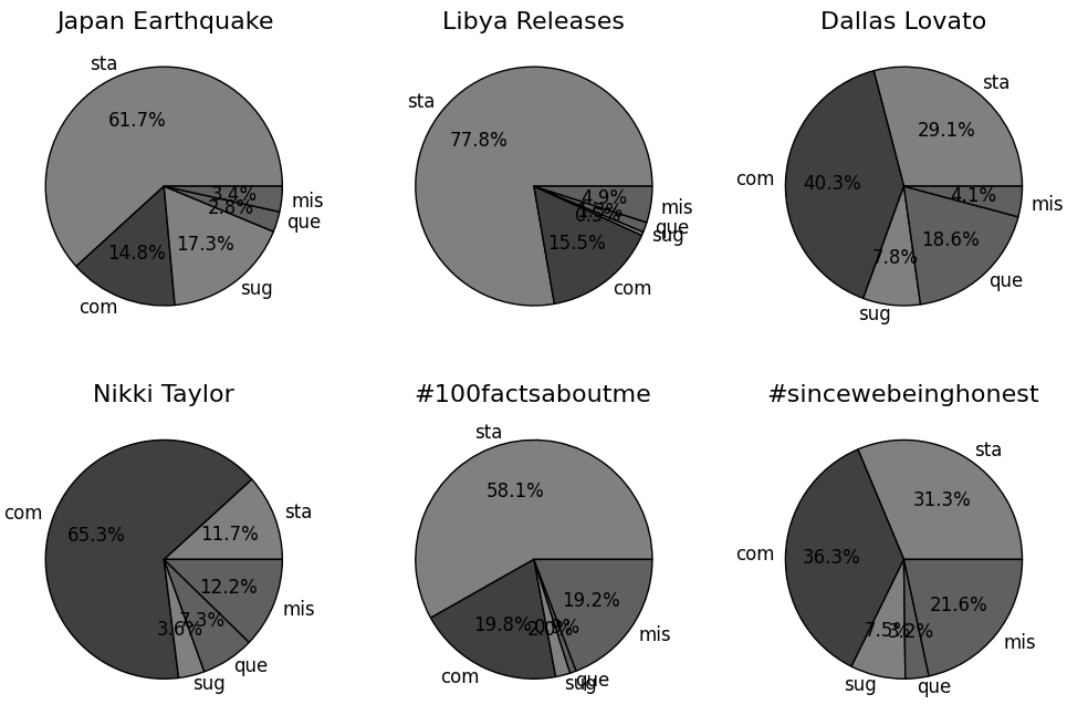

Figure 4.9: Speech Act Distributions in the 6 Twitter Topics

- **Twitter Topic Summarization**

For this task, I used trending topic tweets over a one-year period from March 1, 2011 to February 29, 2012. From those topics I construct two datasets: one for regular topics and the other for hashtag topics, each with 100 trending topics covering a variety of categories. Regular topics include news (e.g., *Frankfurt Airport*), entertainment (e.g., *Grammys*), celebrities (e.g., *Jeremy Lin*), technology (e.g., *Android 5.0*), social life (e.g., *Earth Hour*), etc. Hashtag topics include personal life (e.g., *#oomf*), chitchat (e.g., *#idontunderstandwhy*), social life (e.g., *#teaparty*), entertainment (e.g, *#idol*), etc. For each topic I collect up to 5000 distinct tweets (with unique tweet IDs), with a total of 1 million tweets. The total number of summaries to be evaluated is 200.

As gold standard reference, the human summaries were collected from two public services: *www.whatthetrend.com* and *tagdef.com*. The former asks users to explain why a topic is trending and the latter, dedicated only to hashtag topics, asks users to define a hashtag topic. The explanations or definitions are required to be short[19] and informative, thus good surrogates for "summaries" in the lack of authentic summaries.

We can find a short explanation or definition of all the 100 regular topics (on *whatthetrend.com* only) and the 100 hashtag topics (on either *whatthetrend.com* or *tagdef.com*), and usually there are multiple versions on one service. Fortunately, both services provide peer check mechanisms to help us choose the best version. *Whatthetrend.com* allows users to verify the posted explanations. *Tagdef.com* employs a voting scheme, allowing users to vote for ("upvotes") or against ("downvotes") a definition. Then a score can be calculated (= number of upvotes – number of downvotes) to indicate the quality of the definition. I choose the summary with the highest score (for *tagdef.com*) or the most recently verified (for *whatthetrend.com*) that fits the time span of the collected tweets. If none of the version is peer-checked, I simply choose the one that best fits the time span. For a regular topic, I can only choose among the versions from the *whatthetrend.com* source. A hashtag topic summary may come from one or two sources. If both sources provide a candidate summary for a hashtag topic and only one is peer-checked, that becomes the human summary. Otherwise I choose

[19] *Whatthetrend.com* limits the length to 140 characters and *tagdef.com* has similar requirement.

the one that best fits the time span.

4.3.4.2 Evaluation of Speech Act Recognition

The raw Twitter text data were lightly preprocessed and the features were extracted by regular expression patterns. I did two sets of experiments. In the first set, I classified tweets in each topic using different feature sets. The classifier is SVM with a linear kernel. Since SVM inherently does binary classification, the multi-class case is handled by the one-vs-all paradigm. In the second set, I applied the best feature set from the previous results to three datasets at different levels.

For all classification tasks, I will report the F1 (the harmonic mean of precision and recall) scores from ten-fold cross validation.

- **Comparison of Feature Sets**

To find out what features are useful, I experimented with cue words, non-cue features, symbols (character-based features), and all the proposed (combined, i.e., cue + non-cue + symbols) features. I also used the commonly adopted bag-of-words (BOW) features for comparison. After removing words that occur only once, I come up with a total of 4421 words as BOW features.

Table 4.15 lists the F1 scores on each speech act type with different feature sets, as weighted averages of the 6 topics according to tweet numbers. The "AVG" is the weighted average according to the number of each speech act type.

Feature set	# Features	Sta	Que	Sug	Com	Mis	AVG
Cue	531	0.788	0.455	0.554	0.623	0.422	0.668
Non-cue	4	0.671	0.088	0.068	0.355	0.074	0.447
Symbols	8	0.681	0.473	0.039	0.412	0.097	0.483
Combined	543	0.798	0.597	0.564	0.670	0.446	0.695
BOW	4421	0.788	0.430	0.533	0.620	0.486	0.673

Table 4.15: F1 Scores for Different Feature Sets

Among my proposed feature sets, cue words and phrases are the best overall. On individual speech acts, it defeats non-cue words and symbols with the only exception of "questions" because the punctuation *?* is a more reliable indicator of questions than question cue words. Character-based features (symbols) outperform non-cue features in almost all columns (with the only exception of "suggestion") and occasionally defeat cue features for reasons explained. Since the non-cue features are meta-features derived from non-cue words bearing the characteristics of cyber English, they are less capable of capturing speech act regularities in Twitter than special symbols. Such evidence also shows that the Twitter text has a distinct style and not all purported "noises" are noisy (e.g., *?!?).*

Without exception, using all our proposed feature sets achieves better performance than using any feature set alone. The combined feature set also defeats the much larger BOW feature set. With a small fixed size, the combined

feature set promises good scalability. It will be my choice in subsequent experiments.

For individual speech act types, statements and comments are better recognized than questions and suggestions, partly attributable to the difference in training data amount. Unsurprisingly, the recognition of "miscellaneous" is the worst using our features because the proposed features are aimed to capture the textual characteristics of speech acts, which do not exist in a heterogeneous group made up of different speech act types and non-speech acts. Note that the inferiority in this "speech act" has no adverse effect on summarization based on recognized speech acts since no useful information will be derived from it.

- **Comparison of Speech Act Recognition on Different Levels of Dataset**

It is interesting to find out a desirable level to perform this task on – topic-level, category-level, or Twittersphere-level. A higher level is desirable because that means we don't have to prepare training data for specific topics or categories, thus simplifying the building of practical systems and saving much annotation labor. Drawing on the previous empirical results, I performed speech act recognition using the combined feature set on the three levels of datasets, with results summarized in Table 4.16.

Level of Dataset	Sta	Que	Sug	Com	Mis	AVG
Topic	0.798	0.597	0.564	0.670	0.446	0.695
Category	0.673	0.705	0.581	0.629	0.335	0.673
Twittersphere	0.770	0.636	0.577	0.612	0.209	0.639

Table 4.16: Weighted Average F1 Scores on Three Levels of Datasets

The average score for all speech act types on the category level or Twittersphere level is not much worse than that on the topic level, degrading by only 3% or 8%. The scores on “questions” and “suggestions” are even higher on the category and Twittersphere levels, suggesting that merging data from different topics or categories helps to capture more characteristics of those speech acts. Degradation for “miscellaneous” is attributable to the reasons explained before. But no harm from the “miscellaneous” failure will be inflicted on summarization.

Those evidences enable us to recognize speech acts in Twitter on the most general Twittersphere level, without substantial loss in classification performance, with the benefit of using all our annotated data (over 8000 tweets) and obviating the effort to determine the content domain of unseen data.

4.3.4.3 Evaluation of Twitter Topic Summarization

To evaluate output summaries, as in speech act recognition evaluation, no text cleaning or normalization is done for the raw tweets. The only NLP tool used

is a POS tagger trained on tweet data (Gimpel et al., 2011). The summarization work relies on speech acts recognized by a model trained on all the annotated tweets, using the optimal features according to the previous empirical results.

- **Automatic Evaluation**

For comparison, I generate peer summaries of two kinds. The first is by SumBasic, a simple but very robust extractive summarizer for generic documents (Nenkova and Vanderwende, 2005). The second is by "Hybrid TF-IDF" (Sharifi et al., 2010b) that ranks tweet sentences by the normalized TF-IDF of their words, a simple system that reportedly defeats MEAD, LexRank, and TextRank for Twitter topic summarization (Inouye, 2010). To ensure fairness, all automatic summaries are no more than a tweet long (≤ 140 characters), as are the human summaries.

For automatic evaluation, I use the popular ROUGE metric as in 4.2.4.4. Tables 4.17 and 4.18 report the average ROUGE-1, ROUGE-2, and ROUGE-SU4 F scores for regular topics and hashtag topics respectively. Each score is accompanied by the 99% confidence interval calculated by the ROUGE tool (Lin, 2004). Statistical significance ($p < 0.01$) under the paired t-test between the peer methods and my method is marked by *.

	ROUGE-1	*ROUGE-2*	*ROUGE-SU4*
Our method	**0.1903** (*0.1642 - 0.2191*)	**0.0588** (*0.0438 - 0.0746*)	**0.0555** (*0.0444 - 0.0661*)
SumBasic	*0.1332 (*0.1114 - 0.1541*)	*0.0440 (*0.0310 - 0.0576*)	*0.0419 (*0.0322 - 0.0527*)
Hybrid TF-IDF	*0.1613 (*0.1353 - 0.1919*)	0.0558 (*0.0386 - 0.0776*)	0.0539 (*0.0399 - 0.0723*)

Table 4.17: Rouge F Scores for the Regular Topics

	ROUGE-1	*ROUGE-2*	*ROUGE-SU4*
Our method	**0.1269** (*0.1039 - 0.1511*)	**0.0357** (*0.0228 - 0.0486*)	**0.0380** (*0.0282 - 0.0482*)
SumBasic	*0.0659 (*0.0457 - 0.0863*)	*0.0074 (*0.0013 - 0.0168*)	*0.0170 (*0.0103 - 0.0249*)
Hybrid TF-IDF	*0.0673 (*0.0473 - 0.0881*)	*0.0134 (*0.0039 - 0.0253*)	*0.0193 (*0.0117 - 0.0286*)

Table 4.18: Rouge F Scores for the Hashtag Topics

Obviously, the proposed method leads in all ROUGE measures on both types of topics. Consistent with the results reported in previous work (Liu et al., 2011), regular topic summaries are much better than hashtag topic summaries. Also note that it is on the hashtag topic summaries that our proposed method

more markedly excels, significantly outperforming SumBasic and Hybrid TF-IDF.

- **Manual Evaluation**

The ROUGE scores cannot evaluate summary coherence, and a closer inspection of the summaries reveals that the abstractive summaries guided by speech acts are not only more coherent but also more likely to capture key words or phrases in human summaries than the extractive summaries, which are vulnerable to spam, redundancy, and other noisiness. Table 4.19 shows the human and automatic summaries for the hashtag topic *#agoodboyfriend*.

Human	*People are tweeting the qualities that make a good boyfriend and the things a good boyfriend does.*
Proposed method	*For "a good boyfriend", people state "Team Minaj, DAMN Derrick Rose, Yuri Gagarin" and comment on "love joy, silent cries, good girlfriend".*
SumBasic	*#agoodboyfriend is #agoodboyfriend whether he's around u or not.. "#AGoodBoyfriend" is really a TT ? #agoodboyfriend is not looking for #ago*
Hybrid TF-IDF	*RT @DamnItsTrue: GREAT LIFE = Good Friends Good Food Good Song #agoodboyfriend #DamnItsTrue @DamnItsTrue: GREAT LIFE = Good Friends +*

Table 4.19: Human and Automatic Summaries for *#agoodboyfriend*

In addition to key word overlapping (“people”, “good”, “boyfriend”), our abstractive summary are structurally similar to the human summary (“people are tweeting …” vs. “people state …”) and both are expressed in complete sentences. On the contrary, the two extractive summaries are only incoherent combinations of tweets or tweet fragments. In addition, our abstractive summary seems to include much more useful information than the extractive summaries. I suspect this is a bonus of our approach, one that cannot be directly measured by ROUGE because automatic summaries and human summaries seem to include different kinds of content. So I will also do manual evaluations on both **informativeness** and **coherence**, which are generally accepted yardsticks for a summary’s content and form.

Two human judges were trained to score the summaries according to their informativeness and coherence on a scale of 5 points. The higher the score, the more explanatory / informative / readable a summary is. Each judge was required to score all the human and automatic summaries, totaling $100 \times 4 \times 2 = 800$ summaries. For each topic, they were presented the summaries in a random order so that no pattern could be detected. For each scoring category, Cohen’s Kappa ranges between 0.5 and 0.7, indicating good inter-judge agreement. Tables 4.20 and 4.21 sum up the results on the regular and hashtag topics by averaging the human scores over the 100 topics. Statistical significance of the summaries generated by the proposed method against all the other summaries is indicated by * ($p < 0.001$) on a paired two-tailed t-test.

	Informativeness	**Coherence**
Human	3.84	*4.71
Proposed method	**3.78**	**4.01**
SumBasic	*1.88	*2.60
Hybrid TF-IDF	*1.95	*2.25

Table 4.20: Average Human Scores for the Regular Topics

	Informativeness	**Coherence**
Human	3.26	*4.63
Proposed method	**3.19**	**3.61**
SumBasic	*1.94	*2.55
Hybrid TF-IDF	*2.17	*2.65

Table 4.21: Average Human Scores for the Hashtag Topics

The statistics show that the summaries generated with the proposed method are comparable to human summaries in terms of informativeness, significantly outperforming SumBasic and Hybrid TF-IDF by a large margin. The same is also true for coherence, showing the superiority of abstractive summarization with the coherence-oriented template design.

But our summaries are also significantly less coherent than human writings, mainly because of the lack of coherence between the extracted key words and phrases for the same speech act. Incorporating the contexts of key words and

phrases during their extraction may be a promising solution, which I will explore in future work.

4.4 Chapter Summary

This chapter proceeds from shallow content-driven coherence to deep content-driven coherence in summarization. Showing that coherence is not limited to shallow content like words and entities, my work complements the mainstream research that focuses on shallow content. Coherence driven by deep content has a more profound impact on summarization, serving sentence selection as well as sentence ordering, abstractive summarization as well as extractive summarization.

Among the various forms deep content can take, I explore two of them – genre-specific aspects applied to news documents and speech acts applied to Twitter posts – and show their potential in enhancing summary coherence.

The use of news aspect is inspired by the TAC 2010 guided summarization task. Due to the nature of aspects and their natural or logical relations, an aspect-guided summary holds the promise of content-level coherence. In an extractive style, aspect-guided summarization relies on sentence level aspect recognition and HMM-based modeling with aspect information.

