ABSTRACT

ZHANG, ZHE. Text Mining for Sentiment Analysis. (Under the direction of Professor Munindar P. Singh.)

Over the past few years, with the development of web services and the emergence of user-driven social media, more and more people express their sentiments publicly, generating a large amount of opinionated data. Sentiment analysis is such a research field that aims at developing automated approaches to accurately extract sentiments from opinionated data. Researchers have devoted considerable attention to this field. However, due to the complexity and diversity of linguistic expressions, we are still far from a satisfying solution.

In this dissertation, we have identified four challenges that may hinder current research progress: basic sentiment expressing unit, paucity of labeled data, domain dependence, and author modeling. Accordingly, we propose two approaches, *ReNew* and *Arch*, to address these challenges.

ReNew, a semi-supervised sentiment analysis framework, can leverage unlabeled opinionated data to automatically generate a domain-specific sentiment lexicon and trains a sentiment classifier. The domain-specific sentiment lexicon uses dependency relation pairs as its basic elements to capture the contextual sentiment of words. The sentiment classifier leverages relationships between consecutive sentences, clauses, and phrases to infer sentiments. We evaluate the effectiveness of ReNew using a hotel review dataset. Empirical results show that ReNew greatly reduces the human effort for building a domain-specific sentiment lexicon with high quality. Specifically, in our evaluation, working with just 20 manually labeled reviews, it generates a domain-specific sentiment lexicon that yields weighted average F-Measure gains of 3%. The sentiment classifier achieves approximately 1% greater accuracy than a state-of-the-art approach based on elementary discourse units.

Arch, a probabilistic model for unsupervised sentiment analysis, can discover sentiment-aspect pairs from unlabeled opinionated data. By incorporating authors explicitly as a factor, Arch can capture the association of sentiments and aspects with authors. The generated interpretable author profiles can be used for (1) summarizing authors' preferences in terms of sentiments and aspects and (2) measuring similarities among authors. To assess the generalizability of Arch, we use four datasets in two domains for evaluation. Results show that Arch successfully discovers sentiment-aspect pairs with higher semantic coherence than those generated by state-of-the-art approaches. The author profiles are well correlated with ground truth. To exhibit the prospects for potential applications, we demonstrate the effectiveness of Arch for authorship attribution and document-level sentiment classification.

In both ReNew and Arch, we use *segments* as basic sentiment expressing units to capture

fine-grained sentiments. We also present a rule-based segmentation algorithm based on discourse relations.

Text Mining for Sentiment Analysis

by
Zhe Zhang

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2014

APPROVED BY:

Dr. Christopher G. Healey	Dr. Steffen Heber
Dr. James C. Lester	Dr. Munindar P. Singh Chair of Advisory Committee

DEDICATION

To my parents.

BIOGRAPHY

Zhe Zhang was born in Beijing, China in 1984. He attended High School of Peking University in his hometown, where he graduated in 2003. In 2007, He received a Bachelor of Engineering degree in Computer and Information Science from Renmin University of China. He then join National Laboratory of Pattern Recognition at Chinese Academy of Sciences as a software engineer until 2009, when he started PhD studies at North Carolina State University. He obtained a Master of Science degree from North Carolina State University in 2012. He spent the summer of 2012 and 2013 as an intern at IBM Watson Solutions. His research interests include sentiment analysis, text mining, natural language processing, probabilistic models, machine learning, social network analysis, and social computing.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Munindar Singh, for his continuous support and valuable guidance through my years at NCSU. I am so fortunate to be one of his students. From the very beginning of my graduate studies, it is him who ignited my passion for research. He taught me how to conduct research carefully with enthusiasm and inspired my interest in sentiment analysis. His professional character and personality have deeply influenced me. Working with him makes my PhD studies much more memorable and enjoyable.

I would like to express my sincere appreciation to my committee members: Prof. Christopher Healey, for guiding me through this research and all of the insightful comments; Prof. Steffen Heber, for his invaluable suggestions and all of our engaging discussions on various topics; and Prof. James Lester, for his influences in various aspects of my research and his encouragement throughout my graduate studies. I am also sincerely grateful to Camille Cox, Kathy Luca, Margery Page, Prof. Douglas Reeves, Prof. George Rouskas, Andrew Sleeth, and Prof. David Thuente for their advice and support throughout my studies.

I am especially indebted to Prof. Jianhua Tao at Chinese Academy of Sciences, who initiated my first steps in research when I was in high school and who guided me to my first publication. He has remained a constant source of guidance and encouragement throughout my research.

I have had the privilege of working with many extraordinarily talented people during my internship in IBM Watson. Many thanks to them. A special thanks to Norma Hale, Jeff Jablonowski, Glyn Tomkins, and Rob Yates who made it possible for me to procure a IBM PhD Watson Solutions scholarship and the prestigious IBM PhD fellowship two times. These fellowships enable me to pursue my research interests and career endeavors.

During my PhD studies, I had the luck to work with Nirav Ajmeri, Adel ElMessiry, Anup Kalia, Chung-Wei Hang, Christopher Hazard, Pradeep Murukannaiah, and Guangchao Yuan. A special thanks to Chung-Wei Hang, who provided me a great deal of advice and knowledge in my research and Scott Gerard who helped me find my career path. I would also like to extend thanks to the other members in our group for their valuable help, including Xibin Gao, Dhanwant Singh Kang, Prashant Kediyal, and Pankaj Telang. In addition, I have been blessed to work with the wonderful collaborators outside of NC State, including Prof. Ramesh Govindan and Bin Liu at University of Southern California and Prof. Brian Uzzi at Northwestern University.

Many thanks in particular are due to Huan Lian, who have always been there for me. Her constant support makes all goals attainable. I also wish to thank my friends here in the US and in China, especially, Vera Axelrod, John Colby, Xi Ge, Malcolm Greaves, Yu Mao, Dongqiuye Pu, Rob Rrua, Beata Strack, Vikrant Verma, Minglei Wang, Bo Wang, Pu Yang, Xiaowan Yang,

Wente Zeng, and Yunteng Zhang. Without them, my life would not be as valuable, enjoyable, and worthwhile as it is today.

Last, but the most important, none of this would have been possible at all were it not for my parents. My mom has made countless sacrifices over the years to provide me with the best of everything. She is always a role model for me. My dad has always encouraged me to pursue my goals. Thank you, Mom and Dad! I would also like to thank my grandparents, my uncle and aunt, my sisters, my brothers, and the rest of my family for loving and supporting me at all times.

This dissertation is supported by the Army Research Laboratory in its Network Sciences Collaborative Technology Alliance (NS-CTA) under Cooperative Agreement Number W911NF-09-2-0053 and by an IBM PhD Scholarship and two IBM PhD fellowships.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Introduction

Sentiments describe people's opinion, attitude, emotion, or judgment toward entities such as products, topics, events, or people. They play a critical role in our decision making process. In the past, when we need to make decisions related to an entity, we often ask our family or friends' sentiments toward that entity. When companies, organizations, or governments need to make decisions, they rely on conducting surveys and polls using numerous tools and techniques extensively developed in the 20th century.

Over the past few years, with the rapid development of social media, more and more people express their sentiments publicly, generating a large amount of opinionated text on a great variety of topics. The availability of these opinionated data has almost changed our way of decision making process. For example, nowadays, when we want to buy products, book hotels, or choose restaurants, we are no longer limited to asking the sentiments from our family or friends because there are billions of reviews available publicly. For companies, organizations, or governments, to collect public sentiments, conducting survey or polls is not the only way and is not as important as before because the opinionated data have already been created by people in various forms, e.g., microblogs, comments, or reviews.

Tremendous amount of opinionated data provide us all the information we need. Meanwhile, it also generates a series of new problems: how to effectively process these data? What information is useful? How to distill the information? Finally, what conclusion regarding sentiments can we draw from the data?

Solving these complex problems is far beyond human capabilities, especially with high-volume data. Consider, for example, according to the fact sheet of a well-known travel websites, TripAdvisor [TripAdvisor, 2014c],

- Their branded websites contain over 125 million reviews

- The average number of reviews is 139 for a hotel in the popular destinations
- For every minute, more than 80 new contributions are posted by their members
- For every day, nearly 2,800 new topics on average are posted by their members

From the above statistics, we see that even for one particular source of opinionated data, we are overwhelmed by the explosion of information. Suppose we need to make a choice among a set of candidate hotels, each of which is associated with 139 reviews, assume we could read a review in five minute, we will need about 58 hours to arduously go through all of the 695 reviews. We see that even for making one simple decision, it takes too long for us to read all of the information, not to mention how to select useful information.

We need to develop new techniques that can automatically process opinionated data, extract sentiments, and help us make decisions. However, this is a highly challenging task. With recent advances of machine learning techniques, more and more researchers have turned their attention to this task. Consequently, a new field of study, sentiment analysis (also known as opinion mining), has emerged and become a very active research area since the year 2000 [Liu, 2012]. Until now, researchers have published hundreds of papers that cover a large spectrum of topics, ranging from theoretical problems to practical applications. We also notice that the impact of sentiment analysis is not only on computer science and natural language processing, but also on social science, management sciences, political science, and economics [Tumasjan et al., 2010; Bollen et al., 2011].

As a research field that could generate great business value, sentiment analysis has also drawn considerable attention from industry community. According to [Liu, 2012], about 40–60 sentiment analysis startups in the US is providing their sentiment analysis solutions to financial institutions, governments, and businesses. More and more leading companies now have released their sentiment analysis tools, such as Google, Microsoft, IBM, SAS, and SAP.

One of the major tasks in sentiment analysis is sentiment classification. The goal is to develop automated approaches that can classify sentiments expressed in texts as positive, neutral, or negative. Target texts can be documents, paragraphs, or sentences. Most of existing approaches [Pang et al., 2002; Kennedy and Inkpen, 2006; McDonald et al., 2007; Rentoumi et al., 2012] focus on extracting various features from texts and then applying supervised learning techniques to learn classifiers. Due to the paucity of labeled data, there has recently been a trend towards developing approaches that can reduce human effort. As a result, we see more and more advanced approaches that use semi-supervised [Socher et al., 2011; Glorot et al., 2011] or unsupervised learning techniques[Jo and Oh, 2011; Lin et al., 2012; Kim et al., 2013]. Another important task in sentiment analysis is sentiment lexicon generation. The goal of this task is to build a lexicon containing words typically used for conveying sentiments. Sentiments

of these individual words provide basic key clues for revealing sentiments of longer texts, such as sentences or documents. Also, such lexicons can facilitate people to exploit sentiments in their domains without learning extensive knowledge of natural language processing. For example, SentiWordNet [Esuli and Sebastiani, 2006] is a sentiment lexicon that is widely used in many domains, such as location services [Cheng et al., 2011], link prediction [Yuan et al., 2014], and rating prediction [Siersdorfer et al., 2010].

1.2 Challenges

Although sentiment analysis has been thoroughly studied for many years, until today we are still seeking satisfying solutions. We have identified the following four challenges that may hinder current research progress:

Basic Sentiment Expressing Unit

A basic sentiment expressing unit is a sequence of words that represents at most one sentiment. In opinionated data, people usually use unstructured texts as the vehicle to express their sentiments. Sentiments can change within one document, one sentence, or even one clause. Therefore, the boundary of a basic sentiment expressing unit is often not fixed. For instance, consider the following three sentences from TripAdvisor.

Sentence 1: *Great front desk staff* [TripAdvisor, 2014a].

Sentence 2: *Wireless internet access is free but somewhat slow* [TripAdvisor, 2010].

Sentence 3: *Not much just basic room setup but location is great* [TripAdvisor, 2009].

Sentence 1 contains one aspect, *staff*, associated with one sentiment. Sentence 2 expresses two sentiments toward one aspect, *internet*. Sentence 3 describes two aspects, *room* and *location*, with two sentiments associated with each aspect, respectively.

The step of identifying basic sentiment expressing units is fundamental for developing fine-grained sentiment analysis system. However, Most of existing works falsely assume that one document or sentence holds one sentiment. This results in a substantial performance loss for their models.

Paucity of Labeled Data

Most of the existing sentiment analysis systems heavily rely on labeled data for training. The size and quality of labeled data has a large impact on the performance of their systems. However, there are very few labeled data available. Therefore, in order to make their system working

on a new domain, the first step is often acquiring labels. This process is labor intensive and error prone. For instance, we take two days of effort by six people to label only 200 of hotel reviews. The paucity of labeled data has limited the generalizability of existing sentiment analysis systems.

Domain Dependence

Sentiment analysis is a domain dependent problem. A sentiment classifier trained using datasets from one domain usually performs poorly when testing on datasets from the other domains. The domain dependence problem is caused by two reasons.

First, in opinionated texts, same expressions may represent different sentiments in different domains. For example, consider two sentences from reviews in Amazon and TripAdvisor, respectively.

> Sentence 1: *This speaker is very **small** and easy to store in my laptop case* [Amazon, 2014].
>
> Sentence 2: *The bathroom is very **small**, two people can't stand in the bathroom at the same time* [TripAdvisor, 2014b].

We see word *small* represents a positive sentiment in Sentence 1 to describe a speaker, while it is used for expressing a negative sentiment toward a bathroom in Sentence 2.

Second, people are likely to use different words or phrases to express sentiments in different domains. Although there are a number of general sentiment words, such as *good*, *amazing*, *bad*, and so on, people tend to use more specific words or phrases to precisely describe their sentiments in specific domains. For example, in restaurant reviews, people often use words, such as *crispy* and *chewy*, to describe the crust on the pizza. In hotel reviews, words, such as *attentive* and *rude* are frequently used to comment hotel service.

Domain dependence restricts the role of sentiment lexicons for cross-domain sentiment analysis. It is almost impossible to build an universal lexicon that can cover all domains. General sentiment lexicons, such as ANEW and LIWC are lack of domain-specific expressions. Expanding these sentiment lexicons to domain-specific domain is a non-trivial task that requires knowledge from domain experts. Also, most of existing lexicons simply use words as their basic elements. These lexicons ignore the contextual information surround words.

Modeling Author Information

Opinionated data are often contributed by a number of authors (opinion holders). Different authors may express sentiments with respect to different aspects of the same or similar entities. For example, when reviewing a hotel, a business traveler may consider aspects such as *Internet*

access, *Concierge services*, and *Room*, whereas a tourist may consider aspects such as *Nearby area*, *Dining*, and *Room*. Even for the same aspect, authors may vary the emphasis they place on it. For example, aspects *Comfort* may be the most important factor for one author to assess hotels, while it may not be the primary focus for another author.

Capturing authors' preferences in opinionated data is important for improving the performance of sentiment analysis system. Also, understand subtle similarities and differences among authors would enable a wide range of applications. For example, we could use authors' preferences to provide sentiment summarization. With summarized authors' preferences at hand, we could produce recommendations for services that are better aligned with the expectations of a particular user. Such alignment may be inferred from other users who consider the same aspects important as the given user.

1.3 Contributions

In this dissertation, we focus on the problem of developing the general and robust sentiment analysis system that can address the challenges discussed above. The major contributions of this dissertation are as follows.

- Instead of using documents, sentences, or clauses as basic units for sentiment analysis, we advocate the use of *segments* as basic sentiment units in sentiment analysis. Segments are shorter than sentences and therefore can help capture fine-grained sentiments. The flexibility of the boundaries makes segments more powerful than sentences or clauses to handle the diversity of sentiments representation. We present a rule-based segmentation algorithm based on discourse relations among phrases and clauses.

- To alleviate the problem of lacking labeled data, our approaches leverage unlabeled opinionated data. There are three advantages of using unlabeled data. First, unlabeled data exist in a wide range of domains and they are easy to obtain. For example, our crawler uses only five days to collect a hotel review dataset that contains about 500,000 unlabeled reviews. Second, compared with labeled data, unlabeled data inherently carry more information. They provide broad coverage of various types of sentiment expressions and therefore minimize bias. Third, unlabeled data require minimum human efforts to process.

- To address the domain dependence, our approach automatically build domain-specific lexicons. The lexicons are better fit for domain-specific sentiment analysis than general lexicons. The building process requires minimum human supervision. Our lexicon use dependency relation pairs to as its basic elements to capture contextual information surround words.

- To model the factor of authors in opinionated data, our approach incorporates author information during its training process. It automatically generate authors' preferences profiles. Such profiles can be used to measure author similarities.

1.4 Organization

The rest of this dissertation is organized as follows. Chapter 2 lays out a semi-supervised sentiment analysis framework named ReNew, illustrates its components, and presents our experiments and results. Chapter 3 describes an unsupervised sentiment analysis model named Arch, illustrates an inference method based on collapsed Gibbs sampling, and empirically evaluates the effectiveness of Arch. Chapter 4 summarizes our approaches and outlines some directions for future work.

Chapter 2

ReNew: A Semi-Supervised Framework for Generating Domain-Specific Lexicons and Sentiment Analysis

2.1 Introduction

Over the past few years, with the rapid development of social media, more and more people are making their sentiments publicly available, generating a large amount of opinionated text. Automatically processing and accurately extracting sentiments is important in building applications such as text summarization, rating prediction, and personalized recommendation. However, the complexity and diversity of linguistic representations for sentiments make this problem challenging.

One major direction in sentiment analysis is sentiment lexicon generation. High-quality sentiment lexicons can improve the performance of sentiment analysis models over general-purpose lexicons [Choi and Cardie, 2009]. More advanced methods such as [Kanayama and Nasukawa, 2006] adopt domain knowledge by extracting sentiment words from the domain-specific corpus. However, depending on the context, the same word can have different polarities even in the same domain [Liu, 2012].

Sentiment classification is another major direction in sentiment analysis. The goal of this task is to train a classifier to predict a sentiment label for a document, sentence, or text snippet. Researchers have proposed a number of approaches. Pang et al. [2002] infer the sentiments using basic features, such as bag-of-words. To capture more complex linguistic phenomena, leading

approaches [Nakagawa et al., 2010; Jo and Oh, 2011; Kim et al., 2013] apply more advanced models but assume one document or sentence holds one sentiment. However, this is often not the case. Sentiments can change within one document, one sentence, or even one clause. Also, existing approaches infer sentiments without considering the changes of sentiments within or between clauses. However, these changes can be successfully exploited for inferring fine-grained sentiments. Moreover, most of existing supervised and semi-supervised approaches typically require high-quality labeled data to train classifiers with good accuracy. However, building labeled data is an error prone and labor intensive task.

To address the above shortcomings of lexicon and granularity, we propose a semi-supervised framework named ReNew.

1. Instead of using sentences, ReNew uses *segments* as the basic units for sentiment classification. Segments can be shorter than sentences and therefore help capture fine-grained sentiments.

2. ReNew leverages the relationships between consecutive segments to infer their sentiments and automatically generates a domain-specific sentiment lexicon in a semi-supervised fashion.

3. To capture the contextual sentiment of words, ReNew uses dependency relation pairs as the basic elements in the generated sentiment lexicon.

The upper part of Figure 2.1 shows a hotel review from Tripadvisor [TripAdvisor, 2011]. We split it into 12 segments with sentiment labels. The bottom part of Figure 2.1 visualizes the sentiment changes within the text. The sentiment remains the same across Segments 2 to 5, Segments 9 to 10, and Segments 11 to 12. Of the six sentiment transitions, four are caused by changing topics. The remaining two transitions are indicated by transition cues. For example, Segment 6 starts with *but*—which signals conflict, contradiction, or dismissal. We also notice that some transition cues indicate a continuation of sentiments. For example, Segment 4 expresses the same sentiment as Segment 3 by starting with *even*—which signals emphasis.

Assuming we know Segment 5 is positive, given the fact that Segment 6 starts with *but*, we can infer with high confidence that the sentiment in Segment 6 changes to neutral or negative even without looking at its content. After classifying the sentiment of Segment 6 as NEG, we associate the dependency relation pairs {*sign*, *wear*} and {*sign*, *tear*} with that sentiment. Similarly, assume we know Segment 3 is positive, given the fact that Segment 4 starts with *even*, we can predict, with high confidence, that the sentiment in Segment 4 stays same as in Segment 3.

ReNew can greatly reduce the human effort for building a domain-specific sentiment lexicon with high quality. Specifically, in our evaluation on two real datasets, working with just 20 man-

Review:

(1: NEU) We stayed here for one night before leaving on a cruise out of the San Pedro port. **(2: POS)** The hotel was clean and comfortable. **(3: POS)** Service was friendly **(4: POS)** even providing us a late-morning check-in. **(5: POS)** The room was quiet and comfortable, **(6: NEG)** but it was beginning to show a few small signs of wear and tear. **(7: POS)** The pool area was well-kept with plenty of fresh towels and lounge chairs available. **(8: NEG)** Room service breakfast was subpar even for a three-star hotel **(9: NEU)** , so skip that in favor of Think Cafe just up the street and around the corner. **(10: NEU)** There are many local shops and restaurants in the neighborhood around the hotel if you're willing to walk a few blocks and explore. **(11: POS)** The free shuttle service to the cruise terminal is also a nice perk. **(12: POS)** All in all, a solid choice for a stay of just a night or two.

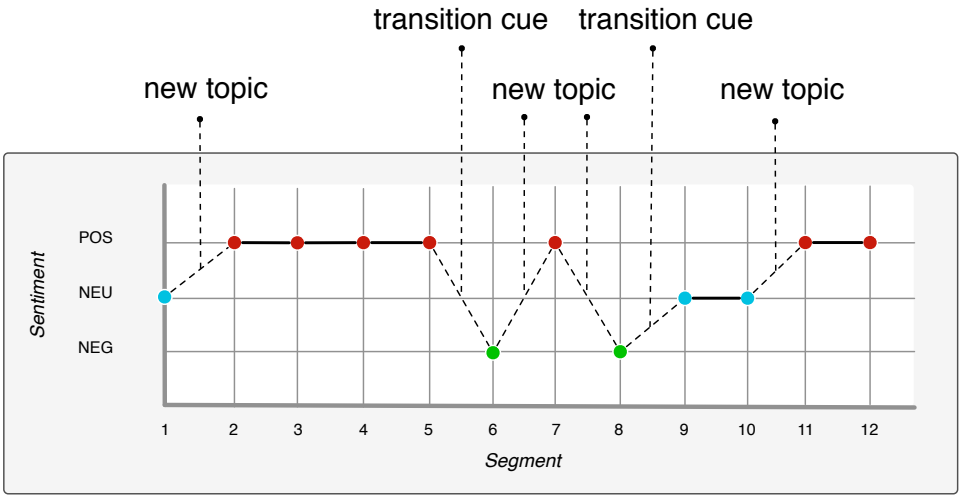


Figure 2.1: Segments in a Tripadvisor review.

ually labeled reviews, ReNew generates a domain-specific sentiment lexicon that yields weighted average F-Measure gains of 3%. Additionally, our sentiment classification model achieves approximately 1% greater mean accuracy than a state-of-the-art approach based on elementary discourse units [Lazaridou et al., 2013].

The rest of this chapter is structured as follows. Section 2.2 defines our problem. In Section 2.3 presents an overview of ReNew. Section 2.4–2.7 illustrate components of ReNew. Section 2.8 presents our experiments and results. Section 2.9 reviews the relevant literature. Section 2.10 concludes this chapter with a discussion of future work.