The use of speech acts is based on the communicative nature of Twitter messaging. Speech acts provide a bird's-eye-view of the communicated messages on Twitter and help to structure a Twitter topic summary in a coherent way. The

summaries are abstractive in nature, drawing on speech act recognition and key word/phrase extraction. The extracted key terms are ranked to fill in coherence-oriented summary templates using a round-robin algorithm.

Both news aspects and speech acts prove valuable in coherence-targeted newswire summarization or Twitter topic summarization. Unlike shallow content, such deep content is usually hidden or implied. Therefore their automatic recognition is an indispensable subtask, requiring carefully designed learning schemes and manually annotated data. In this respect my original work on aspect and speech act recognition has also made solid contributions.

Chapter 5: Cognitive Model-driven Coherence in Summarization

So far I have explored coherence in summarization from the perspective of textual content. An alternative is to take the perspective of summary reader and model coherence in human terms. Since ultimately, whether a summary is coherent or not is decided by its reader, modeling coherence on the human cognitive basis seems a reasonable choice – although such endeavors are seldom reported in the summarization community.

The best theories to account for the human mechanism of coherence lie outside the realm of computational science or artificial intelligence. Coherence is, for cognitive psychologists, concomitant with text comprehension which is intensively studied to understand human cognition. According to many theories and models of cognitive psychology (Tapiero, 2000; van Dijk and Kintsch, 1983; Gernsbacher, 1996; Kintsch, 1988, 1998; van den Broek et al., 1996; Zwaan et al., 1995), a coherent representation is required for text comprehension. In order to make sense of a text, readers must establish coherent relations between textual units. Therefore, coherence and text comprehension are the two sides of the same coin.

In this chapter, I will first build a computational model based on a popular cognitive model (Kintsch, 1998) of narrative text comprehension, establishing the computational counterparts in the model's cognitive process. Coherence is an

underlying constituent of the model, which is then used to summarize narrative text – the genre where the cognitive model works best. Coherence is deeply involved in such kind of summarization – including content selection and sentence realization, a point that will be validated by experiments on event-centric news and fairy tales, both typical instances of narrative text.

In 5.1, I will computerize a cognitive model of narrative text comprehension with all the technical details. In 5.2, the cognitive model-driven coherence will be used to summarize narrative text, where propositions instead of sentences will be taken as the basic processing units. Section 5.3 presents the experimental results on two kinds of narrative text. The highlights of the chapter are wrapped up in 5.4.

5.1 Cognitive Model of Narrative Comprehension and Coherence

Cognitive models of text comprehension and coherence usually focus on narrative text, which are typically rich with events and actions about related characters, because coherence is of greater importance to understanding narrative text than expository or argumentative text.

When reading a typical expository article such as a biography, we can choose to read only the parts that interest us (e.g., birth place, education, marriage) and the lack of coherence between the chosen parts does not affect our understanding of the person. When reading a typical argumentative article such

as a scientific thesis, we can focus on only particular sections to get the *method*, *result*, *conclusion*, etc. to understand the topic despite the lack of global coherence. What about reading a typical narrative article such as a story? Reading only parts of the story disrupts the development of plot and renders an incoherent representation of the characters, their relations, and events in our mind, which prevents us from understanding it.

Our comprehension of a narrative changes constantly at different stages of reading. A new event or the appearance of a new character will alter our mental representation of it as we seek to establish **meaningful or coherent links** between the new elements and the old ones. That is why many cognitive psychologists regard the process of text comprehension as guided by the mechanism for establishing and preserving coherence, such as the **CI (Construction-Integration) model** by Kintsch (1998), the Structure Building Framework by Gernsbacher (1991, 1996) and the Landscape model by van den Broek et al. (1996).

Those models are similar in that they model coherence establishment for a narrative as a dynamic process. A new textual unit (e.g., word) activates related information in the long term memory, and then a cognitive mechanism selects information that is most relevant to the current mental representation of the narrative. The textual units are linearly processed so that the mental representation is constantly updated. As narratives are typically about characters, their relations, and happenings around them, the propositional representation

with a predicate-argument structure is often adopted by such models for capturing such narrative elements that make up a whole plot. It also turns out that proposition can be a good computational unit when modeling cognitive model-driven coherence.

Those models differ from each other only in minor details and among them, Kintsch's (1998) CI model is the best developed in both cognitive and computational terms. It lays emphasis on proposition-based word activation from the long term memory (**construction**) and a spreading activation process of strengthening or inhibiting the activated words (**integration**). Lemaire et al. (2006) add some computational details and implement the CI model on the basis of a human memory model. The recent extension of the CI model, CI-2 (Kintsch and Mangalath, 2011), employs a dual-memory model that highlights the role of the explicit context of words. My cognitive model to account for narrative text comprehension and coherence is built on those previous works.

5.1.1 An Overview of the Model

Figure 5.1 illustrates the overall architecture of my model. The three blocks – **Long Term Memory**, **Working Memory**, and **Episodic Memory** – are based on the popular theory about human memory composition. The solid-line arrows mark major operations between and within the various text representations in the memory parts, among which **Association** and **Spreading Activation** are derived from the CI model. The dashed-line arrows represent influences from contextual

(input text) or general semantic (long term memory) sources.

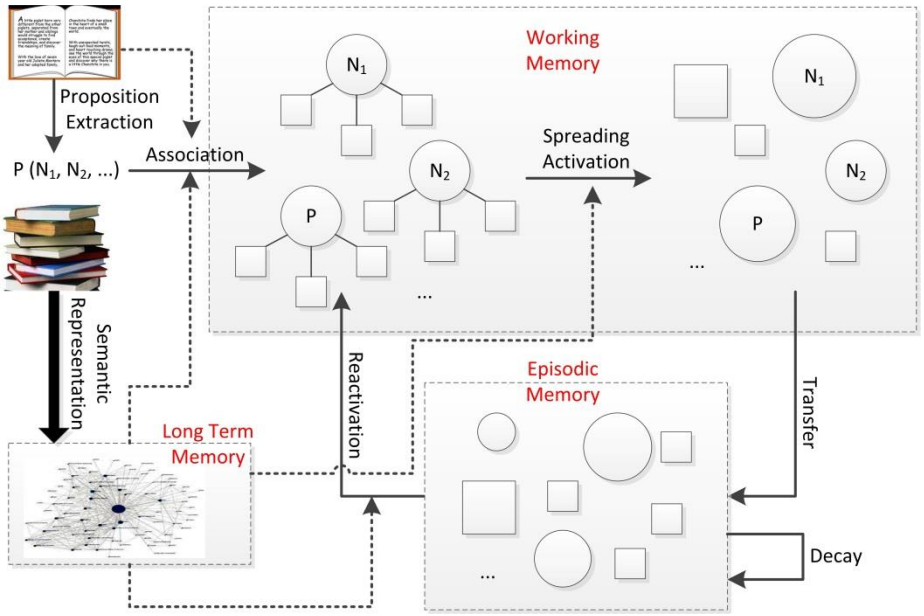

Figure 5.1: Architecture of the Narrative Text Comprehension/Coherence Model

The model starts with the narrative text, indicated by the icon in the upper-left corner of Figure 5.1. The whole text is segmented into sentences, from which propositions are extracted. We read a story sentence by sentence and understand the plot proposition by proposition, which is modeled as a cyclic process. In each reading cycle, the model receives the current proposition with all its elements as input.

Independent of the narrative text is our general knowledge about relations between words, i.e., a semantic network that results from years of language contact such as reading. Its computational analogue is a word vectorial space computed from a corpus. Stored in the long-term memory, the semantic network

is closely tied with the proposition-based comprehension.

At the beginning of each new reading cycle, the predicate (*P*) and nouns (*N*) in the current proposition is each associated with a number of closest words from the semantic network with cues from the narrative context. Each proposition element and associated word has an activation value and each pair of them has an association value, all computed from the semantic network. In the figure, circles represent words from the narrative text and squares represent associated words. Size indicates activation value.

In the next step, the proposition elements and associated words are reassigned activation values via a spreading activation algorithm. Its cognitive analogue is the stabilization of the activation degrees of all related words in the working memory.

Since the working memory has a limited capacity (Just and Carpenter, 1992), the words with their stabilized activation values in the working memory are **transferred** to the episodic memory that stores all the activated words during a narrative's comprehension. As reading proceeds, a human reader tends to gradually forget narrative elements in earlier sentences/propositions, which is modeled by a **decay** process of all the stored words for each reading cycle.

If sufficiently activated (after decay) and sufficiently close (computed from the semantic network) to the elements in a proposition being processed, a word in the episodic memory will be **reactivated** back into the working memory. After all reading cycles are completed, the episodic memory contains all the narrative

element words and their associates with their final activation values. This is also the mental representation of the narrative text according to my model.

In the following two sections, I provide further details of the two main modules of the model: semantic network construction in the long term memory and the proposition-based cyclic comprehension.

5.1.2 Semantic Network in Long Term Memory

A semantic network is supposed to be built on a large corpus and used to decide how semantically close two words are. Kintsch (1998) first applied Latent Semantic Analysis (LSA) to a specialized corpus and built a semantic space crucial to his CI model. In this section, I will discuss the use of different kinds of corpus and alternative ways to construct a semantic network, which have not been explored before.

5.1.2.1 Specialized Corpus and Wiki Corpus

To endow the computer with language experiences comparable to a human reader, we need to prepare a corpus as input to a semantic model. In the literature, a popular choice is the TASA corpus consisting of educational texts for American school students of different grade levels (Quesada, 2007), which contains over 44 thousand documents and 11 million word tokens.

In the current work, I experiment with two kinds of narrative text – event-centric news and fairy tales – and use two specialized corpora accordingly.

The first is the Reuters-21578 benchmark (Reuters) corpus and the second is a freely available 453-story fairy tale (FT) corpus (Lobo and de Matos, 2010). In addition, I use a Wiki corpus from the English Wikipedia articles[20], which is much larger and more generic than TASA.

The details of the above mentioned corpora are listed in Table 5.1. By using both a highly generic and two highly specialized corpora, I intend to study the influence of different kinds of corpus on a cognitive model, which has not been reported to the best of my knowledge.

	Wiki	**TASA**	**Reuters**	**FT**
# documents	3.6M	44k	21578	453
# words	> 2G	11M	3.5M	908k
Degree of Specialization	**Highly generic**	Moderately generic	**Highly specialized**	**Highly specialized**

Table 5.1: Comparison of Corpora

5.1.2.2 Semantic Modeling with Standard LSA/LDA

When constructing a semantic network out of a corpus, we will essentially compute word similarities based on word distributional and co-occurrence patterns in documents. LSA and Latent Dirichlet Allocation (LDA) are two appropriate tools for this purpose.

[20] I use the 20110317 .bz2 dump for our experiment.

LSA (Landauer et al., 1998) uses a term-by-document matrix *A* as word co-occurrence evidence and applies Singular Value Decomposition (SVD) to it so that $A \overset{SVD}{=} USV^T$, where *S* is a diagonal matrix composed of singular values of *A*, *U* and *V* are composed of the corresponding left singular and right singular vectors. Then we take the *k* largest singular values of *S* to get a lower-rank approximation of *A*: $U_k S_k V_k^T$, a dense matrix representing a semantic space. For two words *i* and *j* in this space, we calculate the cosine similarity of their corresponding vectors:

$$Sim(i, j) = Cosine(u_{i,*} S_k, u_{j,*} S_k)$$

where $u_{i,*}$ is the *i*th row vector of U_k.

LDA (Blei et al., 2003) is an alternative model that introduces topic and probability distributions to the observed word co-occurrence pattern. It assumes multinomial distributions for both document-over-topic and topic-over-word distributions with Dirichlet priors. The model parameters can be learned from Bayesian inference such as variational Bayes (Blei et al., 2003) from which we can derive all the posterior topic distributions on word $P(z_n | w)$, $n = 1, 2, \ldots, t$. These *t* probabilities make up a vector for *w*, based on which we can calculate word similarities as vector cosines.

The standard LSA and LDA described above share a common limitation. Once constructed, the LSA/LDA model is fixed. Updating with new documents would mean starting from scratch. This is computationally thwarting because fitting millions of documents (for Wiki) in memory all at once is impractical.

Instead, we would rather send smaller-sized batches of documents and update the trained model continuously. On the other hand, a "fixed" semantic network does not accord with the fact that the human long-term memory is constantly updated with new information from her cognitive environment.

For both computational and cognitive reasons, I will use the updatable variants of LSA/LDA.

5.1.2.3 Semantic Modeling with Updatable LSA/LDA

Distributed LSA (Řehůřek, 2011) is a solution to LSA updating. For the input matrix $A^{m\times n}$ with a large n (number of documents), we partition it into smaller submatrices $[A^{m\times c_1}, A^{m\times c_2}, ..., A^{m\times c_k}]$ where $\sum_{i=1}^{k} c_i = n$. Then for any two such submatrices A_1 and A_2, after SVD and k-dimensionality reduction, $A_1 \overset{SVD^k}{=} U_1 S_1 V_1^T = U_1 S_1^2 U_1^T$, $A_2 \overset{SVD^k}{=} U_2 S_2 V_2^T = U_2 S_2^2 U_2^T$. To merge ($U_1$, S_1) and (U_2, S_2) into (U, S) for [A_1, A_2], we can apply QR decomposition on [$U_1 S_1$, $U_2 S_2$] and get an orthonormal matrix Q with the same span of [U_1, U_2]. Another SVD is then applied to R so that $R \overset{SVD^k}{=} U_R S V_R^T$, S is now diagonal and $U = QU_R$. See (Řehůřek, 2011) for more technical details.

A successful updatable variant of LDA is the online LDA (Hoffman et al., 2010). It is based on batch variational Bayes to fit the parameters $\boldsymbol{\lambda}$ to the variational posterior over the topic distributions with an expectation-maximization (EM) algorithm. In the E-step, the algorithm holds $\boldsymbol{\lambda}$ fixed and fits the per-document variational parameters γ and θ with a new

document. In the M-step, λ is updated by λ', an optimal setting if the whole corpus is a simple repetition of the new document. Hoffman et al. (2010) prove that online LDA converges fast and performs well.

Using distributed LSA and online LDA[21], I can handle a large corpus like Wiki and build a semantic network with the potential of being updated with new knowledge sources.

5.1.3 Proposition-based Cyclic Text Comprehension

Motivated by psychological theories of human memory (Anderson, 1976) and cognitive models of text comprehension (Kintsch, 1998), my computational model of story comprehension simulates the human reading process with coherence as an underlying theme. The whole reading process is a cyclic one and in each cycle, a new proposition is processed and the text representation updated in different parts of human memory. In the following, I provide the details of model components before showing a complete algorithm.

5.1.3.1 Proposition Extraction

As mentioned in 5.1.1, propositions are the basic input units in my model, so the first step is to decompose an incoming sentence into propositions. Previous work on similar models (Kintsch, 2001; Lemaire et al., 2006) is equivocal on this issue or uses manually extracted propositions. I will fill the gap so that the model

[21] In my experiment, I use the Python modules included in *Gensim*: http://radimrehurek.com/gensim/index.html.

works fully automatically.

Essentially a proposition is represented as *Predicate*(*Argument*$_1$, *Argument*$_2$, …) where the predicate is a verb, noun, or adjective and an argument must be a noun. As propositions can be regarded as generalized events (see 3.3.1 for event extraction), I extract propositions from the dependency tuples after parsing (Klein and Manning, 2003) because they contain information about governing verbs, subjects, objects, and modifiers, from which we can derive propositions.

A difficulty with this approach is that nominal and pronominal anaphora is frequently found in a narrative text. The following example is the first paragraph of the fairy tale *Beauty and the Beast*, where the nouns "merchant", "sons", and "daughters" appear only once and then referred to by 8 pronouns.

(5.1) *ONCE upon a time, in a very far-off country, there lived a merchant who had been so fortunate in all* ***his*** *undertakings that* ***he*** *was enormously rich.* (5.2) *As* ***he*** *had, however, six sons and six daughters,* ***he*** *found that* ***his*** *money was not too much to let* ***them*** *all have everything* ***they*** *fancied, as* ***they*** *were accustomed to do.*

If the pronouns are left as is in the dependency tuples, we have no way to tell that it is the same "merchant" who lived somewhere and was rich (sentence anaphora) and had twelve children (discourse anaphora). To extract high-quality

propositions, I apply coreference resolution to the dependency tuples first, using the state-of-the-art multi-pass sieve system[22] (Lee et al., 2011) that resolves both pronominal and nominal expressions to a head noun phrase.

Propositions are predicate-dominated and once the predicate is identified, all its attached nouns can be retrieved from a proper dependency tuple. For example, in the above Sentence (5.1), "lived" is a verb predicate and "merchant" is its attached noun from ***direct_object*** (*lived*, *merchant*).

Apart from extracting propositions based on the simple "Subject – Verb – Object" skeleton or its passive form, I also find predicates and arguments from modifier and complement structures in participles and clauses that characterize complex sentences. This helps to find, in Sentence (5.1), the proposition *fortunate*(*merchant*) from a clause-level dependency. It is possible that two verbs are found using the modifier or complement dependencies, but only one of them is chosen as the real predicate based on the dependency type. For example, ***complement*** (*accustomed*, *do*) in Sentence (5.2) gives us two verbs as possible predicates, but only *do* is chosen because it "complements" the meaning of *accustomed* and shifts the focus of the sentence.

The following shows all the propositions extracted from the first paragraph of *Beauty and the Beast*. (5.3) and (5.4) list all the propositions in (5.1) and (5.2), respectively. The propositions are ordered by the predicate position in the sentence.

[22] It is included in Stanford CoreNLP, which also includes the state-of-the-art Stanford Parser that I use for dependency parsing.

(5.3) *lived*(*merchant*); *fortunate*(*merchant*, *undertakings*); *rich*(*merchant*); (5.4) *had*(*merchant*, *sons*, *daughters*); *found*(*merchant*, *money*, *sons*, *daughters*); *let*(*money*, *sons*, *daughters*); *have*(*everything*); *fancied*(*sons*, *everything*); *do*(*sons*).

5.1.3.2 Contextualized Word Association

With an input proposition, all its words trigger their closest associates from the semantic network stored in the long-term memory. However, word association partly depends on the explicit context for disambiguation, so that "bank" is associated with "money" in the context of "lend" but "river" in the context of "water". As we read a word in text, we understand it with reference to both its semantically related words in general and the explicit context it appears in, which is the underlying tenet for "gist-level and verbatim-level information" (Steyvers and Griffiths, 2008) or the "dual-memory model" (Kintsch and Mangalath, 2011).

For those reasons, I calculate the **contextualized association score** of word *v* with reference to word *u* and its context *C*(*u*), noted as $CAS_u(v)$. Let's denote the similarity of *u* and *v* in the LSA/LDA space as *Sim*(*u*, *v*) and then

$$CAS_u(v) = \frac{1}{|C(u)|+1} \sum_{w \in C(u) \cup \{u\}} Sim(w, v)$$

In my experiment, *C*(*u*) consists of the left and right neighbors of *u*. Table 5.2 shows the difference between using the contextualized association score

calculated by using *u*'s 3 most frequent neighbors in my experimental dataset and decontextualized association score ($C(u) = \varnothing$) to get the top 3 associates of "kill" according to $CAS_{kill}(v)$. The word similarities are from the Wiki-based LSA space. Obviously, the words "disguise" and "terrified" indicate some special context "kill" is found in.

Contextualized	*revenge, disguise, terrified*
Decontextualized	*revenge, dead, steal*

Table 5.2: Top 3 Associates of "kill", using Wiki LSA

5.1.3.3 Spreading Activation in Working Memory

After we get the top *n* associates (in my experiment, *n* = 3) for each word, all the proposition words and their associates are now resident in the working memory, each with an **activation score** *AS* that denotes its degree of activation. Initially, *AS*(*w*) = 1 if *w* is from the proposition. Otherwise it is set to be *w*'s association score. If *w* appears more than once, the maximum *AS*(*w*) is taken so that all scores fall in [0, 1]. But the initial degrees of activation are unstable because of the relations between the words stored in the semantic network. Ultimately, some words may stabilize with higher scores because they are closely related to more activated words and some with lower scores because they are related to less activated words, which is supported by the reinforcement of relevant information and deactivation of irrelevant information (Tapiero, 2007:

87).

This cognitive process can be modeled by a spreading activation algorithm, first introduced by (Kintsch, 1998). Let A be a vector of the activation scores of n words: $w_1, \ldots, w_n$: $A = (a_1, ..., a_n)^T, a_i = AS(w_i)$ and M be a similarity matrix for the n words: $M = [m_{ij}]_{n \times n}$, $m_{ij} = Sim(w_i, w_j)$. Let $A^{(t)}$ denote A at time t and define

$$A^{(t+1)} = MA^{(t)} / \max\{abs(MA^{(t)})\}$$

A is thus constantly updated by multiplying M and normalizing by the vector component with the largest absolute value: $\max\{abs(MA^{(t)})\}$. I now prove its convergence.

Suppose $v_1, \ldots, v_n$ are the n eigenvectors of M, corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$ in descending order of their absolute values. According to the definition, $A^{(t)}$ is also bounded ([0, 1]). Suppose λ_1 is the single root of the characteristic polynomial, then using the eigenvectors,

$$A^{(0)} = a_1 v_1 + ... + a_n v_n,$$

$$A^{(t)} = M^t A^{(0)} / \varphi_t = M^t (a_1 v_1 + ... + a_n v_n) / \varphi_t$$

$$= (a_1 \lambda_1^t v_1 + ... + a_n \lambda_n^t v_n) / \varphi_t \overset{t \to \infty}{\approx} a_1 \lambda_1^t v_1 / \varphi_t$$

where φ_t is the normalization coefficient at time t. This shows that φ_{t+1} is actually dependent on the component of v_1 with the largest absolute value. Therefore, $A^{(t)}$ converges to $v_1 / \max\{abs(v_1)\}$, where $\max\{abs(v_1)\}$ is the component of v_1 with the largest absolute value.

5.1.3.4 Activation Adjustment in Episodic Memory

After the current proposition is processed and before the next proposition comes, the activated words are transferred to the episodic memory with their activation scores copied if they did not exist. Otherwise, the activation scores are updated. If the activation score of *w* in the episodic memory after the *n*th proposition is $ES^n(w)$ and its activation score (after spreading activation) in the working memory is $AS(w)$, then

$$ES^n(w) = Min(1, ES^{n-1}(w) + AS(w) - ES^{n-1}(w)AS(w))$$

It is easy to see that $ES^n(w)$ is no less than $ES^{n-1}(w)$ or $AS(w)$ (Lemaire et al., 2006) and is still bounded by 1.

On the other hand, according to the **Decay Theory** (Berman, 2009), earlier processed words are gradually forgotten over time. To model this phenomenon, I follow (Lemaire et al., 2006) by setting a decay coefficient (δ = 0.9) as a multiplier to $ES^n(w)$ for all *w* in the episodic memory after proposition *n* is processed.

Stories typically mention major characters and happenings in different places, and each later mention makes us recall what was earlier said about them. So a word *w* can be **reactivated** back into the working memory if $ES^{n-1}(w) > \theta_1^{n-1}$ and $Sim(w, u) > \theta_2$ for some *u* in the *n*th proposition. Note that instead of taking a fixed value, the activation score threshold θ_1 is dependent on the current state of the episodic memory: $\theta_1^n = \sum_w ES^n(w) / |ES^n(w)|$. θ_2 is independent of the episodic memory and set to be 0.7.

5.1.3.5 Complete Algorithm

To sum up this part, I provide the complete algorithm of the cyclic text comprehension in Figure 5.2. *WM* and *EM* are mnemonic notations for sets of words with their activation scores in the working memory (WM) and episodic memory (EM). Note that when the algorithm terminates, *EM* contains all the activated words from both the text and the long-term memory, with their final activation scores.

WM = {};

EM = {};

While exists next proposition $P^i = \{p_1^i, p_2^i, ...\}$

$WM \leftarrow P^i \cup$ {top *n* associates of $p_1^i, p_2^i, ...\}$ ∪ {words reactivated from *EM* by P^i };

Apply spreading activation to *WM*;

EM ← *EM* updated with *WM*;

Apply decay to *EM*;

Figure 5.2: Algorithm of Cyclic Comprehension

5.2 Coherent Narrative Summarization

After all propositions in a narrative text have been processed, the episodic memory contains all the activated words with their activation scores. This is the word-level representation of the text according to my cognitive model. Moreover,

the highly activated words are relevant to each other because of the spreading activation mechanism. A passage based on such words is expected to be highly coherent. Therefore, a coherent summary of the narrative text can be constructed by focusing on the highly activated words in the episodic memory.

A summary, however, cannot be a mere collection of words. It is expected to be composed of well-formed sentences well connected to each other. A straightforward method is to interpret the highly activated word as the most salient words and select the original sentences containing such words, as most frequency-based extractive summarizers do. In a psychological study, Lemaire et al. (2005) show that selecting sentences based on the word values calculated from the CI model, which our cognitive model is built on, highly correlates with the human selection of sentences to make up a narrative summary.

Although selecting the original sentences may work in our case, it misses an important aspect of our model – propositions. According to Figure 5.1, the model receives propositions as input in each reading cycle because proposition is the basic unit of human understanding. After reading the whole text, the propositions receive different degrees of salience in the reader's mind, and a summary should maximally cover all salient and non-redundant propositions. The proposition-based summarization architecture is illustrated in Figure 5.3.

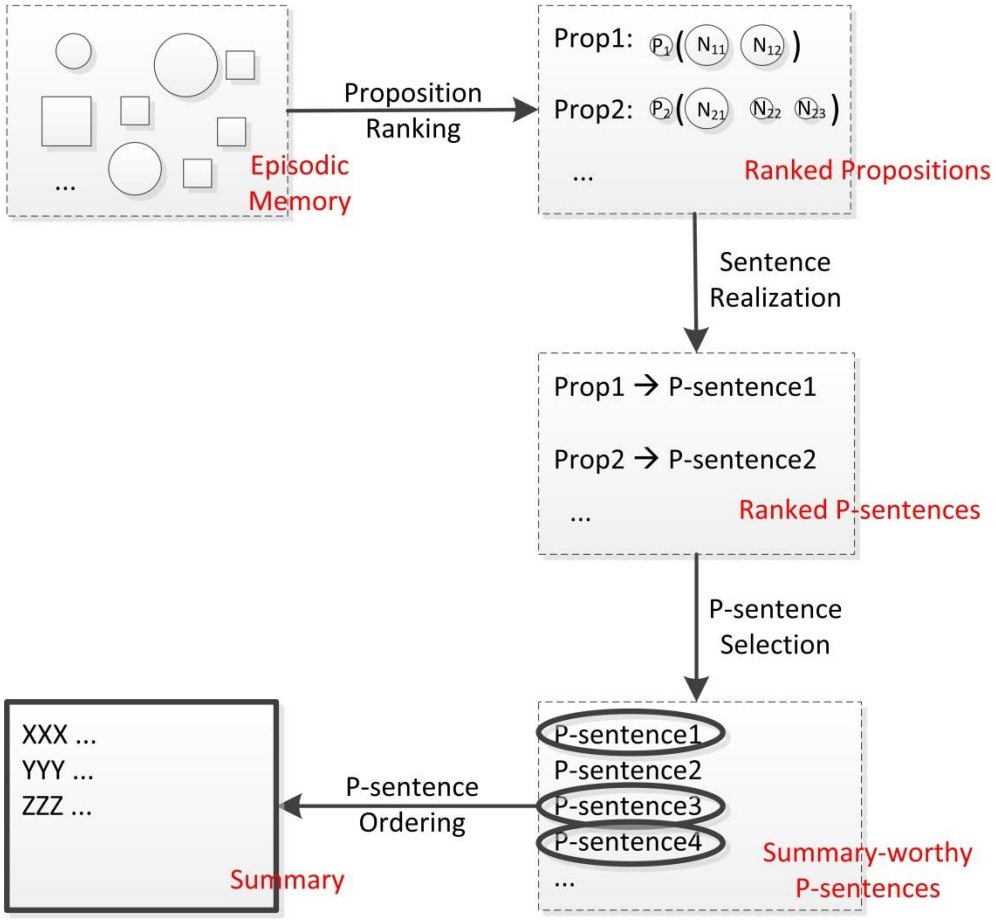

Figure 5.3: Architecture of Narrative Summarization, Based on the Cognitive Model

As is shown in the above, the input propositions are first **ranked** according to the activation scores of their constituent words from the episodic memory after the whole text comprehension is completed. But since the summary cannot be composed of propositions like *killed*(*hunter*, *bear*), the propositions need to be **realized** as sentences, or **p-sentences**, which are not necessarily the original sentences from the text. From the ranked p-sentences I select those worthy of being included in a summary. In principle, the **selected** p-sentences need to be both salient (high-ranking) and non-redundant. Finally, the p-sentences are

ordered to form the output summary.

A major challenge to apply the cognitive model to summarization is sentence realization. Generating sentences directly from our propositions is not feasible because much sentence-building information (verb tense, voice, mood, function words, etc.) cannot be found in the propositions. My solution is to find sub-sentences corresponding to the propositions from the original text, a strategy to be elaborated in the following.

5.2.1 Proposition-based Sentence Extraction

Now let's discuss the detailed algorithm of extracting p-sentences from an original sentence. As discussed above, they are the building blocks of the summary. For that purpose, a p-sentence is expected to be informationally compact (containing as little non-proposition material as possible) and grammatically acceptable. Such agenda can be met by operations on the parsing tree of the original sentence, which contains hierarchical relations between proposition elements as well as syntactical information about how they can be connected in a grammatical way.

Parsing-based methods and tree operations are commonly used in sentence revision (Mani et al., 1999), compression (Cohn and Lapata, 2008; Yousfi-Monod and Prince, 2008; Zajic et al., 2008), reduction (Jing, 2000; Jing and McKeown, 2000), or fusion (Barzilay and McKeown, 2005) to improve the summary quality. My sub-tree deduction algorithm in the following has

borrowed ideas, e.g., tree pruning and adjusting, from those previous works. But to the best of my knowledge, no attempt has been made to deduce sections of a tree to match propositions.

5.2.1.1 P-sentence Extraction as Sub-tree Deduction

If a sentence contains *n* propositions, we can extract *n* p-sentences. Although the *n* p-sentences are all parts of the original sentence, they are not necessarily non-overlapping. Consider sentence (5.5) below, which is selected from my experimental dataset, and its automatically extracted propositions (Prop1 to Prop4) in (5.6)

(5.5) *THERE was once a young fellow who enlisted as a soldier, conducted himself bravely, and was always the foremost when it rained bullets*.

(5.6) Prop1: *fellow* (*THERE*)

Prop2: *enlisted* (*fellow*, *soldier*)

Prop3: *foremost* (*fellow*)

Prop4: *rained* (*bullets*)

Prop2 and Prop3 both have the word “fellow” as an argument, so their p-sentences must be overlapping. Thus, extracting p-sentences from the original sentence is not decomposing the sentence into non-overlapping parts. Rather, it is

formulated as a sub-tree deduction process. Figure 5.4 shows the parsing tree of sentence (5.5), the output of the state-of-the-art Stanford Parser.

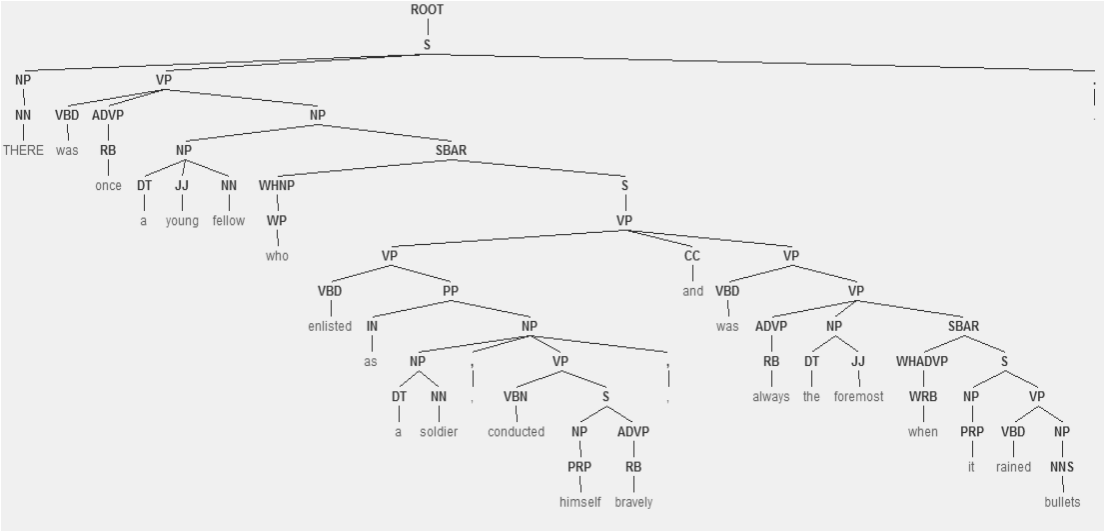

Figure 5.4: Parse Tree of Example Sentence (5.5)

Given such a parse tree and a proposition from the sentence, our goal is to deduce a sub-tree that minimally covers the proposition elements and preserves all the syntactically necessary constituents. The following is the top-level algorithm to attain this goal.