2.2 Problem Definition

In this section, we define our problem in formal. Let's consider the review in Figure 2.1 again. It consists of 12 segments, where each segment is associated with a sentiment label. The changes of sentiments are related to either changes of topics or transition cues. Also, a number of opinion words appear in this review. For example, *clean* and *comfortable* are positive words to define hotels and *wear* and *tear* are negative words to describe rooms. The goal of ReNew is to automatically extract these words and aspects, construct a domain-specific sentiment lexicon, and use the lexicon to classify sentiments of segments. We have the following definitions.

Definition (segment). A segment is a sequence of words that represent at most one sentiment. It can consist of multiple consecutive clauses, up to a whole sentence. It can also be shorter than a clause.

Definition (dependency relation). A dependency relation defines a binary relation that describes whether a pairwise syntactic relation between two words holds in a sentence. In ReNew, we exploit the Stanford typed dependency representations [de Marneffe et al., 2006] that use triples to formalize dependency relations. Each triple $Rel(w_{gov}, w_{dep})$ consists of a name of the relation Rel, a governor word w_{gov} and a dependent word w_{dep}.

Definition (domain-specific sentiment lexicon). A domain-specific sentiment lexicon consists of three lists of dependency triples. Each list is associated with one sentiment including positive, neutral, or negative.

Definition (sentiment analysis). Assume a set of reviews $D = \{r_1, r_2, \dots\}$. Each review r_i consists of a set of segments. Each segment s is associated with a sentiment label $l(s) \in \{\text{Positive}, \text{Neutral}, \text{Negative}\}$. The goal of ReNew is to classify sentiment labels for segments along with automatically generating a domain-specific sentiment lexicon.

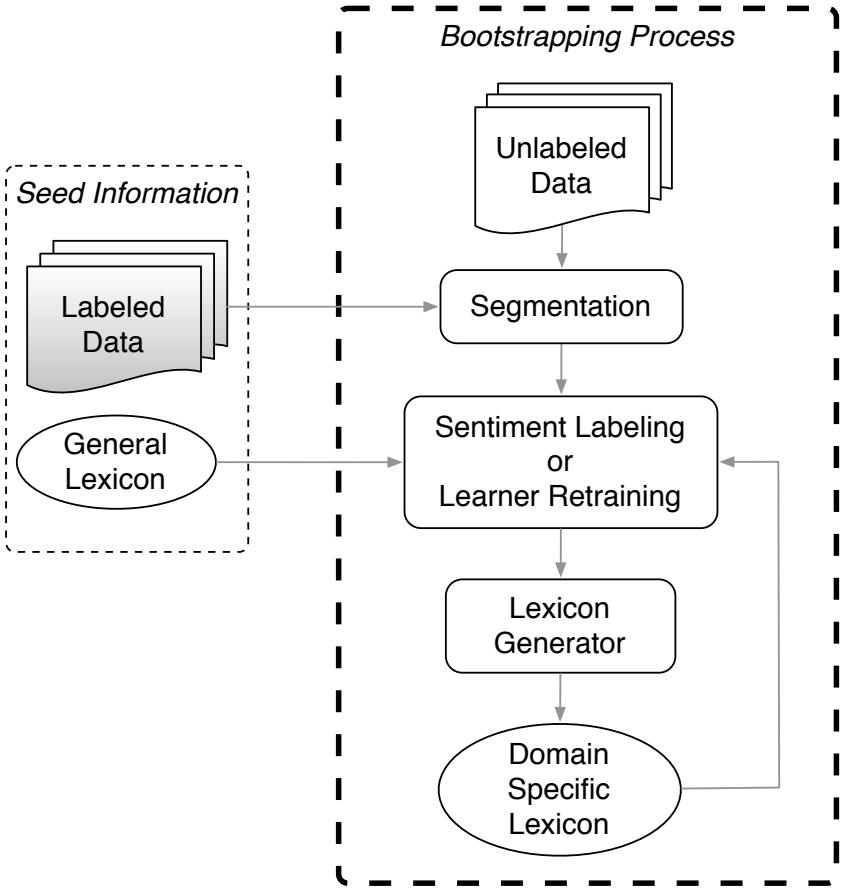


Figure 2.2: The ReNew framework schematically.

2.3 Overview of Proposed Framework

Figure 2.2 illustrates ReNew. Its inputs include a general sentiment lexicon and a small labeled training dataset.

Before starting the bootstrapping process, ReNew uses the general sentiment lexicon and the training dataset as prior knowledge to build the initial learners. On each iteration in the bootstrapping process, a set of unlabeled data is first split into segments using our rule-based segmentation algorithm. Second, the learners predict labels of segments based on current knowledge. Third, the lexicon generator determines which newly learned dependency relation triples to promote to the lexicon. At the end of each iteration, the learners are retrained via the updated lexicon so as to classify better on the next iteration. After labeling all of the data, we obtain the final version of our learners along with a domain-specific lexicon.

One can view this process as an approximation to the Expectation Maximization (EM) algorithm in which the E step involves iteratively estimating the truth labels for segments, and

the M step involves retraining our sentiment learners.

2.4 Segmentation

As we see in the example described in Section 2.1, the changes of sentiments typically happen when authors describe new aspects or mention transition cues. The changes can occur between sentences, clauses, or even within a clause. To capture them, ReNew use segments as the basic units for sentiment analysis using a rule-based segmentation algorithm introduced below.

2.4.1 Rule-Based Segmentation Algorithm

Algorithm 1 Rule-based segmentation.

```
Require: Review dataset T
 1: for all review r in T do
 2:    Remove HTML tags
 3:    Expand typical abbreviations
 4:    Mark special name-entities
 5:    for all sentence m in r do
 6:       while m contains a transition cue and m is not empty do
 7:          Extract subclause p that contains the transition cue
 8:          Add p as segment s into segment list
 9:          Remove p from m
10:       end while
11:       Add the remaining part in m as segment s into segment list
12:    end for
13: end for
```

The algorithm starts with a review dataset T. Each review r from dataset T is first normalized by a set of hard-coded rules (lines 2–4). One goal of this normalization step is to remove all of the unnecessary punctuation. To achieve this goal, we remove all HTML tags and translate typical abbreviations into their fully understandable form using a set of pre-defined regular expressions. The other goal of the normalization step is to mark special name entities. For example, it replaces a webpage link by #LINK#, a monetary amount $78.99 by #MONEY#, and so on. These name entities may help classify the sentiment.

After the normalization step, it splits each review r into sentences, and each sentence into subclauses (lines 6–10) provided transition cues occur. In effect, the algorithm converts each review into a set of segments.

Note that ReNew focuses on capturing and utilizing the sentiment changes. Therefore, this segmentation algorithm considers only two specific types of transition cues including contradiction and emphasis. More general transition cues are considered in the classification model, such as the conclusion, consequence, or sequence.

2.5 Learner Retraining

At the end of each iteration, ReNew retrains each learner as shown in Figure 2.3. Newly labeled segments are first selected by a filter. Then given an updated lexicon, learners are retrained to perform better on the next iteration. Detailed description of the filter and learner are presented below.

2.5.1 Filter

Bootstrapping, as an iterative learning approach, can suffer if labeling errors accumulate. The function of the filter is to mitigate this issue. In ReNew, newly acquired training samples are segments with labels that are predicted by old learners. Each predicted label is associated with a confidence value. A filter is applied to select those labeled segments with confidence greater or equal to a preset threshold.

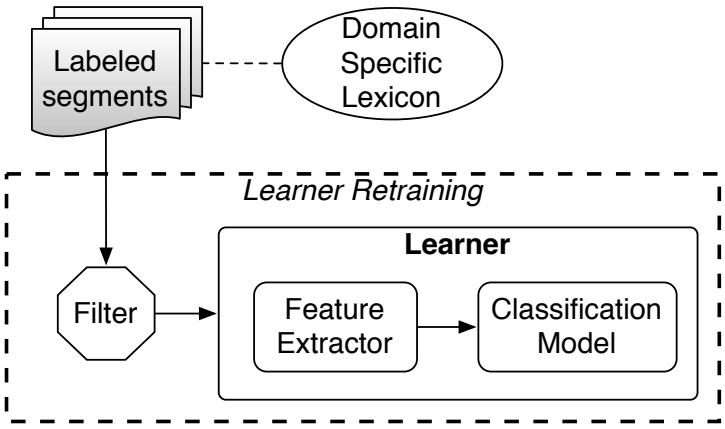


Figure 2.3: Retrain a relationship learner.

2.5.2 Learner

ReNew uses learners to capture different types of relationships among segments to classify sentiments by leveraging these relationships. Each learner contains two components: a feature

Table 2.1: A list of transition types used in ReNew

Transition Types	**Examples**
Agreement, Addition, and Similarity	also, similarly, as well as, ...
Opposition, Limitation, and Contradiction	but, although, in contrast, ...
Cause, Condition, and Purpose	if, since, as/so long as, ...
Examples, Support, and Emphasis	including, especially, such as, ...
Effect, Consequence, and Result	therefore, thus, as a result ...
Conclusion, Summary, and Restatement	overall, all in all, to sum up, ...
Time, Chronology, and Sequence	until, eventually, as soon as, ...

extractor and a classification model. To train a learner, the feature extractor converts labeled segments into feature vectors for training a classification model. In ReNew, we implement two learners that capture the forward and backward relationship among the segments, respectively. We present their detailed description in Section 2.6.1.

Feature Extractor

The feature extractor generates five kinds of features as below.

Grammar: (For each segment) part-of-speech tag of every word, the type of phrases and clauses (if known). Grammar features are binary and extracted using the Stanford Parser [Klein and Manning, 2003].

Opinion word: To exploit the general sentiment lexicon, we use two features with a binary value indicating the presence or absence of a word in the positive or negative list in a general sentiment lexicon.

Dependency relation: The lexicon generated by ReNew uses the Stanford typed dependency representation as its structure. Previous studies show that the effectiveness of using dependency relations for sentiment analysis [Zhang et al., 2010; Qiu et al., 2011].

Transition cue: Transition cues are important as they are good indicators for tracking the changes of the sentiment. ReNew exploits seven types of transition cues as shown in Table 2.1.

Punctuation, special name-entity, and segment position: Some punctuation marks are reliable carriers of sentiment. For example, a sentence ending with an interrobang ?! often conveys disbelief or a sense of excitement about a question. Remember in our

segmentation algorithm, we mark some special name-entities that may help classify the sentiment. These special name-entities include time, money, webpage link, email address, and numbers used for listing (e.g., (1), 1, or 1.). We also use positions of segments in reviews as features including beginning, middle, and end.

Classification Model

ReNew employ Conditional Random Fields (CRFs) [Lafferty et al., 2001] as classification model to classify sentiments for segments. CRFs can be seen as a special case of log-linear models. We start with introducing the general log-linear model.

Log-linear Model Log-linear model has become a widely used tool for Nature Language Processing classification tasks [Tsuruoka et al., 2009]. Assume we have an observation $x \in X$ and a candidate label $y \in Y$ of x, where X and Y are sets of observations and labels, respectively. In a log-linear model, the conditional probability of y given the observation x is computed as

$$p(y|x;w) = \frac{1}{Z(x;w)} \exp \sum_{j}^{J} (\omega_j \cdot F_j(x,y))$$

$$Z(x;w) = \sum_{y'}^{Y} \exp \sum_{j}^{J} (\omega_j \cdot F_j(x,y'))$$

where $F_j(x,y)$ is a feature function that maps pair (x,y) to a binary or real value. ω is a weight describing the influence of F_j. If $\omega_j > 0$, a large positive value of this feature function makes that label y is more likely to be the predicted label, holding everything else fixed. Z is a normalizing factor that makes the exponentiation ranging between 0 and 1.