Input: parse tree *T*, propositions *Prop* = *P*(N_1, N_2, …) **Output**: sub-tree *ST*(*Prop*) covering *Prop*
1. **Find the lowest common parent**, *CP*, of *P*, N_1, N_2, … in *T*; 2. For each element *e* in *Prop*: **Grow a sub-tree** *ST*(*CP*, *e*) with *CP* as the root and *e* as a leaf; 3. **Merge all sub-trees** *ST*(*CP*, *e*) into one sub-tree *ST*(*Prop*); 4. If the root node of *ST*(*Prop*), *CP*, is NP **Adjust** *ST*(*Prop*);

Figure 5.5: Top-level Algorithm of Sub-tree Deduction

In the following, I will discuss the main steps of the algorithm.

- **Find the lowest common parent**

Given proposition elements in different places of the parse tree, we need to find a sub-tree that covers all those nodes. On the sub-tree, there is a path from the root node to all the proposition elements. To get a most specific sub-tree, its root should minimally cover all the proposition elements. In other words, we need to find the lowest common parent of the proposition elements.

For this purpose, we can simply compare the paths from the root to all leaf (element) nodes and take a common node that is the farthest away from the root. In Figure 5.4, the lowest common parent of *fellow* (*THERE*) is S and that of *enlisted* (*fellow*, *soldier*) is NP.

- **Grow sub-trees**

After the lowest common parent (lcp) is determined, we grow a sub-tree for each proposition element with the lcp as the root and the element as a leaf node by "moving up" the tree. In order to make the sub-tree syntactically well-formed, we try to grow all branches by including all the sibling nodes and branches except where **pruning** is possible.

Pruning is applied to sub-trees decided to be subordinate or ancillary, whose absence does not affect the grammaticality of the resultant sentence. Using linguistic knowledge, I use two pruning rules:

- Prune the left or right sub-tree with the root node of SBAR or SBARQ and all its left or right siblings.
- Prune the left or right sub-tree with the root node of CC and all its left or right siblings.

The rules are aimed to eliminate detachable subordinate clauses and coordinate constituents. In Figure 5.4, when growing *fellow* (*THERE*) by moving up the tree, we encounter a node SBAR as the sibling of (NP, (DT: *a*, JJ: *young*, NN: *fellow*)), so the whole sub-tree with SBAR as the root is pruned. Moving up one level, (NP, (DT: *a*, JJ: *young*, NN: *fellow*)) grows into (NP, (NP, (DT: *a*, JJ: *young*, NN: *fellow*))).

- **Merge sub-trees into one**

With the grown sub-trees sharing a common root, we next merge them into

one sub-tree that represents the whole p-sentence. Essentially, the merging process is to adjoin same-root sub-trees as branches of a bigger sub-tree. In this process, redundant branches are eliminated.

In Figure 5.4, we can grow two identical sub-trees for *fellow* (*THERE*): (S (NP (NN: *THERE*)) (VP (VBD: *was*, ADVP (RB: *once*), NP, (NP, (DT: *a*, JJ: *young*, NN: *fellow*))))), which are merged into one copy, corresponding to the p-sentence: *THERE was once a young fellow*.

- **Adjust the sub-tree**

The deduced sub-tree is expected to represent a complete sentence, which means its root must be S. On the other hand, the sub-tree should represent a proposition, which is backboned by NPs and VPs. We find that there are two major root nodes: S and NP. In the former case, we directly output the sub-tree; in the latter, we need to adjust the structure of the sub-tree.

In almost all cases, the NP-rooted sub-tree represents a noun phrase with a clause modifier. Functionally, the head NP plays a role in the clause and can be moved into the clause at an appropriate place, so that the root of the sub-tree becomes S. The following lists the major cases of an NP-rooted sub-tree and the adjusted result.

- (NP_0, (NP_1, SBAR (S_0 (…))))) → (S, (NP_1, (S_0 (…))))
- (NP_0, (NP_1, SBAR (WHNP, S_0 (…))))) → (S, (NP_1, (S_0 (…))))
- (NP_0, (NP_1, SBAR (WHNP, VP (…))))) → (S, (NP_1, VP (…)))

- $(NP_0, (NP_1, SBAR\ (WHPP, S_0\ (\ \dots\ ))))) \rightarrow (S, (NP_1, (S_0\ (\ \dots\ ))))$

In Figure 5.4, the merged sub-tree for *enlisted* (*fellow*, *soldier*) represents the sentence: *a young fellow who enlisted as a solider ...* with the (NP, (NP, SBAR (WHNP, S (VP, …))))) structure. After *a young fellow* (NP) is moved to the inner sentence, we come up with *a young fellow enlisted as a soldier ...* with the (S (NP, S (VP, …))) structure.

5.2.1.2 A Complete Example

Now let's illustrate the algorithm of sub-tree deduction by walking through a complete example, sentence (5.5) with the four propositions shown in (5.6). I show an annotated parse tree in Figure 5.6 to facilitate the discussion. Note that the boxed nodes are proposition elements, the shaded nodes are the lowest common parents, and the "X" indicates pruning places.

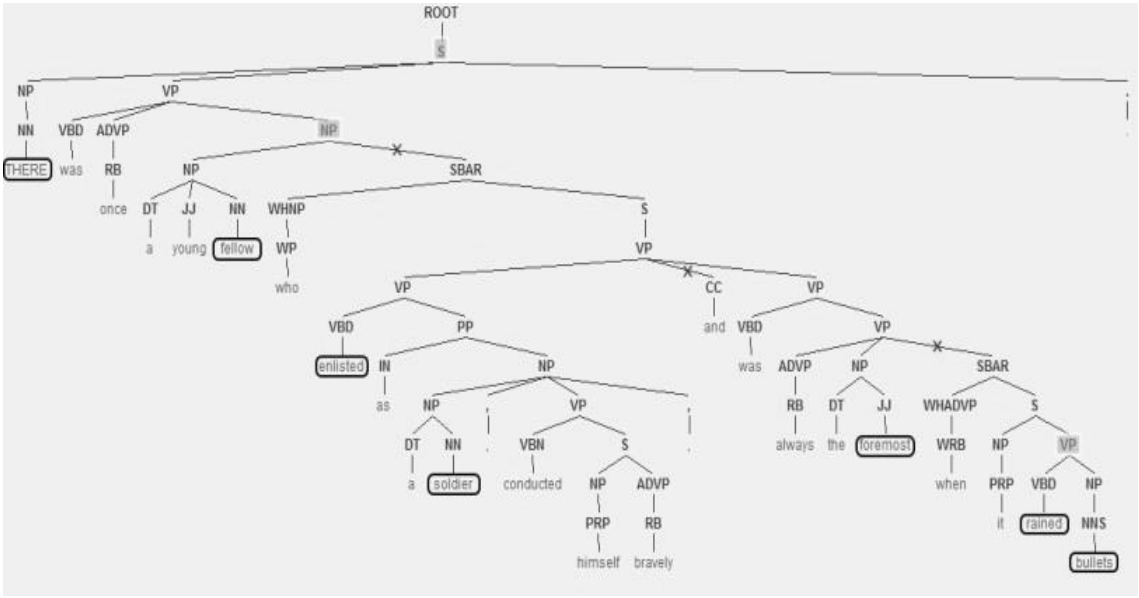

Figure 5.6: Annotated Parse Tree of Example Sentence (5.5)

- **Find the lowest common parent (lcp)**

The lcp of *fellow* (*THERE*) is the top-level S. The lcp's of *enlisted* (*fellow*, *soldier*) and *foremost* (*fellow*) are both NP. The lcp of *rained* (*bullets*) is VP.

- **Grow sub-trees**

For *fellow* (*THERE*), starting from *fellow* and *THERE*, we grow the same sub-tree: (S (NP (NN: *THERE*)) (VP (VBD: *was*, ADVP (RB: *once*), NP, (NP, (DT: *a*, JJ: *young*, NN: *fellow*))))). Note that the SBAR branch is pruned, as indicated in the figure.

For *enlisted* (*fellow*, *soldier*), *fellow* grows into (NP (NP (DT: *a*, JJ: *young*, NN: *fellow*))) as the SBAR branch is pruned. But *enlisted* and *soldier* both grow into (NP (NP (DT: *a*, JJ: *young*, NN: *fellow*)), SBAR (WHNP (WP: *who*), S (VP (VP (VBD: *enlisted*, PP (IN: *as*, NP (NP (DT: *a*, NN: *soldier*), ,:, VP (VBN: *conducted*, S (NP (PRP: *himself*), ADVP (RB: *bravely*))), ,:,)))))))). Note that the CC branch and its right VP sibling are pruned during the growth.

For *foremost* (*fellow*), *fellow* grows into the same sub-tree as the above and *foremost* grows into (NP (NP (DT: *a*, JJ: *young*, NN: *fellow*)), SBAR (WHNP (WP: *who*)), S (VP (VP (VBD: *was*, VP (ADVP (RB: *always*), NP (DT: *the*, JJ: *foremost*)))))). During its growth, the SBAR branch as well as the CC branch and its left VP sibling are pruned.

For *rained* (*bullets*), *rained* and *bullets* both grow into (VP (VBD: *rained*, NP (NNS: *bullets*))).

- **Merge sub-trees into one**

For *fellow* (*THERE*), the two identical sub-trees merge into one: (S (NP (NN: *THERE*)) (VP (VBD: *was*, ADVP (RB: *once*), NP, (NP, (DT: *a*, JJ: *young*, NN: *fellow*))))).

For *enlisted* (*fellow*, *soldier*), the merged sub-tree is (NP (NP (DT: *a*, JJ: *young*, NN: *fellow*)), SBAR (WHNP (WP: *who*), S (VP (VP (VBD: *enlisted*, PP (IN: *as*, NP (NP (DT: *a*, NN: *soldier*), ,:, VP (VBN: *conducted*, S (NP (PRP: *himself*), ADVP (RB: *bravely*))), ,:,)))))).

For *foremost* (*fellow*), the merged tree is (NP (NP (DT: *a*, JJ: *young*, NN: *fellow*)), SBAR (WHNP (WP: *who*)), S (VP (VP (VBD: *was*, VP (ADVP (RB: *always*), NP (DT: *the*, JJ: *foremost*)))))).

For *rained* (*bullets*), the two identical sub-trees merge into one: (VP (VBD: *rained*, NP (NNS: *bullets*))).

- **Adjust the sub-tree and output the p-sentence**

For *fellow* (*THERE*), the root node is S and no adjustment is needed. The corresponding p-sentence is ***THERE was once a young fellow***.

For *enlisted* (*fellow*, *soldier*), the root node is NP. Therefore, we adjust the sub-tree by moving NP (DT: *a*, JJ: *young*, NN: *fellow*)) inside the embedded S, resulting in S (NP (DT: *a*, JJ: *young*, NN: *fellow*), VP (VP (VBD: *enlisted*, PP (IN: *as*, NP (NP (DT: *a*, NN: *soldier*), ,:, VP (VBN: *conducted*, S (NP (PRP: *himself*), ADVP (RB: *bravely*))), ,:,))))). The corresponding p-sentence is ***a young fellow enlisted as a soldier, conducted himself bravely***.

For *foremost* (*fellow*), the root node is also NP. After adjustment, its sub-tree

is S (NP (DT: *a*, JJ: *young*, NN: *fellow*), VP (VP (VBD: *was*, VP (ADVP (RB: *always*), NP (DT: *the*, JJ: *foremost*))))), corresponding to ***a young fellow was always the foremost***.

For *rained* (*bullets*), the root node is VP and cannot be adjusted. Its corresponding p-sentence is thus ***rained bullets***. Note that the sentence is incomplete because we have not included the pronoun "it", which cannot be resolved to a meaningful NP, as a proposition element.

5.2.2 Proposition-level Extractive Summarization

In this section, I will flesh out the details of the summarization module, i.e., Figure 5.3. Proposition is pivotal in that it links the cognitive model of text comprehension and the summarization module.

5.2.2.1 Proposition Ranking

Ranking is not unfamiliar to many traditional extractive summarization approaches, which is often motivated by including the most important information in the summary. But for my task, ranking is motivated by finding the **cognitively salient and coherent** information. Owing to the cognitive model of text comprehension/coherence, the final-state episodic memory contains all text words with their activation scores. The higher the score, the more salient the word in a cognitive sense (i.e., the easier the word is remembered). More importantly, the highest ranking words or word groups (propositions) must be

well connected to each other because of the spreading activation mechanism of the cognitive model. This is how I assimilate coherence into the information selection stage of summarization. In comparison, a coherence account is unavailable for most other ranking-based summarization schemes.

With scored words in the episodic memory, let's consider a proposition *Prop* made up of a predicate *P* and *m* arguments: *Prop* = *P* (N_1, … N_m), with all proposition elements having an activation score *AS*(*P*), *AS*(N_1), … *AS*(N_m). According to the propositional structure, N_1, … N_m are parallel to each other and *P* is associated with them all. So we define the ranking score (*RS*) of *Prop* as:

$$RS(P(N_1...N_m)) = AS(P)\sum_{i=1}^{m} AS(N_i)$$

Proposition ranking is then based on the ranking scores of all propositions.

5.2.2.2 Sentence Realization

Based on p-sentence extraction, realizing propositions as sentences (in fact, p-sentences) is straightforward. To the extracted p-sentences we apply simple modifications to make them real sentences, such as sentence-initial capitalization and sentence-ending punctuation.

I manually checked all the 289 p-sentences of a text ("Bearskin") from the experimental dataset. It turns out that most of them (282) are grammatical. The ungrammatical cases are all due to parsing errors ("*Thee a coat and a cloak.*") and incomplete propositional structures ("*Rained bullets.*").

After sentence realization, we come up with ranked p-sentences that can be

used for summarization by directly using the ranking score of the corresponding propositions. In other words, for a p-sentence PS_i and its corresponding proposition $Prop_i$,

$$RS(PS_i) = RS(Prop_i)$$

Alternatively, we can also discount long p-sentences by word length normalization. Suppose $Words(PS_i)$ denotes all the words in PS_i, then

$$RS(PS_i) = RS(Prop_i) / |Words(PS_i)|$$

5.2.2.3 P-sentence Selection

The selection of ranked p-sentences should follow two principles. First, the selected p-sentences rank as high as possible, so that they are not only cognitively salient by themselves, but also well connected to each other. Second, the selected p-sentences overlap as little as possible.

In summarization, sentence overlap or redundancy is generally avoided. For our proposition-based scheme, this problem is exacerbated by p-sentence extraction. Since proposition elements span different sections of the parse tree, the p-sentences of an original sentence may be nearly identical or subsume each other.

The p-sentence selection algorithm is presented in Figure 5.7

Input: words with ranking scores *RS*(*w*), ranked p-sentences *RP*, summary length *SL*

Output: sum = {selected p-sentences}

1. sum = { };
2. While total length of sum < *SL*

    sum = sum ∪ {*ps**, the top-ranking p-sentence in *RP*};

        for each word *w*' in *ps**

            RS(*w*') = *RS*(*w*') * *ε*;

    Delete redundant p-sentences in *RP*;

        Re-rank the remaining p-sentences in *RP*, using updated *RS*(*w*);

3. Output sum;

Figure 5.7: Algorithm of P-sentence Selection

Summary-worthy p-sentences are selected iteratively until the summary length is reached. In each iteration, I select the top ranking p-sentence *ps** and then discount the ranking score of all the words in *ps** by multiplying *ε* (= 0.9 in my experiments). Redundant sentences are determined by both string comparison and cosine similarity (= 0.75 in my experiments). The remaining sentences are re-ranked using the updated word scores to further avoid redundancy.