Conditional Random Fields (CRFs) Conditional Random Fields are a special case of log-linear model where we are modeling the conditional probability of a sequence of labels instead of an individual label. In particular, for our sentiment classification model, given a sequence of segments $\bar{x} = (x_1, \ldots, x_n)$ and a sequence of sentiment labels $\bar{y} = (y_1, \ldots, y_n)$, the CRFs model $p(\bar{y}|\bar{x})$.

To limit the complexity of the required computation, we adapt the linear-chain CRFs model, which relies on the same first-order Markov chain representation as the HMM, as shown in Figure 2.4. The edges between any two labels and the edges from these two labels to the observation form cliques, such as the bold edges in Figure 2.4. Based on the Hammersley and Clifford theorem [Besag, 1974], in CRFs, the conditional probability can be computed as

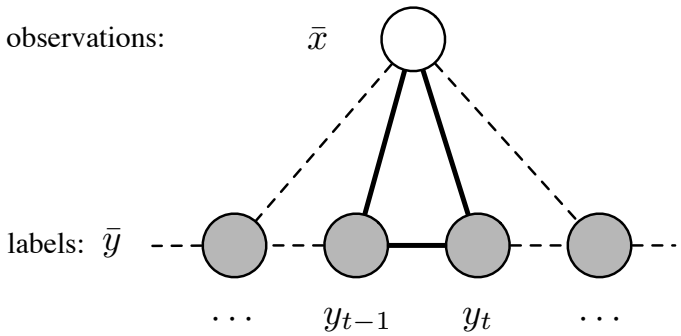


Figure 2.4: Conditional Random Fields.

$$p(\bar{y}|\bar{x}) = \frac{1}{Z(\bar{x})} \exp \sum_{j}^{J} (\omega_j \cdot F_j(\bar{x}, \bar{y}))$$

$$F_j(\bar{x}, \bar{y}) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, \bar{x}, i)$$

where ω is a set of weights learned in the training process to maximize $p(\bar{y}|\bar{x})$. $Z(\bar{x})$ is a normalization constant that is the sum of all possible label sequences. And, F_j is a feature function that is a sum of f_j over $i \in (1, n)$, where n is the length of $\bar{y}$. f_j can have arbitrary dependencies on the observation sequence $\bar{x}$ and neighboring labels. For example, to capture the forward contradiction relationship between two segments, we can define the following feature function:

$$f_{j,\text{contradiction}} = \{y_{i-1} = \text{POS},\ y_i = \text{NEG},\ x_i \text{ contains but|however}\}$$

This feature captures the case where current segment contains word *but* or *however* with the label *POS* and its previous segment is *NEG*.

Why Choose CRFs? Naïve Bayes (NB) is a simple approach that assumes conditional independence between features. Logistic regression (LR) relaxes the independence assumption among features by modeling the conditional distribution and leveraging the coefficients to learn correlations among features. Generally, LR performs better than NB for many classification tasks [Caruana and Niculescu-Mizil, 2006]. However, LR cannot capture relationships among segments for sentiment classification.

Sequential classification models are well-suited to our task since they model the statistical

dependencies among entities across the time. Hidden Markov Model (HMM) [Rabiner, 1990] is a classic sequential model. It uses a linear chain to connect a sequence of observations generated by a sequence of hidden states. Training an HMM model involves finding the optimal transition (between states) and emission (between states and observations) matrices that maximize the joint probability of the observations. However, due to its generative structure, HMM cannot model overlapping and dependent features extracted from observations. For example, in our task, an HMM can capture the probability of a positive segment that contains word *slow* followed by a negative segment that starts with transition cue *but*. In a word, we cannot use feature combinations for classification in HMM.

Conditional Random Fields (CRFs) [Lafferty et al., 2001] avoid the above limitations of HMM. Features in CRFs can be overlapping and involve any part of observations. Also by modeling the conditional probability $p(\bar{y}|\bar{x})$ directly, we can remain agnostic about $p(\bar{x})$. This is especially useful for NLP tasks, where the diversity of linguistic representations make it impossible to accurately model observations.

2.6 Sentiment Labeling

ReNew starts with a small amount of labeled training set. Knowledge from this initial training set is not sufficient to build an accurate sentiment classification model or generate a domain-specific sentiment lexicon. Unlabeled data contains rich knowledge, and it can be easily obtained. To exploit this resource, on each iteration, the sentiment labeling component, as shown in Figure 2.5, labels the data by using multiple learners and a label integrator. ReNew uses two learners to learn the relationship among segments.

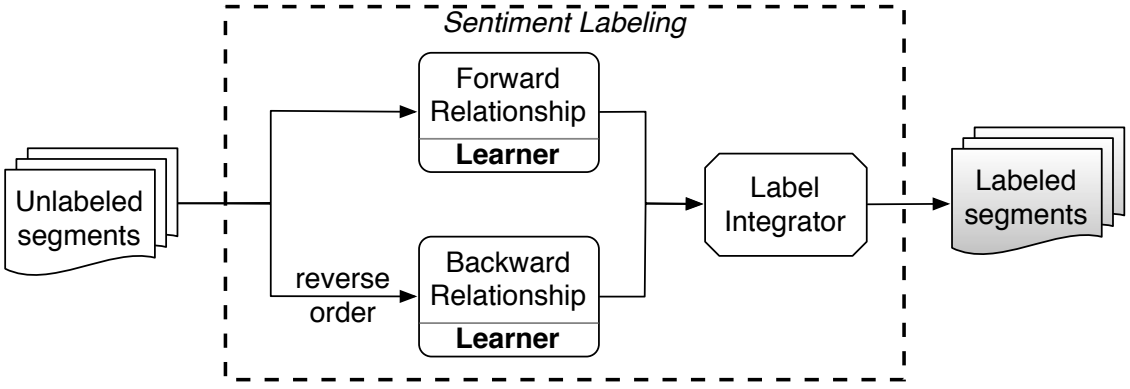


Figure 2.5: Sentiment labeling.

2.6.1 Forward and Backward Relationship Learner

The Forward Relationship (FR) Learner seeks to learn the relationship between the current segment and the next one. More specifically, it tries to answer the question that given the sentiment label and content of a segment, what will be the best possible sentiment label of its next segment.

The FR Learner is applicable to the following situation: assume two segments are connected by a transition word. But existing knowledge is not sufficient to infer the sentiment of the latter segment. For instance, consider the following sentence [TripAdvisor, 2007].

(1) *The location is great,* (2) ***but*** *the staff was pretty ho-hum about everything from checking in, to AM hot coffee, to PM bar.*

The sentence has been split into two segments by the segmentation algorithm. We can easily infer the sentiment polarity of Segment 1 based on the word *great* which is commonly included in many general sentiment lexicons. For Segment 2, without any context information it is difficult to infer its sentiment polarity. Although word *ho-hum* indicates a negative polarity, it is not a frequent word, and is not included in any general sentiment lexicons. Conjunction *but* clearly signals an opposite relationship between the two segments, however. So, given the fact that the formal segment is positive, a pre-trained FR learner can easily classify the sentiment of the latter segment.

The Backward Relationship (BR) learner is similar to the FR learner. The major difference is that the BR learner tries to learn and utilize the relationship between the current segment and the previous one. Therefore, to train or use the BR learner, ReNew first sorts the segments in each review in reverse order.

2.6.2 Label Integrator

Given the candidate sentiment labels suggested by multiple learners, the label integrator selects the label with the highest confidence. Segments are left unlabeled if their candidate sentiment labels belong to mutually exclusive categories with the same confidences.

The label integrator promotes labeled segments to knowledge discovery using a hard-coded and intuitive strategy. Labeled segments that have a confidence higher than pre-set threshold are promoted.

2.7 Lexicon Generator

In each iteration, after labeling the segments, the lexicon generator identifies new triples automatically. This module contains two parts: a Triple Extractor and a Lexicon Integrator. For each sentiment, the Triple Extractor (TE) extracts candidate dependency relation triples using

a novel rule-based approach. The Lexicon Integrator (LI) evaluates the proposed candidates and promotes the most supported candidates to the corresponding sentiment category in the domain-specific lexicon. Figure 2.6 depicts the process described above.

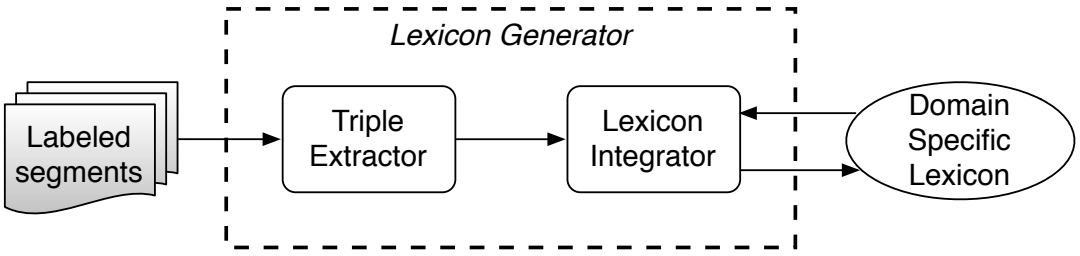


Figure 2.6: Knowledge discovery component.

2.7.1 Dependency Relation

We start with introducing the dependency relation used heavily by ReNew's knowledge discovery module. In essence, it describes grammatical relations between words in a sentence. For example, in the sentence *This is a great hotel*, one of the dependency relations is between *hotel* and *great* in which *great* is the modifier modifies *hotel*. In ReNew, we exploit the Stanford typed dependency representations [de Marneffe et al., 2006] to formalize the dependency relations.

Any sentence can be represented by a set of dependency relations. ReNew uses five basic and two collapsed dependency relations, which commonly contain the sentiment information in segments. We give their brief introduction below. We will consider additional dependency relations in the future.

- ***amod*: adjectival modifier**
  In the ***amod*** relation, the governor is a Noun Phrase (NP) and the dependent is any adjectival phrase that is used to modify the meaning of the NP.
  e.g.,
  emphGreat hotel,friendly helpful staff.
  ↪ *amod* (hotel, Great)
  ↪ *amod* (staff, friendly)
  ↪ *amod* (staff, helpful)

- ***acomp*: adjectival complement**
  In the ***acomp*** relation, the governor is a verb and the dependent is an adjectival phrase that serves as the complement (like an object of the verb).

e.g., *Pool looked nice especially at night.*
↪ *acomp* (looked, nice)

- ***nsubj*: nominal subject**
  In the ***nsubj*** relation, the dependent is a noun phrase that is the syntactic subject of a clause. Generally, the governor is the verb related to the dependent. But the governor can be an adjective or noun if the verb is the copular verb.
  e.g., *The hotel and staff were perfect.*
  ↪ *nsubj* (perfect, hotel)
  ↪ *nsubj* (perfect, staff)

- ***neg*: negation modifier**
  In the ***neg*** relation, the dependent is a negation word and the governor is the word it modifies.
  e.g., *but the location is not great.*
  ↪ *neg* (great, not)

- ***root*: root**
  In the ***root*** relation, the dependent is the root of the sentence and the governor is a fake node *ROOT*.
  e.g., *but the location is not great.*
  ↪ *root* (ROOT, great)

- ***conj_and*: conjunct with a word that is similar to *and***
  In the ***conj_and*** relation, the governor and its dependent are the two words that are connected by coordinating conjunctions, such as *and*, *as well as*, and so on.
  e.g., *The place is eclectic and quirky.*
  ↪ *conj_and* (eclectic, quirky)

- ***prep_with*: prepositional modifier *with***
  In the ***prep_with*** relation, the governor and its dependent are the two words that are connected by the preposition *with*
  e.g., *The room was spacious with a balcony that overlooked 53rd street.*
  ↪ *prep_with* (spacious, balcony)

2.7.2 Triple Extractor

In ReNew, we use the Stanford typed dependency triples as the basic unit in the domain-specific lexicon. Table 2.2 lists the definition of the seven types of triples used in ReNew.