5.2.2.4 P-Sentence Ordering

Since the cognitive model works only for a **single** narrative text, the

summaries to be produced are single-document summaries by nature. To output the final summary, the selected p-sentences are textually ordered, i.e., according to the position of their subsuming original sentences in the original text. P-sentences belonging to the same original sentence are ordered according to their string positions in the original sentence.

An alternative is to use the grouping-based ordering scheme developed in Chapter 3. In the current task, however, I will drop this option and focus on the effect of cognitive model-driven coherence for summarization.

5.3 Experiments with Event-centric News and Fairy Tales

In order to evaluate the effectiveness of the cognitive model of text comprehension and the model-driven coherence for summarization, I experimented on two kinds of dataset: event-centric news and fairy tales. Essentially, the datasets are narrative, which fit the proposition-based mechanism of the cognitive model.

I select event-centric news and fairy tales for experimentation mainly because the data are freely available and copyright-free. The news articles are also different from the fairy tales in content and style, which provides an opportunity to compare the model's effectiveness on two different kinds of narrative text. Note that presently the cognitive model-driven coherence deals only with single-document summarization.

5.3.1 Event-centric News

In this section, I report the experimental results on the selected DUC 01 and 02 datasets.

5.3.1.1 Data Preparation

The DUC/TAC summarization track provides an abundance of newswire documents, together with human summaries that can be used for evaluation purposes. Among them, only DUC 01, 02, 03, and 04 have single-document summarization tasks[23]. But for DUC 03 and 04, the single-document summaries are very short – 10 words or 75 bytes – for which my approach can hardly show its advantage. In comparison, DUC 01 and 02 ask for 100-word single-document summaries, from which I selected event-centric news articles.

The news articles of DUC 01 and 02 are of two types: event-centric and entity-centric. The former focuses on a central event, such as a terrorist attack or an earthquake; the latter centers on a central person, thing, or other entities, such as a celebrity or a socio-cultural phenomenon. I manually selected the event-centric news articles from the DUC 01 and 02 datasets, totaling 637 documents. Table 5.3 lists the details.

[23] http://www-nlpir.nist.gov/projects/duc/pubs.html

	# All News Articles	# Event-centric News Articles
DUC 01	600	249
DUC 02	567	388
Total	1167	637

Table 5.3: Composition of the Event-centric News Dataset

The DUC annotators provided two human summaries for each news articles, which can be used as reference summaries in automatic evaluation described in the following.

5.3.1.2 Experimental Design

The DUC summarization tasks require single-document summaries of a fixed length: 100 words. I match this length by generating 100-word summaries based on the cognitive model. The evaluation metric is the widely accepted ROUGE (Lin, 2004) that has been used in previous chapters. Admittedly, ROUGE is a good measure of a summary's information coverage, not its coherence. However, I regard it as an indirect measure of coherence. On the one hand, ROUGE measures how similar an automatic summary is to the human-written reference summary, which is reasonably coherent. On the other hand, coherence underlies information selection according to our cognitive model. The selected information is simultaneously (cognitively) important and coherent, so a high score on information coverage should indicate good

coherence. ROUGE is also an expedient choice as manually evaluating thousands of summaries is currently unaffordable.

As the success of the cognitive model depends considerably on its knowledge base – the semantic network, I will first evaluate the different ways of its construction: using different corpora (Wiki, Reuters) and different semantic models (LSA, LDA). Note that the use of updatable LSA/LDA enables us to combine Wiki and Reuters in an incremental way (Wiki&Reuters) and observe the effect. On the Wiki/Wiki&Reuters corpus, the LSA reduced dimensionality and the LDA number of topics are both set to be 400; on the Reuters corpus, both are 100.

With the best cognitive model, I compare two different ways of using the model output (in the episodic memory) to generate summaries: proposition-based summarization and sentence-based summarization. Proposition-based summarization is the approach described in 5.2, using p-sentences to compose summaries. By contrast, sentence-based summarization uses the original sentences selected by ranking them with scores calculated as the sum of the word activation scores. This is a straightforward application of the cognitive model to extractive summarization and an implementation of (Lemaire et al., 2005), which shows that selecting sentences based on values calculated from the CI model highly correlates with the human selection of sentences to make up a narrative summary. What I'm interested in is whether summarizing on the proposition level can improve on summarizing on the sentence level.

Next, the summaries generated from the best model and best summarization scheme are compared with baseline summaries and peer summaries that participated in DUC. The baseline summaries are the "Lead" summaries composed of the first 100 words – a strong baseline for news summarization (Brandow et al., 1995). The two sentence scoring schemes – normalized or un-normalized – are also evaluated.

5.3.1.3 Evaluation Results

I first present the ROUGE scores, including ROUGE-1 (unigram overlap), ROUGE-2 (bigram overlap), ROUGE-SU4 (skip bigrams, up to the distance of four), of using different semantic networks as the cognitive basis. The other summarization parameters are held to be the same: all the summaries are proposition-based using un-normalized sentence scoring.

	ROUGE-1	*ROUGE-2*	*ROUGE-SU4*
LSA + Reuters	0.412	0.124	0.185
LSA + Wiki	0.393	0.118	0.169
LSA + Wiki&Reuters	**0.423**	**0.137**	**0.191**
LDA + Reuters	0.401	0.120	0.179
LDA + Wiki	0.386	0.115	0.164
LDA + Wiki&Reuters	0.417	0.129	0.186

Table 5.4: Comparison of Semantic Network Constructions

According to the results, the LSA-based versions consistently outperform their LDA-based counterparts, which lends credence to the wide use of LSA as a cognitive modeling tool in many domains. The specialized Reuters corpus works better than the generalized Wiki corpus, showing that the cognitive model works better on documents similar to the training corpus. Not surprisingly, enlarging the size of the corpus boosts performance further.

Using the best cognitive basis (LSA + Wiki&Reuters), I compare four summary variants: proposition-based/sentence-based summarization + normalized / un-normalized sentence scoring. Table 5.5 shows the results.

	ROUGE-1	*ROUGE-2*	*ROUGE-SU4*
Proposition + Un-normalized	0.423	0.137	0.191
Proposition + Normalized	**0.434**	**0.141**	**0.196**
Sentence + Un-normalized	0.411	0.128	0.185
Sentence + Normalized	0.417	0.133	0.190

Table 5.5: Comparison of Summarization Schemes

Proposition-level extraction obviously outperforms sentence-level extraction, which confirms my hypothesis about the significance of proposition in both cognitive modeling and summarization. Normalizing p-sentences also works, which suggests that as length increases, news sentences are likely to include non-essential information.

Based on those results, I proceed to compare the best summaries produced by our system (LSA + Wiki&Reuters, proposition-based, normalized sentence scoring) with DUC peer summaries. There are 11 peer summaries for each DUC 01 source document and 13 peer summaries for each DUC 02 source document, most of which are produced by different systems. Therefore, the summaries for DUC 01 and DUC 02 are evaluated separately.

	ROUGE-1	*ROUGE-2*	*ROUGE-SU4*
DUC 01			
Lead (Baseline)	0.429	0.140	0.192
Best DUC peer	0.433	0.142	0.193
My method	**0.437**	**0.148**	**0.199**
DUC 02			
Lead (Baseline)	0.425	0.128	0.187
Best DUC peer	0.427	0.133	0.191
My method	**0.432**	**0.137**	**0.194**

Table 5.6: Comparison of Summaries for DUC 01/02 Event-centric Articles

The results in Table 5.6 are hard evidence that my method outperforms the best known systems, although the superiority is not very obvious. In fact, single-document news summarization has been long given little credit to its research value. Part of the reason is the simplicity and robustness of the Lead

baseline, as is shown by the little gap between the Lead and the best DUC peer system. The difference between the Lead and my method, however, is more noticeable.

Now it is interesting to ask whether the cognitive model of narrative text comprehension and coherence also works for non-narrative news text (i.e., entity-centric text). Theoretically, non-narrative news text lacks "plot development" that can be well captured by the cyclic reading process, so the model should not work well. In order to test this hypothesis, I also experimented on all the entity-centric news articles from the DUC 01 and 02 datasets. According to Table 5.3, there are a total of 530 such articles. Table 5.7 shows the result, using the same model and summarization scheme.

	ROUGE-1	*ROUGE-2*	*ROUGE-SU4*
DUC 01			
Lead (Baseline)	0.427	0.139	0.189
Best DUC peer	0.430	0.141	0.190
My method	0.428	0.137	0.185
DUC 02			
Lead (Baseline)	0.429	0.138	0.192
Best DUC peer	0.432	0.143	0.192
My method	0.427	0.136	0.188

Table 5.7: Comparison of Summaries for DUC 01/02 Entity-centric Articles

This time, the summaries produced by our method performs poorly, defeated even by the Lead baseline. Since the different results from Table 5.6 and 5.7 can only derive from the different natures of the text, I conclude that the cognitive model and proposition-based approach works best with narrative text.

5.3.2 Fairy Tales

A more typical genre of narrative text is story. Story summarization is rarely reported in early days (Lehnert, 1999) but sees a burgeoning growth in recent years (Kazantseva, 2006; Mihalcea and Ceylan, 2007; Kazantseva and Szpakowicz, 2010). None of them, however, makes use of a cognitive model like the one developed in this chapter. In this set of experiments, I used fairy tales as they have clear plots and narrative structures, which is ideal for the cognitive model.

5.3.2.1 Data Preparation

The fairy tales used as experimental data are mostly by Brothers Grimm and Hans C. Anderson because those classic works are copyright-free and quality human summaries can be found on dedicated websites[24] or Wikipedia. Using free online resources[25], I built a dataset of 50 fairy tales, each accompanied with a human summary. All the human summaries are manually checked to ensure that they are truly descriptive, not evaluative, summaries (Ceylan and Mihalcea,

[24] http://www.comedyimprov.com/music/schmoll/tales.html
[25] http://www.surlalunefairytales.com/

2009). Table 5.8 lists the length statistics.

	Max	***Min***	***Average***
Original Text (# words)	48190	461	4025.6
Summary (# words)	1594	74	396.3
Summary / Original Ratio	0.52	0.01	0.16

Table 5.8: Fairy Tale Dataset Length Statistics

Unlike the news articles used in the first set of experiments, both the fairy tale text lengths and compression (summary/original) ratios vary a lot. So for an automatic summary, I match its length to the human summary length instead of taking a fixed length or ratio, such as the 100 words for news articles.

5.3.2.2 Experimental Design

The evaluation objects are similar to those for the event-centric news. First, I compare the different ways of constructing the semantic network to feed the cognitive model: using LSA/ LDA and 3 different corpora: Wiki, FT, Wiki&FT. On the Wiki/Wiki&FT corpus, the LSA reduced dimensionality and the LDA number of topics are both set to be 400; on the FT corpus, both are 100. Next, I test the efficacy of proposition-based summarization scheme and sentence normalization.

For summary comparison, no peer summaries are available. So I will

compare our summaries with those produced with 3 well-known and popular methods: Luhn's (1958) algorithm, MEAD (Radev et al., 2004) as implemented in (Mihalcea and Ceylan, 2007), and TextRank (Mihalcea and Tarau, 2004). Luhn's classic algorithm is one of the best known for single-document summarization. MEAD and TextRank are popular summarization methods that have been applied to story summarization (Mihalcea and Ceylan, 2007). All of them produce extractive summaries based on sentence scoring by using word frequency, position information, sentence relation, etc. As in the previous set of experiments, I produce "Lead" summaries.

Both automatic evaluation and human evaluation will be done for this set of experiments. For the automatic evaluation, I still use the ROUGE measures for reasons explained in 5.3.1.2. But this smaller dataset also makes it possible to do human evaluation so that coherence can be more directly evaluated. Using the best summaries from previous results, I ask 2 human judges to score 4 different summaries for each of the 50 fairy tales, on a scale of 5 points, in response to the following statements.

S1: This summary gives me enough information to understand what the story is about.

S2: The sentences in the summary of the story are coherent and well connected to each other.

S3: Except for the last sentence, the sentences in the summary are

grammatical and complete.

Complete agreement with a statement leads to a score of 5 and complete disagreement leads to a score of 1. The three statements are aimed to evaluate *informativeness*, *coherence*, and *grammaticality* respectively. Note that because of the truncation to meet the word limit, the last sentence of an automatic summary is probably incomplete. This factor should be excluded in grammaticality evaluation.

5.3.2.3 Evaluation Results

Using different semantic network constructions to build the cognitive model, I report the ROUGE scores in Table 5.9. As in the first set of experiments, the other summarization parameters all take default settings, i.e., proposition-based summarization and un-normalized sentence scoring.

	ROUGE-1	*ROUGE-2*	*ROUGE-SU4*
LSA + FT	0.440	0.097	0.170
LSA + Wiki	0.447	0.101	0.176
LSA + Wiki&FT	**0.452**	**0.102**	**0.179**
LDA + FT	0.449	0.097	0.174
LDA + Wiki	0.444	0.100	0.174
LDA + Wiki&FT	0.444	0.098	0.173

Table 5.9: Comparison of Semantic Network Constructions

Compared with Table 5.4, the results are less consistent. Using the specialized FT corpus, the LSA-based model underperforms the LDA-based model. But using the larger Wiki and Wiki&FT corpora, the LSA-based model performs better. Interestingly, if LDA is used, a larger corpus does not necessarily help fairy tales whereas it does help news (Table 5.4). Since LDA works with topic modeling, a plausible explanation is that the topics of fairy tales, which include particular characters and settings, are more specific than those of news and the mostly non-fairy tale text in Wiki cannot help in finding such topics to build the semantic network. Combining Wiki and FT introduces a lot of noise to fairy tale topics and is thus counterproductive. The LSA-based models, on the other hand, are more robust and consistently benefit from larger training corpora.

I observe that when the training corpus is small and specialized, LDA is more effective than LSA. But when the training corpus is large and generic, LSA shows pronounced advantage. Similar to the results on event-centric news, LSA + all available training data (Wiki&FT) gives the best performance. Based on this construction of the semantic network, I compare summaries produced from the different combinations of proposition-based/sentence-based summarization and normalized / un-normalized sentence scoring.

	ROUGE-1	*ROUGE-2*	*ROUGE-SU4*
Proposition + Un-normalized	**0.452**	**0.102**	**0.179**
Proposition + Normalized	0.446	0.098	0.172
Sentence + Un-normalized	0.430	0.097	0.170
Sentence + Normalized	0.412	0.093	0.162

Table 5.10: Comparison of Summarization Schemes

Using the cognitive model output, proposition-level extraction proves more effective than sentence-level extraction for fairy tales as well as event-centric news. But unlike the summarization of event-centric news, sentence normalization is counterproductive for fairy tale summarization. This shows a textual difference between news and fairy tales. In terms of narrative content (proposition elements), longer sentences in news contain more noise (non-narrative content). During p-sentence extraction, such noise is usually indispensable for syntactic completeness. By contrast, sentences in fairy tales contain mostly narrative content and during p-sentence extraction, long sentences can often be decomposed into shorter p-sentences. An illustrative case is shown in examples (5.7) and (5.8).

(5.7)

(original sentence) *SQUADS of workers fanned out across storm-battered Louisiana yesterday to begin a massive rebuilding effort after Hurricane Andrew*

had flattened whole districts, killing two people and injuring dozens more, agencies report from Florida and New Orleans.

(p-sentences)

SQUADS of workers fanned out across storm-battered Louisiana yesterday to begin a massive rebuilding effort.

Hurricane Andrew had flattened whole districts, killing two people and injuring dozens more.

Agencies report from Florida and New Orleans.

(5.8)

(original sentence) *So long as the war lasted, all went well, but when peace was made, he received his dismissal, and the captain said he might go where he liked.*

(p-sentences)

The war lasted.

Peace was made.

He received his dismissal.

The captain said.

He might go.

He liked.

(5.7) is selected from the news dataset and (5.8) from the fairy tales dataset.