Table 2.2: Domain-specific lexicon triple types.

Triple Types	**Explaination**
$amod(w_{gov}, w_{dep})$	adjectival modifier
$acomp(w_{gov}, w_{dep})$	adjectival complement
$nsubj(w_{gov}, w_{dep})$	nominal subject
$root\_amod(\text{ROOT}, w_{dep})$	adjectival modifier root node
$root\_acomp(\text{ROOT}, w_{dep})$	adjectival complement root node
$root\_nsubj(\text{ROOT}, w_{dep})$	nominal sub ject root node
$neg\_pattern(\text{NO}, w_{dep})$	*neg* pattern

In Table 2.2, *amod*, *acomp*, and *nsubj* use the same definitions as described in Section 2.7.1. And, *root_amod* captures the root node of a sentence when it also appears in the adjectival modifier triple, similarly for *root_acomp* and *root_nsubj*. We observe that the word of the root node is often related to the sentiment of a sentence and this is especially true when this word also appears in the adjectival modifier, adjectival complement, or negation modifier triple.

Zhang et al. [2010] first propose *no_pattern* that describes a word pair whose first word is *no* followed by a noun or noun phrase. They show that this pattern is a useful indicator for sentiment analysis. In our dataset, in addition to the usage of the word *no*, we also observe the frequent usage of the word *nothing* followed by an adjective. For example, users always express their negative feeling about a hotel using sentence such as *Nothing special*. Therefore, we create the *neg_pattern* to capture a larger range of possible word pairs. In ReNew, *neg_pattern* is *no* or *nothing* followed by a noun or noun phrase or an adjective.

Triple Extractor uses a novel approach for extracting sentiment triples. For segments that contain only one clause, the extracting process follows the steps below.

1. Given a segment contains one clause, generate its dependency parse tree.
2. Remove all triples except those listed in Section 2.7.1.
3. Apply the rules in Table 2.3 to add or modify triples.
4. Add all seven types of triples to the list corresponding to the label of this sentence.

As an example, Figure 2.7 illustrates how ReNew extracts triples from the segment below:

The staff was slow and definitely not very friendly.

The extracted triple set is {$root\_nsubj(\text{slow}, \text{staff})$, $nsubj(\text{slow}, \text{staff})$, $nsubj(\text{not\_friendly}, \text{staff})$}.

Table 2.3: Rules for extracting sentiment triples.

Rule	**Function**	**Condition**	**Result**
$Rule_1$	Handle Negation	word w_i;	$w_i = w_{dep} + \text{“\_”} + w_i$
		$neg(w_{gov}, w_{dep})$;	
		$w_i = w_{gov}$;	
$Rule_2$	Build Relationships	word w_i and w_j;	$amod(w_{gov}, w_i)$
	($conj\_and$ and $amod$)	$conj\_and(w_i, w_j)$;	$amod(w_{gov}, w_j)$
		$amod(w_{gov}, w_i)$;	
$Rule_3$	Build Relationships	word w_i and w_j;	$acomp(w_{gov}, w_i)$
	($conj\_and$ and $acomp$)	$conj\_and(w_i, w_j)$;	$acomp(w_{gov}, w_j)$
		$acomp(w_{gov}, w_i)$;	
$Rule_4$	Build Relationships	word w_i and w_j;	$nsubj(w_i, w_{dep})$
	($conj\_and$ and $nsubj$)	$conj\_and(w_i, w_j)$;	$nsubj(w_j, w_{dep})$
		$nsubj(w_i, w_{dep})$;	

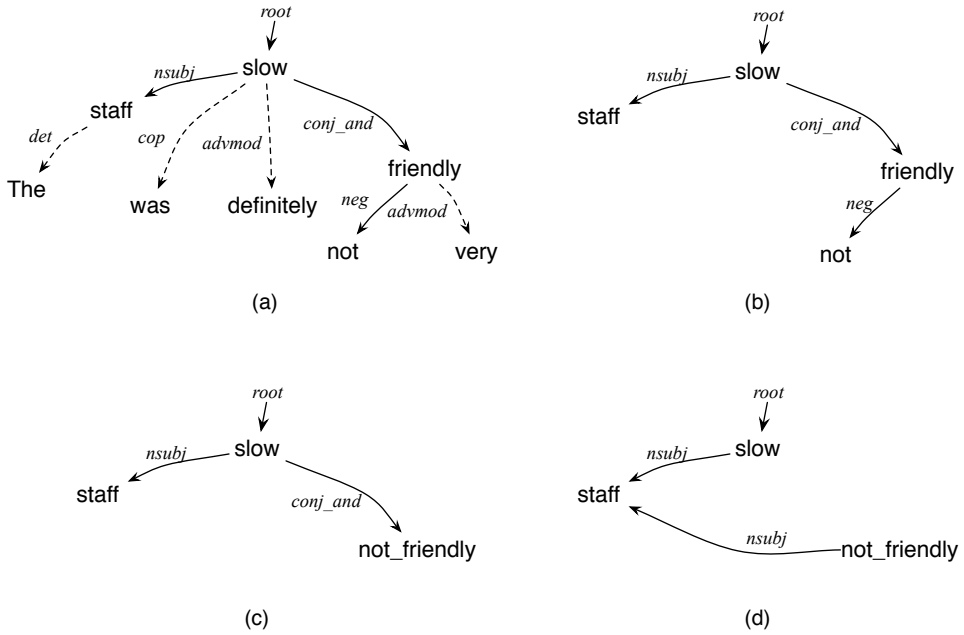


Figure 2.7: How to extract sentiment triples from a segment that contains one clause. (a) The initial dependency parse tree. (b) Remove non-sentiment triples. (c) Handle negation triples. (d) Build relationships.

For segments that contain multiple clauses, the KE first applies the Stanford parser to split these segments into clauses. Then for each clause, the KE extracts triples by following the steps described above.

Figure 2.8 illustrates how ReNew extracts triples from the segment below:

> *The rooms were big enough with 2 very comfy double beds, large TV, air conditioning and heater, decent sized bathroom and a computer with Internet access.*

The above segment contains multiple clauses. To make the figure easily readable, we use light-gray lines to show the dependency triples (*conj_and* and *prep_with*) spanning over the clauses. The extracted triple set is $\{root\_nsubj(\text{rooms}, \text{rooms}), nsubj(\text{rooms}, \text{big}), amod(\text{very}, \text{bed}), amod(\text{comfy}, \text{bed}), amod(\text{double}, \text{bed}), amod(\text{large}, \text{TV}), amod(\text{decent}, \text{bathroom}), amod(\text{sized}, \text{bathroom}), amod(\text{access}, \text{internet})\}$.

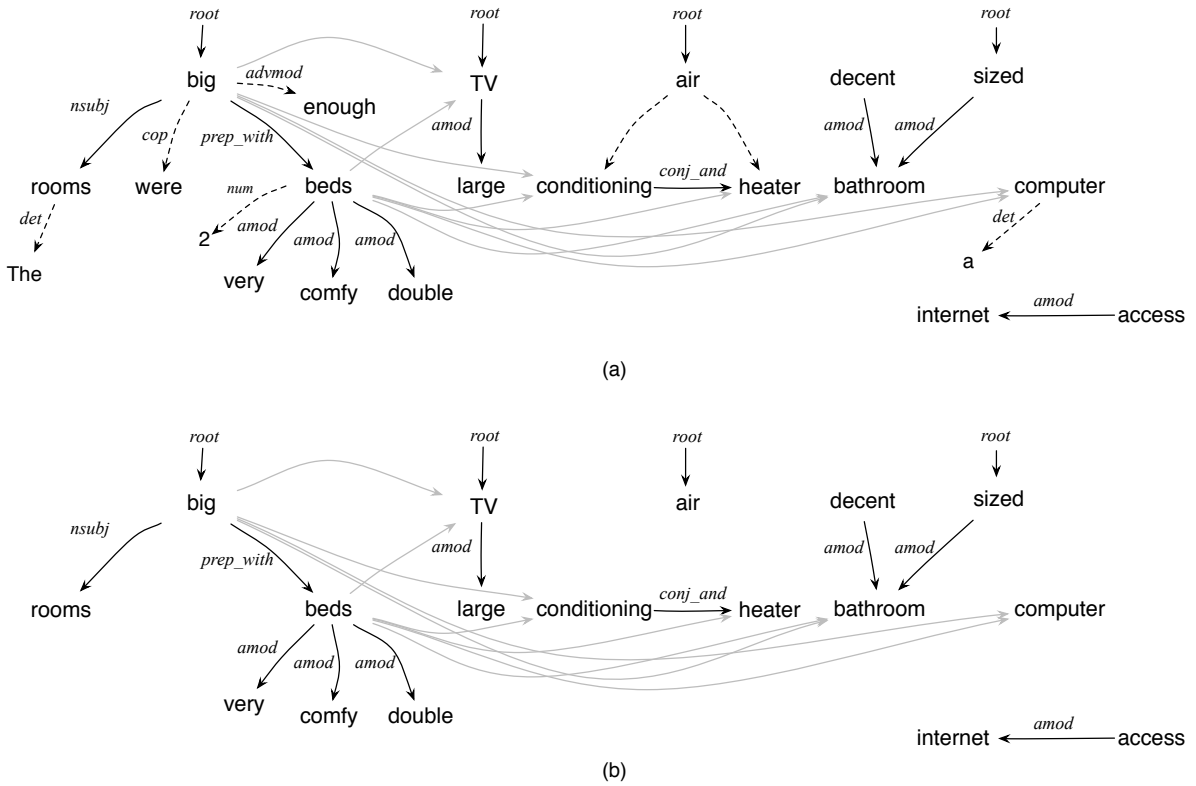


Figure 2.8: How to extract sentiment triples from a segment that contains multiple clauses. (a) The initial dependency parse trees for multiple clauses. (b) Remove non-sentiment triples, handle negation, and build relationships for each clause.

2.7.3 Lexicon Integrator

The Lexicon Integrator (LI) promotes candidate triples with a frequency greater than or equal to a preset threshold. The frequency list is updated in each iteration. The LI first examines the prior knowledge represented as an ordered list of the governors of all triples, each is attached with an ordered list of its dependents. Then, based on the triples promoted in this iteration, the order of the governors and their dependents is updated. Triples are not promoted if their governors or dependents appear in a predetermined list of stopwords.

The LI promotes triples by respecting mutual exclusion and the existing lexicon. In particular, it does not promote triples if they exist in multiple sentiment categories or if they already belong to a different sentiment category.

Finally, for each sentiment, we obtain seven sorted lists corresponding to *amod*, *acomp*, *nsubj*, *root_amod*, *root_acomp*, *root_nsubj*, and *neg_pattern*. These lists form the domain-specific sentiment lexicon.

2.8 Experiments

2.8.1 Datasets

To assess ReNew's effectiveness, we prepare two hotel review datasets crawled from TripAdvisor. One dataset contains a total of 4,017 unlabeled reviews regarding 802 hotels from seven US cities. The reviews are posted by 340 users, each of whom contributes at least ten reviews. The other dataset contains 200 reviews randomly selected from TripAdvisor. We collected ground-truth labels for this dataset by inviting six annotators in two groups of three. Each group labeled the same 100 reviews. We obtained the labels for each segment consist as positive, neutral, or negative. Fleiss' kappa scores for the two groups were 0.70 and 0.68, respectively, indicating substantial agreement between our annotators.

The results we present in the remainder of this section rely upon the following parameter values. The confidence thresholds used in the Label Integrator and filter are both set to 0.9 for positive labels and 0.7 for negative and neutral labels. The minimum frequency used in the Lexicon Integrator for selecting triples is set to 4.