Obviously, the p-sentences of (5.8) are more compact than those of (5.7) in terms of narrative content. Consequently, sentence normalization for fairy tales is not helpful.

Next, I compare the best summaries produced by my system (LSA + Wiki&FT, proposition-based, un-normalized sentence scoring) with 4 peer summaries introduced in 5.3.2.2: Lead, Luhn (1958), MEAD, and TextRank. For fairness, except for Lead, the sentence scoring for the peer summaries are un-normalized. The result is shown in Table 5.11.

	ROUGE-1	*ROUGE-2*	*ROUGE-SU4*
My method	**0.452**	**0.102**	**0.179**
Lead	0.395	0.080	0.147
Luhn (1958)	0.410	0.088	0.157
MEAD	0.419	0.091	0.160
TextRank	0.421	0.092	0.163

Table 5.11: Comparison of Summaries for Fairy Tales

It seems that the superiority of my method over the peer systems is obvious on fairy tales. This result, joined with the result on event-centric news (Table 5.6), testifies the efficacy and robustness of the cognitive model and proposition-based approach to narrative summarization. Interestingly, the Lead summaries of fairy tales perform the worst, showing that a commonly held strong baseline for

single-document summarization does not work well in a typical narrative domain. Therefore, developing new and powerful summarization techniques for narrative text is a very meaningful endeavor.

ROUGE scores can indirectly measure the coherence of the output summaries. But the human evaluation of coherence provides a more direct yardstick. Moreover, since cognitive model-driven coherence is rooted in human cognition and understanding, it makes good sense to validate the end product with human criteria.

For each of the 50 fairy tales, I provide two human judges with 4 summaries: one human summary, one best peer summary (TextRank, according to Table 5.11), and two summaries produced by my method which differ only in the level of sentence extraction – one uses proposition-level extraction and the other sentence-level extraction. Note that human scoring is very time-consuming and labor-intensive. In my experiment, it takes a total of 60 human/days to finish the work.

As is introduced in 5.3.2.2, I asked two human judges to score summaries for coherence as well as informativeness and grammaticality. The human assessment of informativeness will lend further credence to the ROUGE metric. Grammaticality is also evaluated because it is important to find out even though proposition-level extractive summarization renders more informative/coherent summaries than sentence-level extractive summarization, whether it is done at the cost of grammaticality.

For each scoring category, inter-judge agreement is measured by Cohen's Kappa, which ranges between 0.48 and 0.63, indicating good agreement. Then I take the average of the two human scores over the 50 fairy tales on each category and report the result in Table 5.12. Statistical significance of the proposition-level extractive summaries ("My method – proposition-level") against all the other summaries is indicated by * ($p < 0.01$) on a paired two-tailed t-test.

	Informativeness	*Coherence*	*Grammaticality*
Human	*4.32	*4.63	*4.88
My method – proposition-level	**3.27**	**3.39**	**3.87**
My method – sentence-level	3.10	*2.95	3.95
TextRank	*2.98	*2.87	3.84

Table 5.12: Average Human Scores for the Fairy Tale Summaries

The "proposition-level" version represents the best output of my method. Informatively, it is superior to the "sentence-level" version and TextRank summaries, which is consistent with the ROUGE results. In terms of coherence, the proposition-level version outperforms the sentence-level version and TextRank significantly, proving the validity of the cognitive model-driven coherence when effectively integrated into summarization. This is also hard evidence that the proposition-level extractive summarization outperforms sentence-level extractive summarization not only in essential information

coverage, but also (and more importantly) in coherence.

Are the gains in informativeness and coherence achieved at the cost of grammaticality? This concern is relieved by the small gap between the proposition-level version and the sentence-level version, the former being slightly better than TextRank. Such differences, however, are statistically insignificant.

A huge gap does exist between the human summaries and all the automatic summaries in all aspects, a cold fact showing that fairy tale summarization is indeed a challenge. The cognitive model and the summarization scheme proposed in this chapter, however, make a good attempt to take the challenge.

5.4 Chapter Summary

This chapter completes my quest for cognitive modeling in summarization by tapping into the human domain of coherence – cognitive model-driven coherence. Different from content-driven coherence, cognitive model-driven coherence is interpreted by cognitive psychologists as a built-in mechanism in text comprehension. Modeling such coherence is technically equivalent to modeling text comprehension.

The computational model of text comprehension and coherence is based on theoretical models from psychology and cognitive science. A semantic network is computed from a corpus to simulate knowledge stored in the long-term memory, and a proposition-based cyclic comprehension algorithm is proposed to model

the human reading process and the interactions between different parts of the human memory. Upon completion of all the reading cycles, the episodic memory contains all proposition elements with their activation scores.

The scored proposition elements are used to select cognitively salient and coherent information for summarization. Different from most other extractive summarization approaches, I summarize on the proposition level. Propositions are first ranked according to the predicate-argument structure and the word activation scores in the episodic memory. Then they are realized as grammatical sentences, or p-sentences, that are the proper constituents of a summary. The highest ranking and non-redundant p-sentences are then selected for the summary.

The cognitive model-driven coherence works best on narrative text. Therefore, I experimented with two datasets of narrative text: event-centric news and fairy tales. On both datasets, my method outstrips peer systems, proving that for single-document narrative summarization, cognitive model-driven coherence can benefit both informativeness and coherence in the output summaries.

Chapter 6: Conclusion and Future Directions

"In three words I can sum up everything

I've learned about life: it goes on."

Robert Frost

In this chapter, I will wrap up the dissertation by assembling the main technical chapters into a complete account of coherence-targeted text summarization and map out future extensions of my work.

6.1 Research Summary

This dissertation makes a systematic study of coherence and its modeling in automatic text summarization. I have argued that summary coherence is no less important than summary informativeness, and that the development of coherence-targeted summarization technology is much needed to satisfy human readers and advance the state of the art.

I set out by regarding coherence as an analyzable concept and exploring its multi-faceted and multi-disciplinary implications for text summarization. On the one hand, coherence is a textual effect that arises from different levels of content – shallow content represented by words, sentences, discourse units, etc. or deep content represented by domain-specific textual aspects, user-oriented speech acts,

etc. On the other hand, coherence is a cognitive construct driven by the human cognition of text comprehension. In this work, I have attempted to model coherence understood in all those dimensions – shallow content-driven coherence, deep content-driven coherence, and cognitive model-driven coherence.

Shallow content-driven coherence is theoretically rooted in lexical semantics and discourse analysis and works typically with micro-textual constructs such as words, phrases, sentences, and discourse units. It is computationally represented as a measure derived from literal information, such as entity overlap, word cohesion patterns, and sentence similarity. It can be used to order the summary-worthy sentences after they are selected. My efforts on single-document and multi-document summarization have shown that a proper modeling of this coherence renders better arranged sentences in the output summary, which leads to enhanced readability.

Deep content-driven coherence is theoretically situated in semantic and pragmatic accounts of macro-textual constructs such as news aspects or speech acts. Such deep content units are naturally related and, if appropriately organized, make a coherent text. On the other hand, deep content units are usually hidden or implied, which means most of the computational load is their automatic recognition. In my experimentation with newswire articles and Twitter posts, I have designed effective machine learning schemes to recognize news aspects and Twitter user speech acts. With a trained content model or speech act templates,

the recognized aspects or speech acts can be organized in a highly coherent way.

Cognitive model-driven coherence has its theoretical root in cognitive psychology. Accommodating human factors, the notion of coherence is extended to an extra-textual scope. I have introduced cognitive theories and models that interpret coherence as an inherent property in the process of text comprehension. A computational model that accounts for the cognitive mechanism of text comprehension is simultaneously a model of coherence. The model output is fed into a proposition-based summarization scheme for narrative text, which meets the needs of coherence and informativeness from the reader's perspective. The model's effectiveness has been tested on two kinds of narrative text: event-centric news and fairy tales.

I have shown that coherence represented as such can and should play vital roles in summarization, participating in both content selection and information ordering, producing extractive (sentence-level and proposition-level) as well as abstractive summaries. To evaluate coherence modeling under different circumstances, I experimented extensively with different genres of text: newswire, social media messages, fairy tales, etc. with satisfactory results.

6.2 Technical Highlights

My work is primarily motivated by improving summary readability that has received unduly less attention than summary informativeness. Different from most other works on coherence-oriented or "coherence-based" summarization,

my work places coherence in the foreground and explores its multidimensional nature. Coherence interpreted against different backgrounds calls for different modeling strategies. Most of the existing works on coherence in summarization dwell only on shallow content-driven coherence as a textual effect. My original work on deep content-driven coherence and cognitive model-driven coherence has charted new territories that hold promises for major breakthroughs in automatic summarization technology.

I have proposed new tasks, schemes, algorithms and made interesting findings when trying to model different kinds of coherence as an integral part of summarization. The following list itemizes the major highlights from my research.

- Applying shallow content-driven coherence to single-document news summarization, I developed a grouping-based algorithm to sentence ordering that leads to significantly more coherent summaries. My work shows that coherence-based ordering can improve on the default text ordering, which has never been addressed before.
- I have discovered that shallow content is not limited to simple entities, words or sentences. A composite shallow content unit – event – is employed in my effort to improve multi-document summary coherence. My work shows that event-enriched sentence information results in more coherent orderings than event-agnostic sentence information.
- One kind of deep content unit is news aspects, for which I have designed a

supervised learning approach with the aid of meta-phrases. A probabilistic model based on HMM is built to accommodate both aspect-biased sentence selection and aspect-based sentence ordering. My work based on aspect recognition is the first of its kind for aspect-guided summarization and I have proved the usefulness of aspects in generating coherent news summaries.

- The other kind of deep content unit is speech acts, which are recognized by a set of Twitter-oriented features and used to summarize Twitter posts. Drawing on coherence between speech acts and speech act-oriented summary templates, I have built a Twitter summarizer that significantly defeats all known rivals. My work on speech act-level coherence and abstractive Twitter summarization sheds new light on summarizing numerous, short, and noisy pieces of information.
- Borrowing theories and models from cognitive psychology, I reinterpret coherence as an inherent requirement by successful text comprehension. Proposition elements are fed to a cognitive model that draws on both stored knowledge and contextual information to update word activation degrees in a cyclic fashion. On narrative text, a proposition-level extractive summary is then generated by using the output of the model. My work has broken up a new promising field in coherence by looking beyond the text per se.

6.3 Future Directions

The current work on coherence-targeted text summarization can be extended in several directions. I now list some of the major ones in the following.

Although I have explored coherence applicable to text summarization from three dimensions – shallow content-driven, deep content-driven, and cognitive model-driven – we still lack a complete theory of coherence, or at least an integral account of its implications for summarization. Such a theory is necessary and possible, which awaits further research based on my preparatory modeling work and solid empirical evidences.

I have employed shallow content-driven coherence mainly for information ordering. Its application to sentence selection is worth exploring in the future. On single-document as well as multi-document summarization, ordering algorithms are primarily based on heuristic groups or blocks. In the future, statistical models and machine learning methods should be used to derive more empirically sound algorithms.

Deep content-driven coherence is not limited to genre-specific aspects or speech acts. More instances of this coherence, such as rhetorical roles and functional components, can be pursued in the future. Better machine learning methods, especially semi-supervised or transfer learning methods to address the lack of training data in real life, will be developed. For aspect-guided summarization, aspect-level (cf. proposition-level) extraction is presumably superior to sentence-level extraction. For Twitter summarization, template

induction should be automatized and local coherence between key phrases can be improved. Those are all promising directions.

The prototype model Chapter 6 is developed on is derived from (Kintsch, 1998). In future work, I will explore computerizing several other cognitive models and compare their effects. Many model parameters, now heuristically set, can be learned from annotated data or stochastic modeling. The proposition processing is a promising direction for finer-level extractive summarization, but better tree-adjustment algorithms as well as a good integration of proposition ranking with the cognitive model set future agendas for this line of research.

Finally, I have tried automatic, semi-automatic, and manual ways of summary evaluation in the experiments. What is urgently needed is a systematic and complete answer to summary coherence evaluation – specifying when and why some evaluation methods should be used for what kind of coherence. Standard models and tools like Pyramid and ROUGE are also to be developed for the advancement of coherence-based text summarization.

References

Alonso I, Alemany, L. and Fuentes F. M. 2003. Integrating cohesion and coherence for automatic summarization. In *Proceedings of EACL2003*, pages 1–8. Budapest, Hungary.

Anderson, J. R. 1976. *Language, Memory and Thought*. Mahwah, NJ: Erlbaum.

Aone, C., Okurowski, M. E., Gorlinsky, J., and Larsen, B. 1999. A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques. In I. Mani & M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. 71–80. Cambridge, Massachusetts: MIT Press.

Austin, J. 1962. *How to Do Things with Words*. Oxford: Oxford University Press.

Baccianella, S., Esuli, A., and Sebastiani, F. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 2200–2204.

Barzilay, R., Elhadad, N., and McKeown, K. 2002. Inferring Strategies for Sentence Ordering in Multidocument News Summarization. *Journal of Artificial Intelligence Research*, 17(1):35–55.

Barzilay, R. and Elhadad, M. 1997. Using Lexical Chains for Text Summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Barzilay, R. and Lapata, M. 2005. Modeling Local Coherence: An Entity-based

Approach. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 141–148. Ann Arbor.

Barzilay, R. and Lapata, M. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics*, 34(1):1–34.

Barzilay, R. and Lee, L. 2004. Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization. In *HLT-NAACL 2004: Proceedings of the Main Conference*, pages 113–120.

Barzilay R. and McKeown, K. 2005. Sentence Fusion for Multidocument News Summarization. *Computational Linguistics*, 31(3): 297–328.

Beaver, D. 2004. The Optimization of Discourse Anaphora. *Linguistics and Philosophy*, 27(1):3–56.

Berg-Kirkpatrick, T., Gillick D., and Klein D. 2011. Jointly Learning to Extract and Compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 481–490.

Berman, M. G. 2009. In Search of Decay in Verbal Short Term Memory. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 35(2):317–333.

Binwahlan, M. S., Salim, N., and Suanmali, L. 2009. Swarm Diversity Based Text Summarization. In Leung C. S., Lee, M., and Chan, J. H. (eds.) *ICONIP 2009, Part II, LNCS 5864*, pages 216–225, Berlin: Springer-Verlag.

Bird, S., Klein, E., Loper, E., and Baldridge, J. 2008. Multidisciplinary instruction with the Natural Language Toolkit. In *Proceedings of the Third*

Workshop on Issues in Teaching Computational Linguistics, pages 62–70, Columbus, Ohio.

Blair-Goldensohn, S. and McKeown, K. 2006. “Integrating Rhetorical-Semantic Relation Models for Query-focused Summarization”. In *DUC 2006*.

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4–5): 993–1022.

Blum-Kulka, S. 1986. Shifts in Cohesion and Coherence in Translation. In L. Venuti (ed.), *The Translation Studies Reader*. 2001, pp. 293-313. London and New York: Routledge.

Bollegala, D, Okazaki, N., and Ishizuka, M. 2006. A Bottom-up Approach to Sentence Ordering for Multi-document Summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 385–392. Sydney, Australia.

Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. 2004. Learning Multi-label Scene Classification, *Pattern Recognition*, 37(9):1757–1771.

Brandow, R., Mitze, K., and Rau, L. F. 1995. Automatic Condensation of Electronic Publications by Sentence Selection. *Information Processing and Management*, 31(5):675–685.

Brennan, S. E., Marilyn A. F., and Carl J. P. 1987. A Centering Approach to Pronouns. In *Proceedings of ACL 1987*, pages 155–162. Stanford, CA.

Brin, S. and Page, L. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. *Proceedings of the seventh international conference on*

World Wide Web 7, pages 107–117, Brisbane, Australia.

Budanitsky, A. and Hirst, G. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics* 32(1):13–47.

Cai, X., Li, W., Ouyang, Y., and Hong, Yan. 2010. Simultaneous Ranking and Clustering of Sentences: A Reinforcement Approach to Multi-document Summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 134–142.

Carbonell, J. and Goldstein, J. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *SIGIR-98*, pages 335–336.

Ceylan, H. and Mihalcea, R. 2009. The Decomposition of Human-written Book Summaries. In *Proceedings of the 10th international conference on computational linguistics and intelligent text processing*, pages 582–593.

Chklovski, T. and Pantel, P. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, pages 11–13. Barcelona, Spain.

Clarke, J. and Lapata, M. 2007. Modelling Compression with Discourse Constraints. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1–11. Prague.

Cohn, T. and Lapata, M. 2008. Sentence Compression Beyond Word Deletion. In

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 137–144. Manchester.

Conroy, J. M., Schlesinger, J. D., and Goldstein, J. 2006. CLASSY Tasked Based Summarization: Back to Basics. In *proceedings of the Document Understanding Conference (DUC-06)*.

Cristea, D., Ide, N., and Romary L. 1998. Veins Theory: A Model of Global Discourse Cohesion and Coherence. In *Proceedings of COLING/ACL'98*, pages 281–285. Montreal.

Cristea, D., Postolache, O., and Pistol, I. 2005. Summarisation through Discourse Structure. In *Proceedings of the computational linguistics and intelligent text processing, 6th International conference (CICLing 2005)*, pages 632–644.

Crystal, D. 2006. *Language and the Internet. 2nd edition.* Cambridge: Cambridge University Press.

Dang, H. T. 2005. Overview of DUC 2005. In *Proceedings of DUC 2005*, Gaithersburg, Maryland.

Dang, H. T. and Owczarzak, K. 2008. Overview of the TAC 2008 Update Summarization Task. In *Proceedings of TAC 2008*, Gaithersburg, Maryland.

Davies, D. L. and Bouldin, D. W. 1979. A Cluster Separation Measure. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1(2):224–227.

De Beaugrande, R., and Dressler W. 1996. *Introduction to Text Linguistics*. New York, 1996.

Dejong, G. 1982. An Overview of the FRUMP System. In Lehnert, W. G. and

Ringle, M. H. (eds.), *Strategies for Natural Languages Processing*. Hillsdale, NJ: Erlbaum.

Dunning, T. 1993. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1): 61–74.

Edmundson, H. P. 1969. New Methods in Automatic Abstracting. *Journal of the Association for Computing Machinery*, 16(2):264–285.

Elhadad, N. and McKeown, K. 2001. Towards Generating Patient Specific Summaries of Medical Articles. In *Proceedings of the NAACL-01*, pages 32–40.

Elsner, M., Austerweil, J., and Charniak E. 2007. A Unified Local and Global Model for Discourse Coherence. In *Proceedings of NAACL HLT 2007*, pages 436–443. Rochester, NY.

Endres-Niggemeyer, B. 1998. *Summarizing Information*. Berlin, Germany: Springer-Verlag.

Erkan, G. and Radev, D. 2004. LexRank: Graph-based Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22(1): 457–479.

Farzinder, A. and Lapalme, G. 2004. Legal Text Summarization by Exploration of the Thematic Structure and Argumentative Roles. In *Proceedings of ACL04*, pages 27–34.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, Massachusetts: MIT Press.

Filatova, E. and Hatzivassiloglou, V. 2003. Domain-Independent Detection, Extraction, and Labeling of Atomic Events. In *Proceedings of RANLP*, pages 145–152, Borovetz, Bulgaria.

Filatova, E. and Hatzivassiloglou, V. 2004. Event-Based Extractive Summarization. In *Proceedings of ACL-04*, pages 104–111.

Filippova, K. and Strobe, M. 2006. Using Linguistically Motivated Features for Paragraph Boundary identification. In *Proceedings of EMNLP 2006*, pages 267–274. Sydney.

Fillmore, C. J. 1985. Frames and the Semantics of Understanding. *Quaderni di Semantica*, VI(2):222–254.

Firmin, T. and Chrzanowski, J. 1999. An Evaluation of Automatic Text Summarization Systems. In I. Mani and M. T. Maybury (eds.), *Advances in Automatic Text Summarization*, 325–336. Cambridge, Massachusetts: MIT Press.

Foltz, P. W., Kintsch, W., and Landauer, T. K. 1998. Textual Coherence Using Latent Semantic Analysis. *Discourse Processes*, 25(2, 3):285–307.

Fuentes M, Alfonseca E, and Rodr ́guez H. 2007. Support Vector Machines for Query-focused Summarization Trained and Evaluated on Pyramid Data. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 57–60.

Fum, D., Guida, G., and Tasso, C. 1982. Forward and Backward Reasoning in

Automatic Abstracting. In *The Proceedings of COLING 82*, pages 83–88.

Ganesan, K., Zhai, C, and Han, J. 2010. Opinosis: A Graph-Based Approach to Abstractive Summarization of Highly Redundant Opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 340–348. Beijing, China.

Genest, P. and Lapalme, G. 2010. Text Generation for Abstractive summarization. In *Proceedings of the Third Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology.

Gernsbacher, M. A. 1990. *Language Comprehension as Structure Building*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Gernsbacher, M. A. 1991. Comprehending Conceptual Anaphors. *Language and Cognitive Processes*, 6(2):81–105.

Gernsbacher, M. A. 1996. Coherence Cues Mapping During Comprehension. In J. Costermans and M. Fayol (eds.), *Processing Interclausal Relationships in the Production and Comprehension of Text* (pp. 3–21). Mahwah, NJ: Lawrence Erlbaum Associates.

Gibbs, F. 1993. Knowledge-Based Indexing in SIMPR: Integration of Natural Language Processing and Principles of Subject Analysis in an Automated Indexing System. *Document and Text Management*, 1(2):131–153.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. 2011. Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments.

In *Proceedings of the Annual Meeting of the Association for Computational Linguistics, companion volume*, pages 42–47, Portland, OR.

Grosz, B. J., Aravind K. J., and Scott W. 1995. Centering: A framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21(2):203–225.

Grover, C., Hackey, B., and Korycinski, C. 2003. Summarizing Legal Texts: Sentential Tense and Argumentative Rules. In *Proceedings of the HLT-NAACL-03*, pages 33–40.

Hahn, U. 1990. TOPIC Parsing: Accounting for Text Macro Structures in Full-Text Analysis. *Information Processing and Management*, 26(1):135–170.

Halliday, M. A. K. and Hasan, R. 1976. *Cohesion in English*. London: Longman.

Harabagiu, S. and Lacatusu, F. 2002. Generating Single and Multi-Document Summarization with GISTexter. In *Proceedings of DUC 2002*, pages 30–38.

Hasler, L. 2004. An Investigation into the Use of Centering Transitions for Summarization. In *Proceedings of CLUK'04*, pages 100–107. Birmingham, UK.

Hirst, G. and St-Onge, D. 1998. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms. In Fellbaum, C. (ed.), *WordNet: An Electronic Lexical Database and Some of its Applications*, 305–332. Cambridge, Massachusetts: MIT Press.

Hobbs, J. 1985. On the Coherence and Structure of Discourse. *Report No.*

CSLI-85-37. Stanford, California: Center for the Study of Language and Information, Stanford University.

Hoey, M. 1991. *Patterns of Lexis in Text*. Oxford: Oxford University Press.

Hoffman, M. D., Blei, D. M., and Bach, F. 2010. Online Learning for Latent Dirichlet Allocation. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, A. Culotta (eds.), *Advances in Neural Information Processing Systems 23*, pp. 856.

Hovy, E. 1988. *Generating Natural Language under Pragmatic Constraints*. Hillsdale, NJ: Erlbaum.

Hovy, E. 2005. Automated Text Summarization. In R. Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*, pp. 583–598. Oxford: Oxford University Press.

Hovy, E. and Lin, C-Y. 1999. Automated Text Summarization in SUMMARIST. In I. Mani and M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. 81–94. Cambridge, Massachusetts: MIT Press.

Hovy, E., Lin, C-Y., and Zhou, L. 2005. Evaluating DUC 2005 Using Basic Elements. In *Proceedings of the Fifth Document Understanding Conference (DUC)*, Vancouver, Canada.

Inouye, D. 2010. Multiple Post Microblog Summarization. *REU Research Final Report*.

Jacobs, P. S. and Rau, L. F. 1990. SCISOR: Extracting Information from Online News. *Communications of the ACM*, 33(11):88–97.

Ji, H., Favre, B., Lin, W., Gillick, D., Hakkani-Tur, D., and Grishman, R. 2011. Open-domain Multi-document Summarization via Information Extraction: Challenges and Prospects. in *Multi-source, Multilingual Information Extraction and Summarisation Volume of "Theory and Applications of Natural Language Processing"*. Springer.

Jiang, J. and Conrath, D. 1997. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of ROCLING*, pages 19–33.

Jing, H. 1998. Summary Generation through Intelligent Cutting and Pasting of the Input Document. *Technical Report, Columbia University*.

Jing, H. 2000. Sentence Reduction for Automatic Text Summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 310–315.

Jing, H. and McKeown, K. 2000. Cut and Paste Based Text Summarization. In *Proceedings of the Sixth Applied Natural Language Conference (ANLP-00) and the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-00)*, pages 178–185.

Joachims, T. 1999. Transductive Inference for Text Classification using Support Vector Machines. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, pages 200–209.

Jones, K. S. 1999. Automatic Summarizing: Factors and Directions. In I. Mani and M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. pp. 1–12. Cambridge, Massachusetts: MIT Press.

Jurafsky, D. and Martin, J. H. 2009. *Speech and Language Processing, Second Edition*. Upper Saddle River, NJ: Pearson Education International.

Just, M. A. and Carpenter, P. A. 1992. A Capacity Theory of Comprehension: Individual Differences in Working Memory. *Psychological Review*, 99(1):122–49.

Karamanis, N., 2001. Exploring Entity-based Coherence. In *Proceedings of CLUK4*, pages 18–26.

Karamanis, N. and Mellish C. 2005. Using a Corpus of Sentence Orderings Defined by Many Experts to Evaluate Metrics of Coherence for Text Structuring. *In Proceedings of ENLG05*, pages 174–179.

Karamanis, N., Mellish, C., Oberlander, J., and Poesio M. 2004a. A Corpus-Based Methodology for Evaluating Metrics of Coherence for Text Structuring. In A. Belz et al. (eds.), *INLG 2004, LNAI 3123*. 90–99. Heidelberg, Berlin: Springer-Verlag.

Karamanis, N., Poesio M., Mellish C., and Oberlander, J. 2004b. Evaluating Centering-based Metrics of Coherence for Text Structuring Using a Reliably Annotated Corpus. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 391–398, Barcelona, Spain.

Kazantseva, A. 2006. An Approach to Summarizing Short Stories. In *Proceedings of the student research workshop at the 11th conference of the European chapter of the association for computational linguistics*, pages

55–62.

Kazantseva, A. and Szpakowicz, S. 2010. Summarizing Short Stories. *Computational Linguistics*, 36(1):71–109.

Kehler, A. 2002. *Coherence, Reference, and the Theory of Grammar*. Stanford, California: CSLI Publications.

Kibble, R. and Power, R. 2004. Optimizing Referential Coherence in Text Generation. *Computational Linguistics*, 30(4):401–416.

Kintsch, W. 1988. The Role of Knowledge in Discourse Comprehension: A Construction-Integration Model. *Psychological Reviews*, 95(2):163–182.

Kintsch, W. 1998. *Comprehension: A Paradigm for Cognition*. New York: Cambridge University Press.

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

Kintsch, W. and Mangalath, P. 2011. The Construction of Meaning. *Topics in Cognitive Science*, 3(2):346–370.

Klein, D. and Manning, C. D. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.

Knight, K. and Marcu, D. 2000. Statistics-Based Summarization — Step One: Sentence Compression. In *Proceedings of AAAI*, pages 703–710, Austin, Texas.

Knott, A., Oberlander J., O'Donnell M., and Mellish C. 2001. Beyond Elaboration: The Interaction of Relations and Focus in Coherent Text. In

Sanders, T., Schilperoord, J. and Spooren, W. (eds.), *Text Representation: Linguistic and Psycholinguistic Aspects*, 181–196. Benjamins.

Kupiec, J., Pedersen, J., and Chen, F. 1995. A Trainable Document Summarizer. In *Proceedings of the 18th International Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 68–73. Seattle, Washington: Association for Computing Machinery.

Landauer, T. and Dumais, S. 1997. A solution to Plato's problem: The Latent Semantic Analysis Theory of the Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, 104(2):211–240.

Landauer, T. K., Foltz, P. W., and Laham, D. 1998. Introduction to Latent Semantic Analysis. *Discourse Processes*, 25(2, 3):259–284.

Lapata, M. 2003. Probabilistic Text Structuring: Experiments with Sentence Ordering. In *Proceedings of the Annual Meeting of ACL*, pages 545–552. Sapporo, Japan.

Lapata, M. 2006. Automatic Evaluation of Information Ordering: Kendall's tau. *Computational Linguistics*, 32(4):1–14.

Lapata, M. and Barzilay, R. 2005. Automatic Evaluation of Text Coherence: Models and Representations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1085–1090. Edinburgh.

Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D. 2011. Stanford's Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task. In *Proceedings of the CoNLL-2011 Shared Task*,

pages 28–34.

Lehnert, W. G. 1999. Plot Units: A Narrative Summarization Strategy. In I. Mani and M. T. Maybury (eds.) *Advances in Automatic Text Summarization*, 177–214. Cambridge, Massachusetts: MIT Press.

Lemaire, B., Denhiere, G., Bellissens, C., and Jhean-Larose, S. 2006. A Computational Model for Simulating Text Comprehension. *Behavior Research Methods*, 38(4): 628–637.

Lemaire, B., Mandin, S., Dessus, P., and Denhi ère, G. 2005. Computational Cognitive Models of Summarization Assessment Skills, in *Proceedings of the 27th Annual Conference of the Cognitive Science Society (CogSci' 2005)*, pages 1266–1271.

Lesk, M. 1986. Automatic Sense Disambiguation: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 1986 Special Interest Group in Documentation*, pages 24–26.

Li, P., Wang, Y., Gao, W., and Jiang, J. 2011. 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.

Li, W., Wu, M., Lu, Q., Xu, W., and Yuan, C. 2006. Extractive Summarization Using Inter- and Intra- Event Relevance. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 369–376. Sydney.

Lin, C-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *ACL 2004 Workshop on Text Summarization Branches out , post-conference workshop of ACL 2004*, pages 74–81.

Lin, C-Y. and Hovy, E. 1997. Identifying Topics by Position. In *Proceedings of the Applied Natural Language Processing Conference*, 283–290, Washington, DC

Lin, C-Y. and Hovy, E. 2002. Automated Multi-document Summarization in NeATS. In *Proceedings of the Human Technology Conference 2002*, pages 50–53.

Lin, C-Y. and Hovy, E. 2003. Automatic Evaluation of Summaries Using N-gram Co-Occurrence Statistics. In *Proceedings of the Human Technology Conference 2003 (HLT-NAACL-2003)*, pages 71–78.

Liu, M., Li, W., Wu, M., and Lu, Q. 2007. Extractive Summarization Based on Event Term Clustering. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 185–188. Prague.

Liu, F., Liu, Y., and Weng, F. 2011. Why is "SXSW" Trending? Exploring Multiple Text Sources for Twitter Topic Summarization. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pages 66–75. Portland, Oregon.

Lobo, P. V. and de Matos, D. M. 2010. Fairy Tale Corpus Organization Using Latent Semantic Mapping and an Item-to-item Top-n Recommendation Algorithm, In *Language Resources and Evaluation Conference - LREC 2010,*

European Language Resources Association (ELRA) , pages 1472–1475, Malta.

Luhn, H. P. 1958. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

Madnani, N., Zajic, D., Dorr, B., Ayan, N. F., and Lin, J. 2007. Multiple Alternative Sentence Compressions for Automatic Text Summarization. In *Proceedings of the HLT/NAACL Document Understanding Conference Workshop*, Rochester, New York.

Mani, I. 2001. *Automatic Summarization*. Amsterdam/Philadelphia: John Benjamins.

Mani, I. and Bloedorn E. 1999. Summarizing Similarities and Differences among Related Documents. *Information Retrieval*, 1(1, 2):35–67.

Mani, I., Firmin, T., House, D., Klein, G., Sundheim, B., and Hirschman, L. 2002. The TIPSTER SUMMAC text summarization evaluation. *Natural Language Engineering*, 8(1):35–67.

Mani, I., Gates, B., and Bloedorn, E. 1999. Improving Summaries by Revising Them. In *Proceedings of ACL99*, pages 558–565, College Park, Maryland.