To facilitate the annotation task, we developed a sentiment flow labeling toolkit, ReNewal, based on the Multi-Purpose Annotation Editor developed by Stubbs [2011]. A screenshot of ReNewal is shown in Figure 2.9. For facilitating online annotation task, we implement a web-based tool. A screenshot of the tool is shown in Figure 2.10.

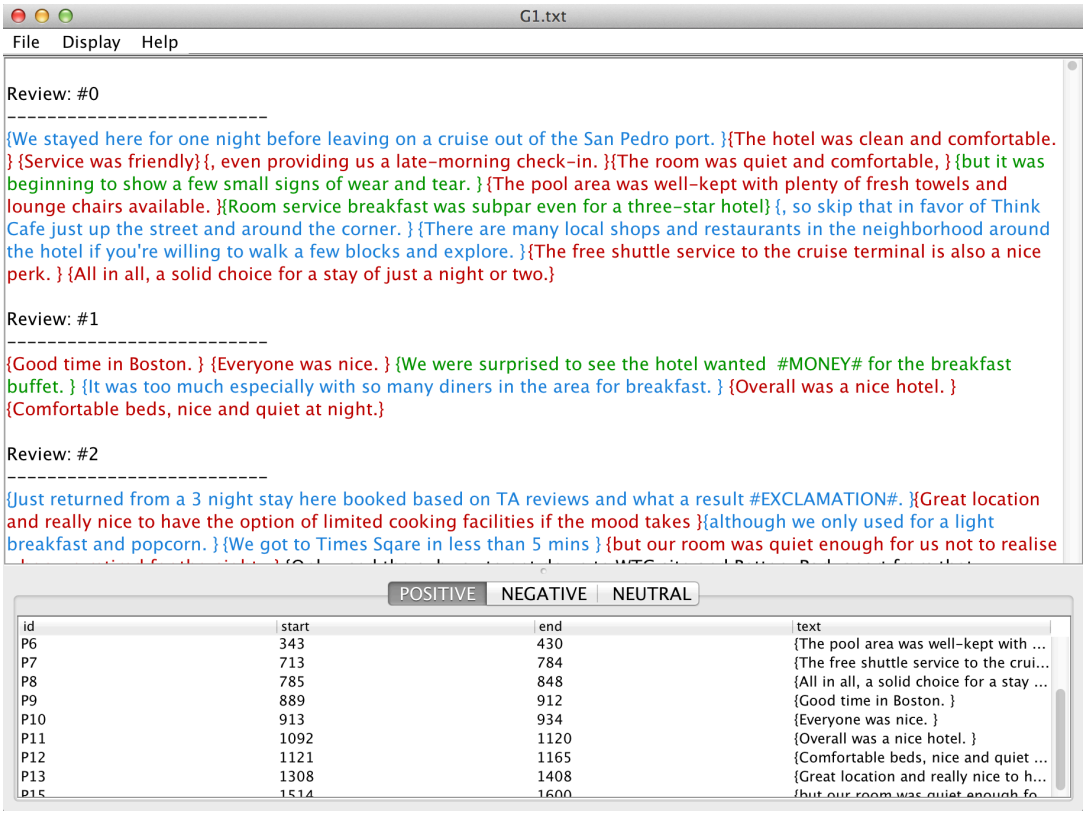


Figure 2.9: A screenshot of ReNewal.

2.8.2 Performance Measures

To assess the performance of our sentiment classification model, for a given testing set, we count the number of true positives (TP), true negatives (TN), false positives (FP) (negative samples but classified as positives), and false negatives (FN) (positive samples but classified as negatives). We use accuracy, macro-averaged and micro-averaged F-score to combine these numbers. Accuracy describes the proportion of segments that are correctly classified. F-score is the harmonic mean of precision and recall. As a task with multiple class labels, we compute macro-averaged and micro-averaged F-score. Macro averaging gives equal weight to each class, whereas micro averaging gives equal weight to each classification decision. These metrics are defined below.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

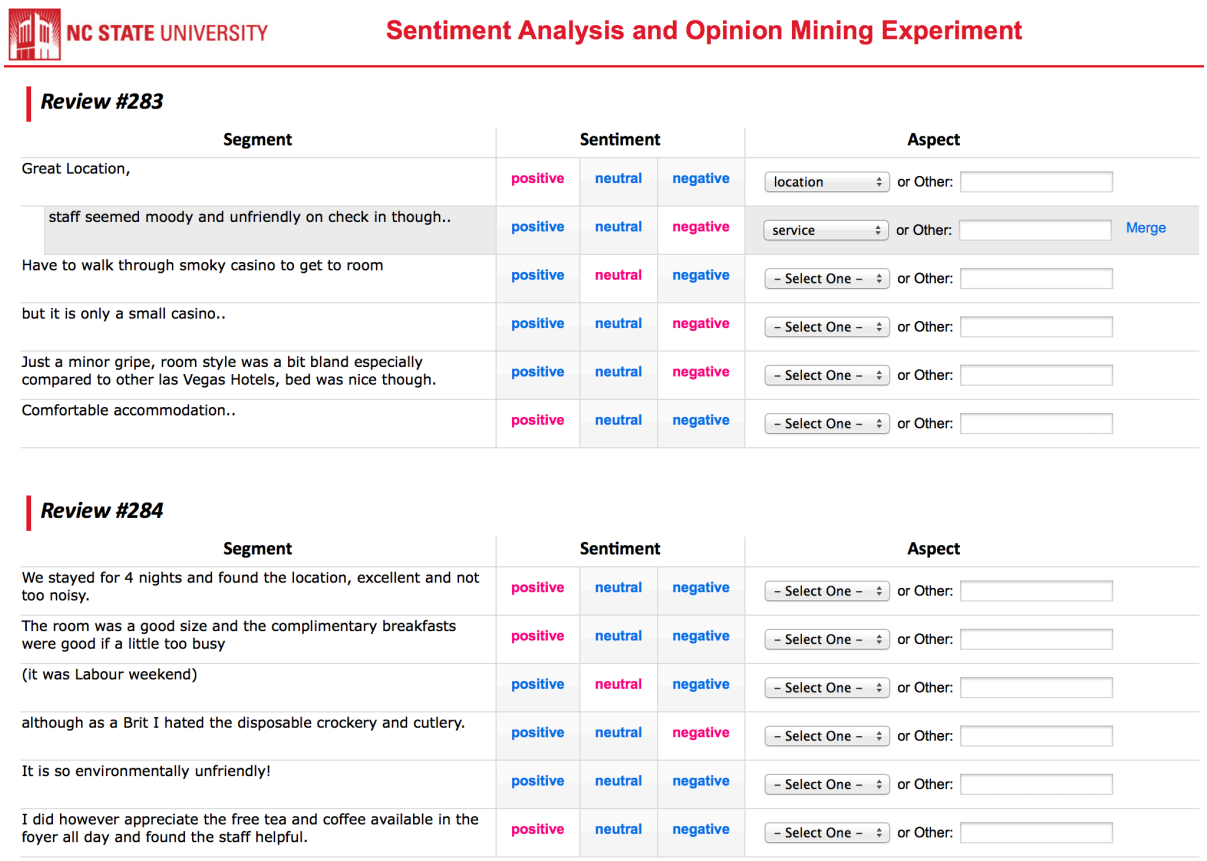


Figure 2.10: A screenshot of the web-based annotation tool.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{F-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{Macro F-score} = \frac{1}{n} \sum_{i=1}^{n} \text{F-score}_i$$

$$\text{Micro F-score} = \frac{2 \times \sum_{i=1}^{n} \text{precision}_i \times \sum_{i=1}^{n} \text{recall}_i}{\sum_{i=1}^{n} \text{precision}_i + \sum_{i=1}^{n} \text{recall}_i}$$

2.8.3 Feature Function Evaluation

Our first experiment evaluates the effects of different combinations of features. To do this, we first divide all features into four basic feature sets: T (transition cues), P (punctuation, special name-entities, and segment positions), G (grammar), and OD (opinion words and dependency

relations). We train 15 sentiment classification models using all basic features and their combinations. Figure 2.11 shows the results of a ten-fold cross validation on the 200-review dataset.

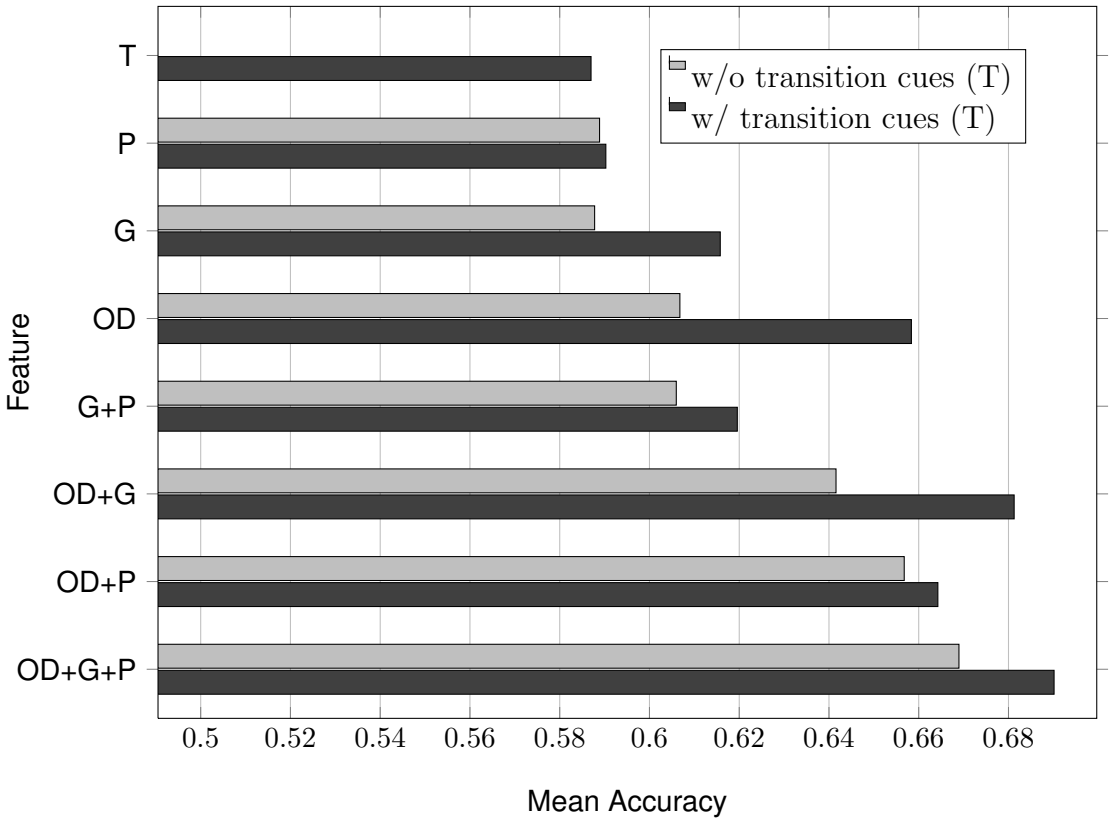


Figure 2.11: Mean accuracy using different features.

In the Figure, the light grey bars show the accuracy of the modeled trained without using transition cue features. Among the basic features, *OD* yields the best accuracy, followed by *G*, *P*, and *T*. Although *T* yields the worst accuracy, the incorporation of *T* with all of the other features improves the resulting accuracy, as shown by the dark grey bars. In particular, the accuracy of *OD* is remarkably improved by adding *T*. The model trained by using all of the feature sets yields the best accuracy.

2.8.4 Relationship Learners Evaluation

To evaluate the impact of the relationship learners and the label integrator, in our second experiment, we train and compare sentiment classification models using three configurations. The first configuration, named as *FW-L*, uses only the forward relationship learner. The second

configuration, named as *BW-L*, uses only the backward relationship learner. The last one, named as *ALL-L*, uses both forward and backward relationship learners, together with a label integrator. We evaluate them with ten-fold cross validation on a dataset consist of 200 randomly selected reviews. We report the accuracy, macro F-score, and micro F-score in Figure 2.12.