Mani, I. and Maybury M. 1999. *Advances in Automatic Text Summarization*. Cambridge, Massachusetts: MIT Press.

Mann, W. C. and Thompson, S. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8(3):243–281.

Marcu, D. 1997. The Rhetorical Parsing of Natural Language Texts. In

Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL-97), pages 96–103.

Marcu, D. 1999. Discourse Trees Are Good Indicators of Importance in Text. In I. Mani and M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. 123–136. Cambridge, Massachusetts: MIT Press.

Marcu, D. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, MA: The MIT Press.

Martins, A. F. T. and Smith, N. A. 2009. Summarization with a Joint Model for Sentence Extraction and Compression. In *Proceedings of the NAACL HLT Workshop on Integer Linear Programming for Natural Language Processing*, pages 1–9.

Maybury, M. 1999. Generating Summaries from Event Data. In I. Mani and M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. 265–281. Cambridge, Massachusetts: MIT Press.

McDonald, R. 2006. Discriminative Sentence Compression with Soft Syntactic Constraints. In *EACL-06*, pages 297–304.

McKeown, K., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J. L., Nenkova, A., Sable, C., Schiffman, B., and Sigelman, S. 2002. Tracking and Summarizing News on a Daily Basis with Columbia's Newsblaster. In *Proceedings of HLT 2002*, pages 280–285.

McKeown, K., Jordan, D., and Hatzivassiloglou, V. 1998. Generating Patient-Specific Summaries of Online Literature. In *Proceedings of the*

AAAI-98, pages 34–43.

McKeown, K. and Radev, D. R. 1995. Generating Summaries of Multiple News Articles. In *Proceedings, 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–82.

McKeown, K., Robin J., and Kukich, K. 1995. Generating Concise Natural language Summaries. *Information Processing and Management*, 31(5):703–733.

Mihalcea, R. 2004. Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization. In *The Companion Volume to the Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 170–173.

Mihalcea, R. 2006. Random Walks on Text Structures. In *CICLing 2006, LNCS 3878*, pages 249–262.

Mihalcea R. and Ceylan, H. 2007. Explorations in Automatic Book Summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 380–389.

Mihalcea, R. and Radev, D. 2011. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge: Cambridge University Press.

Mihalcea, R. and Tarau, P. 2004. TextRank: Bringing Order into Texts. In *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain.

Mitkov, R., Evans, R., Orasan, C., Ha, L. A., and Pekar, V. 2007. Anaphora

Resolution: To What Extent Does it Help NLP Applications? In *Proceedings of the 6th discourse anaphora and anaphor resolution colloquium*, pages 179–190.

Morales, L. P., Esteban, A. D., and Gervás, P. 2008. Concept-graph Based Biomedical Automatic Summarization Using Ontologies. In *Coling 2008: Proceedings of the 3rd textgraphs workshop on graph-based algorithms for natural language processing*, pages 53–56.

Morris, J. and Hirst, G. 1991. Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text. *Computational Linguistics*, 17(1):21–48.

Morris, A. H., Kasper, G. M., and Adams, D. A. 1992. The Effects and Limitations of Automated Text Condensing on Reading Comprehension Performance. *Information Systems Research*, 3(1):17–35.

Myaeng, S. H. and Jang, D-H. 1999. Development and Evaluation of a Statistically-Based Document Summarization System. In I. Mani and M. T. Maybury (eds.) *Advances in Automatic Text Summarization*, 61–70. Cambridge, Massachusetts: MIT Press.

Nahnsen, T. 2009. Domain-Independent Shallow Sentence Ordering. In *Proceedings of the NAACL HLT Student Research Workshop and Doctoral Consortium*, pages 78–83. Boulder, Colorado.

Nenkova, A. and Vanderwende, L. 2005. The Impact of Frequency on Summarization. *Technical Report MSR-TR-2005-101*, Microsoft Research,

Redmond, WA.

Nenkova, A., Vanderwende, L., and McKeown, K. 2006. "A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors That Influence Summarization". In *Proceedings of SIGIR'06*, pages 573–580. Seattle, Washington.

Okazaki, N., Matsuo, Y., and Ishizuka, M. 2004. Improving Chronological Ordering by Precedence Relation. In *Proceedings of 20th International Conference on Computational Linguistics (COLING 04)*, pages 750–756.

Orăsan, C. 2003. An evolutionary Approach for Improving the Quality of Automatic Summaries. In *Proceedings of the Multilingual Summarization and Question Answering — Machine Learning and Beyond Workshop*, pages 37–45. Sapporo, Japan.

Orăsan, C. 2007. Pronominal Anaphora Resolution for Text Summarisation. In *Proceedings of the recent advances on natural language processing*, pages 430–436.

Ouyang Y., Li, W., Li, Su., and Lu, Q. 2011. Applying Regression Models to Query-focused Multi-document Summarization. *Information Processing and Management*, 47(2): 227–237.

Ouyang, Y., Li, W., Lu, Q., and Zhang, R. 2010. A Study on Position Information in Document Summarization. In *COLING 2010: Poster Volume*, pages 919–927, Beijing.

Owczarzak, K. and Dang, H. T. 2011. Who Wrote What Where: Analyzing the

Content of Human and Automatic Summaries. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, pages 25–32.

Passonneau, R. J., Nenkova, A., McKeown, K., and Sigelman, S. 2005. Applying the Pyramid Method in DUC 2005. In *Proceedings of the Fifth Document Understanding Conference (DUC)*, Vancouver, Canada.

Poesio, M., Stevenson, R., Di Eugenio B., and Hitzeman, J. 2004. Centering: A Parametric Theory and Its Instantiations. *Computational Linguistics*, 30(3):309–363.

Polanyi, L., Culy, C., van den Berg, M., Thione, G. L., and Ahn, D. 2004. A Rule-Based Approach to Discourse Parsing. In *Proceedings of the Fifth SIGdial Workshop on Discourse and Dialogue, ACL*, pages 108–117.

Pollock, J. J. and Zamora, A. 1975. Automatic Abstracting Research at Chemical Abstracts Service. *Journal of Chemical Information and Computer Sciences*, 15(4):226–232.

Quesada, J. 2007. Creating your own LSA spaces. In T. K. Landauer, D. S. McNamara, S. Dennis, and W. Kintsch (eds.), *Handbook of Latent Semantic Analysis* (pp. 71–88). Mahwah, NJ: Erlbaum.

Radev, D., Jing, H., Styś, M., and Tam, D. 2004. Centroid-Based Summarization of Multiple Documents. *Information Processing and Management*, 40(6): 919–938.

Rand, W. M. 1971. Objective Criteria for the Evaluation of Clustering Methods.

Journal of the American Statistical Association (American Statistical Association), 66(336): 846–850.

Rath, G. J., Resnick, A., and Savage, T. R. 1961. The formation of Abstracts by the Selection of Sentences. *American Documentation*, 12(2):139–143.

Řehůřek, R. 2011. Subspace Tracking for Latent Semantic Analysis. In *Advances in Information Retrieval, volume 6611 of Lecture Notes in Computer Science*, pages 289–300. Springer.

Rich C. and Harper, C. 2007. *Writing and Reporting News: A Coaching Method, Fifth Edition*. Thomason Learning, Inc. Belmont, CA.

Riezler, S., King, T. H., Crouch, R., and Zaenen, A. 2003. Statistical Sentence Condensation Using Ambiguity Packing and Stochastic Disambiguation Methods for Lexical-Functional Grammar. In *HLT-NAACL-03*, pages 118–125. Edmonton, Canada.

Saggion, H. and Lapalme, G. 2002. Generating Indicative-Informative Summaries with SumUM. *Computational Linguistics*, 28(4):497–526.

Salton, G. 1971. *The SMART Retrieval System—Experiments in Automatic Document Processing*, Upper Saddle River, NJ, USA: Prentice-Hall, Inc..

Salton, G., Singhal, A., Mitra M., and Buckley, C. 1997. Automatic Text Structuring and Summarization. *Information Processing and Management*, 33(2):193–207.

Schank, R. C. and Abelson, R. P. 1977. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum.

Schilder, F. and Kondadadi, R. 2008. FastSum: Fast and Accurate Query-based Multi-document Summarization. In *Proceedings of ACL-08: HLT, short papers*, pages 205–208.

Searle, J. 1975. *Indirect speech acts*. In P. Cole and J. Morgan (eds.), Syntax and semantics, vol. iii: Speech acts (pp. 59–82). New York: Academic Press.

Sharifi, B., Hutton, M-A., and Kalita, J. 2010a. Summarizing Microblogs Automatically. In *Proceedings of HLT/NAACL* 2010, pages 685–688.

Sharifi, B., Hutton, M-A., and Kalita, J. 2010b. Experiments in Microblog Summarization. In *Proceedings of IEEE Second International Conference on Social Computing*, pages 49–56.

Silber, H. G. and McCoy, K. F. 2000. Efficient Text Summarization Using Lexical Chains. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*, pages 252–255. New Orleans, Louisiana.

Soricut, R. and Marcu D. 2006. Discourse Generation Using Utility-Trained Coherence Models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 803–810.

Sperber, D. and Wilson, D. 1995. *Relevance: Communication and Cognition, Second Edition*, Oxford/Cambridge: Blackwell Publishers.

Steibach, M., Karypis, G., and Kumar V. 2000. A Comparison of Document Clustering Techniques. *Technical Report 00-034.* Department of Computer Science and Engineering, University of Minnesota.

Steinberger, J, Poesio, M., Kabadjov, M. A., and Ježek, K. 2007. Two Uses of

Anaphora Resolution in Summarization. *Information Processing and Management*, 43(6):1663–1680.

Stevenson, M. and Greenwood, M. A. 2005. A Semantic Approach to IE Pattern Recognition. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 379–386.

Steyvers, M., and Griffiths, T. L. 2008. Rational Analysis as a Link between Human Memory and Information Retrieval. In N. Chater and M. Oaksford (eds.), *The probabilistic mind: Prospects for a Bayesian Cognitive Science* (pp. 329–350). Oxford, England: Oxford University Press.

Strube, M. and Hahn, U. 1999. Functional Centering: Grounding Referential Coherence in Information Structure. *Computational Linguistics*, 25(3):309–344.

Strzalkowski, T., Stein, G., Wang, J., and Wise, B. 1999. A Robust Practical Text Summarizer. In I. Mani and M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. 137–154. Cambridge, Massachusetts: MIT Press.

Taboada, M. and Wiesemann, L. 2010. Subjects and Topics in Conversation. *Journal of Pragmatics*, 42(7):1816–1828.

Tan, Y. F., Kan, M., and Cui, H. 2006. Extending Corpus-based Identification of Light Verb Constructions Using a Supervised Learning Framework. In *Proceedings of the EACL 2006 Workshop on Multi-word-expressions in a multilingual context*, pages 49–56, Trento, Italy.

Tapiero, I. 2000. *Construire une représentation mentale cohérente: Structures,*

relations et connaissances [Building a Coherent Mental Representation: Structures, Relations, and Knowledge]. Postdoctoral thesis for the "Habilitation á diriger des recherches", University of Lyon 2, Lyon, France.

Tapiero, I. 2007. *Situation Models and Levels of Coherence: Towards a Definition of Comprehension*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

Teufel, S. and Moens, M. 1999. Argumentative Classification of Extracted Sentences as a First Step towards Flexible Abstracting. In I. Mani and M. T. Maybury (eds.), *Advances in Automatic Text Summarization*. 155–171. Cambridge, Massachusetts: MIT Press.

Teufel, S., and Moens, M. 2002. Summarizing Scientific Articles: Experiments with Relevance and Rhetorical Status. *Computational Linguistics*, 28(4): 409–445.

Thione, G. L., van den Berg, M., Polanyi, L., and Culy, C. 2004. Hybrid Text Summarization: Combining External Relevance Measures with Structural Analysis. In the *Proceedings of the ACL2004 Workshop*, pages 51–55. Barcelona, Spain.

Tsoumakas, G. and Katakis, I. 2007. Multi label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13.

Unger, C. 2006. *Genre, Relevance and Global Coherence: The Pragmatics of Discourse Type*. New York: Palgrave Macmillan.

van den Broek, P., Risden, K., Fletcher, C. R., and Thurlow, R. 1996. A

‘Landscape’ View of Reading: Fluctuating Patterns of Activation and the Construction of a Stable Memory Representaion. In B. K. Britton & A. C. Graesser (eds.), *Models of Understanding Text*, 165–187, Mahwah, NJ: Erlbaum.

van Dijk, T. A. and Kintsch, W. 1983. *Strategies of Discourse Comprehension*. New York: Academic Press.

Vanderwende, L., Suzuki, H., Brockett, C., and Nenkova, A. 2007. “Beyond SumBasic: Task-Focused Summarization with Sentence Simplification and Lexical Expansion”. *Information Processing and Management* 43(6):1606–1618.

Vapnik, V. 1998. *Statistical Learning Theory*. New York: John Wiley & Sons.

Wan, X. and Yang, J. 2008. Multi-Document Summarization Using Cluster-Based Link Analysis. In *Proceedings of SIGIR-08*, pages 299–306.

Wang, L., Shen, X., and Pan, W. 2007. On Transductive Support Vector Machines. In J. Verducci, X. Shen, and J. Lafferty (eds.), *Prediction and Discovery*. American Mathematical Society.

Wei, F., Li, W., Qin, L., and He, Y. 2009. A Document-sensitive Graph Model for Multi-document Summarization. In *Knowledge and Information Systems*, 22(2):245–259.

Wierzbicka, A. 1987. *English Speech Act Verbs: A Semantic Dictionary*. Orlando: Academic Press.

Wilpon, J. G. and Rabiner, L. R. 1985. A Modified K-means Clustering

Algorithm for Use in Isolated Word Recognition. In *Proceedings of IEEE Trans. Acoustics, Speech, Signal, ASSP-33,* pages 587–594.

Wilson, T. and Wiebe, J. 2003. Identifying Opinionated Sentences. In *Proceeding of NAACL 03*, pages 33–34.

Wolf, F. and Gibson, E. 2004. Paragraph-, Word-, and Coherence-based approaches to Sentence Ranking: A Comparison of Algorithm and Human Performance. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pages 383–390. Barcelona, Spain.

Wolf, F. and Gibson, E. 2006. *Coherence in Natural Language*. Cambridge, MA: MIT Press.

Wong, K., Wu, M., and Li, W. 2008. Extractive Summarization Using Supervised and Semi-supervised Learning. In *Proceedings of the 22nd international conference on computational linguistics*, pages 985–992.

Wu, Z. and Tseng, G. 1993. Chinese Text Segmentation for Text Retrieval Achievements and Problems. *JASIS*, 44(9):532–542.

Yih, W., Goodman, J., Vanderwende, L., and Suzuki, H. 2007. "Multi-Document Summarization by Maximizing Informative Content-Words". In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1776–1782.

Yousfi-Monod, M. and Prince V. 2008. Sentence Compression as a Step in Summarization or an Alternative Path in Text Shortening. In *COLING 2008*, pages 139–142. Manchester, UK.

Zajic, D. M., Dorr, B. J., Lin, J. 2008. Single-document and Multi-document Summarization Techniques for Email Threads Using Sentence Compression. *Information Processing and Management* 44(4): 1600–1610.

Zha, H. 2002. Generic Summarization and Key Phrase Extraction using Mutual Reinforcement Principle and Sentence Clustering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 113–120.

Zhang, R., Li, W., and Lu, Q. 2010. Sentence Ordering with Event-Enriched Semantics and Two-Layered Clustering for Multi-Document News Summarization. In *COLING 2010: Poster Volume*, pages 1489–1497, Beijing.

Zhang, R., Ouyang, Y., and Li, W. 2011. Guided Summarization with Aspect Recognition. In *Proceedings of Textual Analysis Conference* (*TAC 2011*).

Zhao, X. and Jiang, J. 2011. An Empirical Comparison of Topics in Twitter and Traditional Media. *Technical Report*, Singapore Management University School of Information Systems.

Zwaan, R. A., Langston, M. C., and Graesser, A. C. 1995. The Construction of Situation Models in Narrative Comprehension: An Event-Indexing Model. *Psychological Science*, 6(5):292–297.