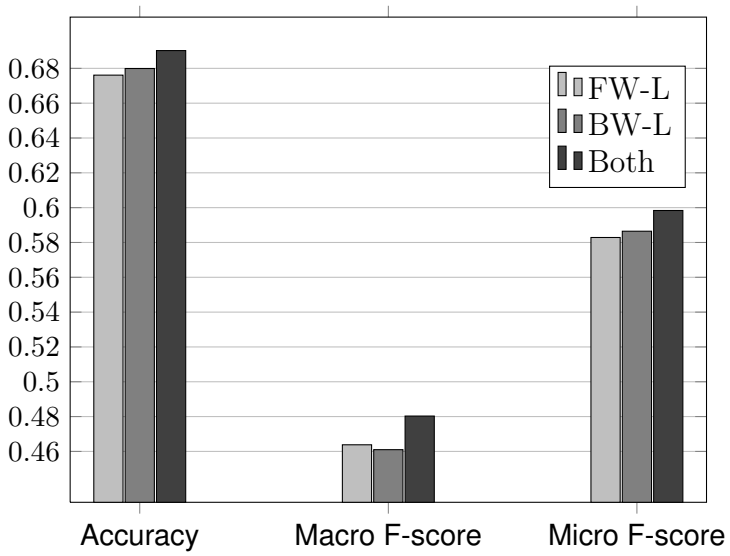


Figure 2.12: Comparison among the learners.

Figure 2.12 reports the accuracy, macro F-score, and micro F-score. It shows that BR learner produces better accuracy and a micro F-score than FR learner but a slightly worse macro F-score. Jointly considering both learners with the label integrator achieves better results than either alone. The results demonstrate the effectiveness of our sentiment labeling component.

2.8.5 Domain-specific Lexicon Assessment

In the third experiment, we evaluate the quality of the domain-specific lexicon automatically generated by ReNew. To do this, we first transform all of the 200 labeled reviews into feature vectors. Then we retrain Logistic Regression models using WEKA [Hall et al., 2009]. Note that we only use the features extracted from the lexicons themselves. This is important because to compare only the lexicons' impact on sentiment classification, we need avoid the effect of other factors, such as syntax, transition cues, and so on. We evaluate the models trained by using our domain-specific lexicon and two general sentiment lexicons including Affective Norms for English Words (ANEW) [Bradley and Lang, 1999] and Linguistic Inquiry and Word Count (LIWC) [Tausczik and Pennebaker, 2010].

ANEW contains 1,034 words in the English language. Each word is associated with ratings in the three affective dimensions, namely *valence* (which ranges from pleasant to unpleasant), *arousal* (which ranges from calm to excited), *dominance* (ranging from in control to out of control). In our experiment, we calculate arousal and valence value for each segment and use them as features.

LIWC is a text analysis program that uses a lexicon to identify the frequency of words in psychological dimensions. The lexicon used in LIWC consists of 105 categories where two of them are directly related to the sentiment, namely, *positive emotion* and *negative emotion*. We use them to generate two binary-valued features where the binary value indicates the presence or absence of a word in the positive or negative list in LIWC.

Table 2.4 shows the results obtained by ten-fold cross validation. Each weighted average is computed according to the number of segments in each class. The table shows the significant advantages of the lexicon generated by ReNew. ANEW achieves the highest recall for the positive class, but the lowest recalls in the negative and neutral classes. Regarding the neutral class, both ANEW and LIWC achieve poor results. The weighted average measures indicate our lexicon has the highest overall quality.

Table 2.4: Comparison results of different lexicons.

	ANEW			**LIWC**			**ReNew**		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Positive	0.59	**0.994**	0.741	0.606	0.975	0.747	**0.623**	0.947	**0.752**
Negative	0.294	0.011	0.021	**0.584**	0.145	0.232	0.497	**0.202**	**0.288**
Neutral	0	0	0	0	0	0	**0.395**	**0.04**	**0.073**
Weighted average	0.41	0.587	0.44	0.481	0.605	0.489	**0.551**	**0.608**	**0.518**

Table 2.5 presents part of the lexicon discovered by ReNew. ReNew automatically discovers distinguishable aspects from the reviews that clearly apply to hotels, such as Location, Bed, Room, and so on. Additionally, ReNew extracts the sentiment words related to each aspect. For instance, on the positive side, it discovers *friendly*, *help*, *nice* for the Staff aspect. On the negative side, the lexicon connects *old*, *ridiculous*, *crazy*, *worn* with the Looks aspect. We notice that some positive words also occur on the negative side. This may be for two reasons. First, some sentences that contain positive words may convey a negative sentiment, such as *The staff should be more efficient*. Second, the bootstrapping process in ReNew may introduce some wrong words by mistakenly labeling the sentiment of the segments. These challenges suggest useful directions for the future work.

Table 2.5: Part of a customized lexicon learned by ReNew.

	amod-positive
Hotel	other nice great new mani good best larg favorit excel
Room	nice clean larg live standard separ comfort other great upgrad
#exclam	great amaz nice good excel awesom free huge better wonder
Bed	comfort doubl comfi great clean nice new super good rollawai
	acomp-positive
Looks	nice good great dirti old date clean pretti fine cool
Feel	comfort welcom safe modern free gener sad dirti good cramp
Work	great fine hard perfect med better reliabl excel high low
Smell	good fresh nice great better strang amaz wonder excel weird
	nsubj-positive
Bed	comfort comfi wa good #exclam clean great is nice amaz
Locat	great good perfect conveni excel beat fantast #colon best superb
Room	clean had nice comfort larg spaciou have quiet wa readi
Staff	friendli help nice great pleasant effici welcom accommod profession excel
	amod-negative
Room	new small upgrad live tini non-smok delux onli mani extra
Desk	front small other grouch welcom onsight onlin check-in oval
Area	common small nice other littl indoor flamingo crowd fremont larg
Servic	quick internet terribl overal turn-down poor person small call amen
	acomp-negative
Looks	nice old ridicul crazi worn dirti tire
Felt	safe awkward roomi indiffer worn-out bad vulner under-serv modern
Wa	avail safe neg ash fridg noth intens bad miss
Mean	old okai easi horribl tini
	nsubj-negative
Servic	slow better #colon spotti shine effici avail bad start
Bathroom	small ha compact wa need #colon larg equip had
Bed	firm uncomfort look small need nice wa low
Room	small wa readi face avail quiet larg smell

2.8.6 Lexicon Generation and Sentiment Classification

Our fourth experiment evaluates the robustness of ReNew's lexicon generation process as well as the performance of the sentiment classification models using these lexicons.

We build a small unlabeled dataset by selecting reviews from the large dataset. This dataset contains 4,017 reviews posted by 340 users. Each user contributes at least ten reviews in this dataset.

We first generate ten domain-specific lexicons by repeatedly following steps as described below.

- For the first iteration

  - Build a training dataset by randomly selecting 20 labeled reviews (about 220 segments)
  - Train the learners using the training dataset and Linguistic Inquiry and Word Count (LIWC) sentiment lexicon

- For each iteration thereafter
  - Label 400 unlabeled reviews that randomly selected from our dataset (4,071 reviews)
  - Discover knowledge
  - Retrain the learners

- After labeling all of the data
  - Output a domain-specific lexicon

To evaluate the benefit of using domain-specific sentiment lexicons, we train ten sentiment classification models using the ten lexicons and then compare them, pairwise, against models trained with the general sentiment lexicon LIWC. Each model consists of an FR learner, a BR learner, and a label integrator. Each pairwise comparison is evaluated on a testing dataset with ten-fold cross validation. Each testing dataset consists of 180 randomly selected reviews (about 1,800 segments). For each of the pairwise comparisons, we conduct a paired t-test to determine if the domain-specific sentiment lexicon can yield better results.

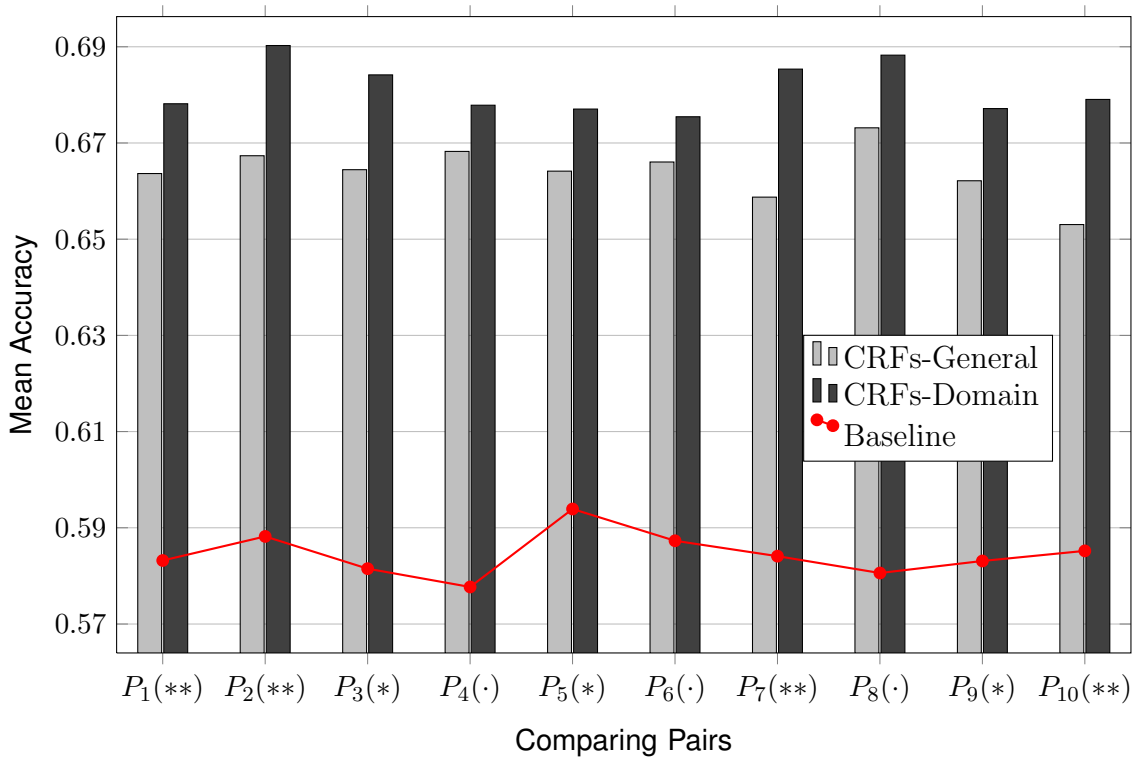


Figure 2.13: Mean accuracy with different lexicons.

Figure 2.13 shows the pairwise comparisons of accuracy between the two lexicons. Each group of bars represents the accuracy of two sentiment classification models trained using LIWC (CRFs-General) and the generated domain-specific lexicon (CRFs-Domain), respectively. The solid line corresponds to a baseline model that takes the majority classification strategy. Based on the distribution of the datasets, the majority class of all datasets is positive. We can see that models using either the general lexicon or the domain-specific lexicon achieve higher accuracy than the baseline model. Domain-specific lexicons produce significantly higher accuracy than general lexicons. In the figures below, we indicate significance to 10%, 5%, and 1% as '$\cdot$', '$*$', and '$**$', respectively.

Figure 2.14 and 2.15 show the pairwise comparisons of macro and micro F-score together with the results of the paired t-tests. We can see that the domain-specific lexicons (dark-grey bars) consistently yield better results than their corresponding general lexicons (light-grey bars).

As we described earlier, ReNew starts with LIWC and a labeled dataset and then generate ten lexicons and sentiment classification models by iteratively learning 4,017 unlabeled reviews without any human guidance. The above results show that the generated lexicons contain more domain-related information than the general sentiment lexicons. Also, note that the labeled

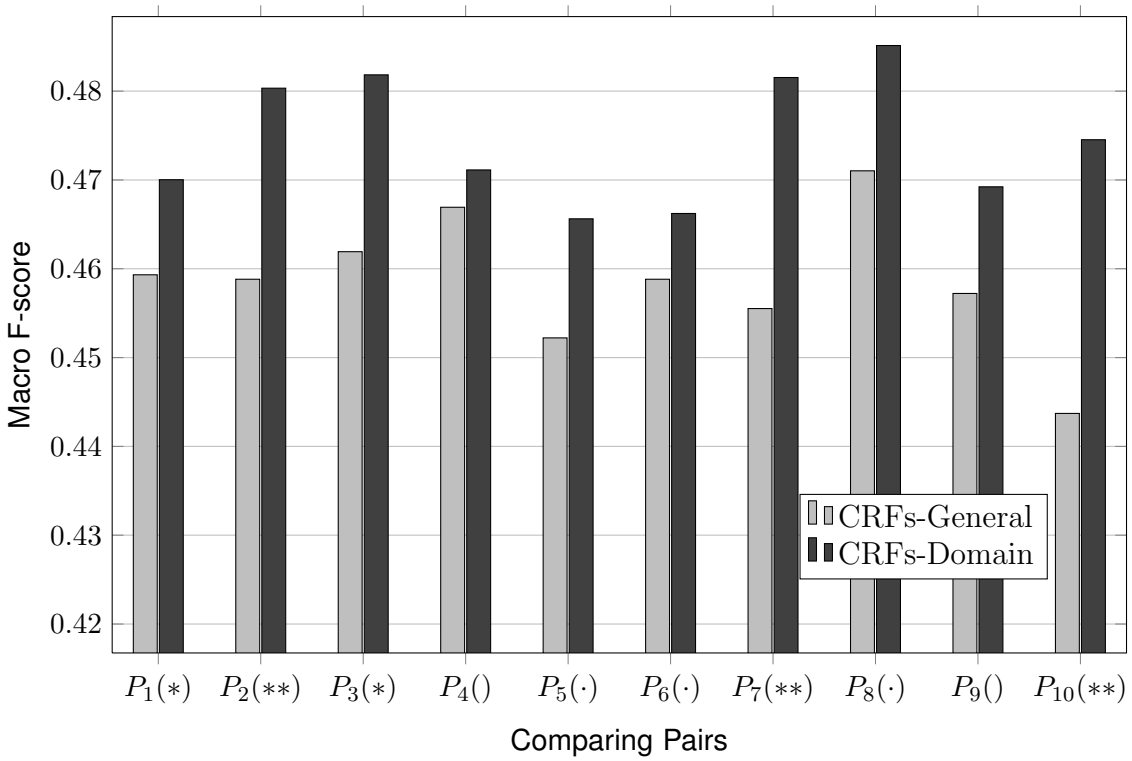


Figure 2.14: Macro F-score with different lexicons.

datasets used in our experiment contain only 20 labeled reviews. This is a particularly easy requirement to meet, which demonstrates that ReNew is suitable for practical use.

2.8.7 Comparison with Previous Work

Our fifth experiment compares ReNew with Lazaridou et al.'s [2013] approach for sentiment classification using discourse relations. Like ReNew, Lazaridou et al.'s approach works on the sub sentential level. However, it differs from ReNew in three aspects. First, the basic units of their model are elementary discourse units (EDUs) from Rhetorical Structure Theory (RST) [Mann and Thompson, 1988]. Second, their model considers the forward relationship between EDUs, whereas ReNew captures both forward and backward relationship between segments. Third, they use a generative model to capture the transition distributions over EDUs whereas ReNew uses a discriminative model to capture the transition sequences of segments.

EDUs are defined as minimal units of text and consider many more relations than the two types of transition cues underlying our segments. We posit that EDUs are too fine-grained for sentiment analysis. Consider the following sentence from Lazaridou et al.'s dataset with its EDUs identified.

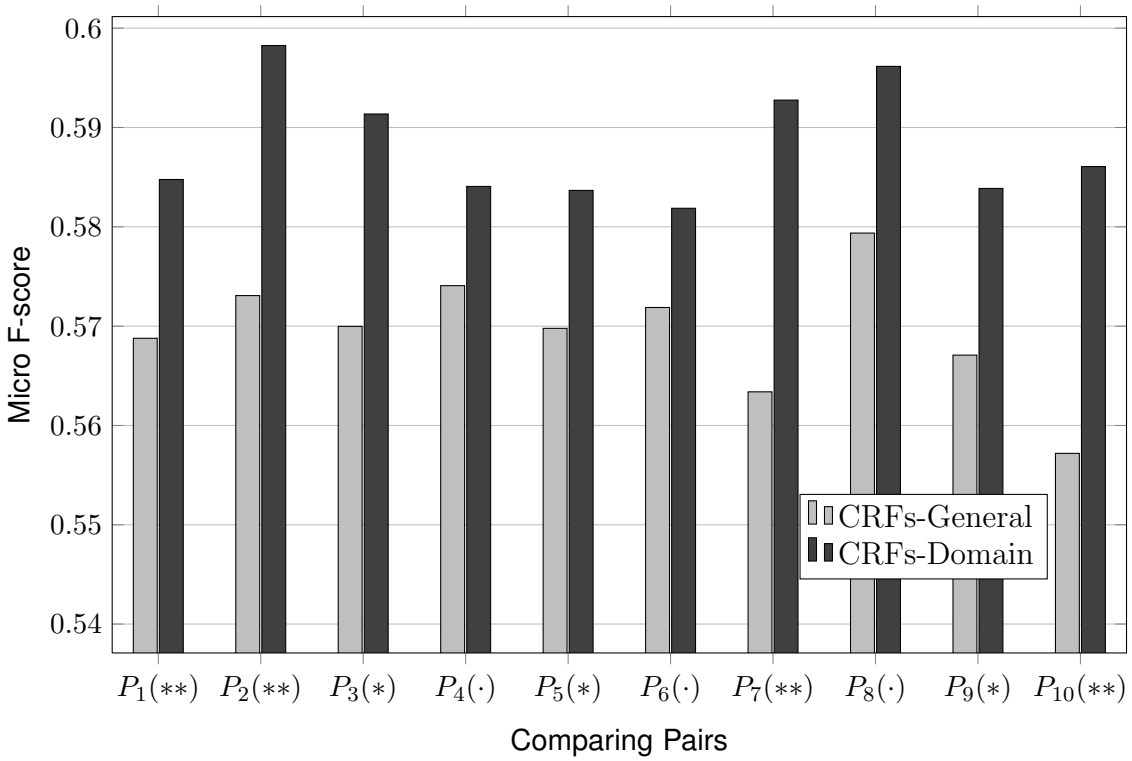


Figure 2.15: Micro F-score with different lexicons.

(1) *My husband called the front desk* (2) *to complain.*

Unfortunately, EDU 1 lacks sentiment and EDU 2 lacks the topic. Although Lazaridou et al.'s model can capture the forward relationship between any two consecutive EDUs, it cannot handle such cases because their model assumes that each EDU is associated with a topic and a sentiment. In contrast, ReNew finds just one segment in the above sentence.

Table 2.6: Comparison of our framework with previous work on sentiment classification.

Method	**Accuracy**
EDU-Model (Lazaridou et al.)	0.594
ReNew (our method)	**0.605**

Just to compare with Lazaridou et al., we apply our sentiment labeling component at the level of EDUs. Their labeled dataset contains 65 reviews, corresponding to 1,541 EDUs. Since

this dataset is also extracted from TripAdvisor, we use the domain-specific lexicon automatically learned by ReNew based on our 4,071 unlabeled reviews. Follow the same training and testing regimen (ten-fold cross validation), we compare ReNew with their model. As shown in Table 2.6, ReNew outperforms their approach on their dataset: Although ReNew is not optimized for EDUs, it achieves better accuracy. A paired t-test shows that the improvement is not statistically significant with p-value equals to 0.15.

2.9 Related Work

In this section, we briefly survey existing literature related to sentiment classification and sentiment lexicon generation.

2.9.1 Sentiment Classification

Sentiment classification, as one of the major tasks in sentiment analysis, has been extensively studied. Much of the earlier research [Wiebe et al., 1999; Bruce and Wiebe, 1999; Wiebe, 2000; Hatzivassiloglou and Wiebe, 2000] focuses on subjectivity classification in which texts are labeled as subjective or objective. More recent studies concentrate on developing automated approaches to classify sentiments expressed in texts as positive, neutral, or negative. Target texts can be documents, paragraphs, or sentences. Based on learning strategies, existing approaches fall into three categories: supervised, semi-supervised, and unsupervised.

Supervised Approach [Pang et al., 2002] is one of the earliest approaches applying supervised learning techniques to sentiment classification. They build three document-level sentiment classifiers using naïve Bayes, maximum entropy, and support vector machine (SVM), respectively. The classifiers use four types of features including unigram, bigram, parts of speech (POS), and position of words. Their empirical results on a movie review dataset suggest that more advanced techniques, such as discourse parsing and aspect identification, are needed for further improvements.

Pang and Lee [2004] exploit sentence subjectivity to improve document-level sentiment classification. They propose a graph-based subjectivity classification algorithm. Given a review, this algorithm first formulates it as a graph in which nodes are sentences in that review. Then it finds the best cut of this graph. This partition is the subjectivity classification of the review.

Kennedy and Inkpen [2006] explore the benefit of using valence shifters to classify sentiments of movie reviews. Ng et al. [2006] examine the role of four types of linguistic knowledge sources in sentiment classification. They conclude that bigrams, trigrams, and predefined term polarity information (e.g., a sentiment lexicon) are very useful sources for improve the classification models. McDonald et al. [2007] proposed a structured model to infer sentiments of

documents and sentences simultaneously. Their results show that jointly modeling sentiments of both levels improves classification accuracy. Nakagawa et al. [2010] introduce an approach using conditional random fields with hidden variables. Their approach uses dependency trees and achieves good accuracy on both Japanese and English sentiment datasets. Mejova and Srinivasan [2011] explore a number of feature selection strategies and examine the contribution of each feature. Rentoumi et al. [2012] introduce an approach using metaphorical expressions for sentiment classification. Their approach infers sentiments of words based on their sense in metaphorical expressions and use sentiments of words to infer sentiments of sentences. For deep learning approaches, Socher et al. [2013] introduce a model named Recursive Neural Tensor Network (RNTN). RNTN first represents phrases as word vectors and parse tree. Then it uses a tensor-based composition function to compute sentiment distribution vectors for higher nodes in the tree. Dos Santos et al. [2014] propose a convolutional neural network for sentiment analysis of short texts using character- to sentence-level information.

Semi-Supervised Approach Gamon et al. [2005] propose a semi-supervised sentiment analysis framework that jointly discovers topics and sentiments. They use a clustering algorithm to extract topics and train a naïve Bayes classifier using the expectation-maximization algorithm in a bootstrapping fashion. He's framework [2010] takes a predefined sentiment lexicon and an unlabeled dataset as its inputs and trains a classifier by using a predefined generalized expectation criteria. It is different from our framework, ReNew, in two ways. First, their framework does not focus on generating a domain-specific lexicon. Second, ReNew trains a classifier using relationships between sentences and clauses while their framework does not consider such relationships. Socher et al. [2011] introduce a deep learning approach for sentiment analysis. Their approach first train word vectors using an unsupervised neural language model [Bengio et al., 2003; Collobert and Weston, 2008]. Then it uses an unsupervised algorithm to learn tree structures of sentences. Finally, it trains autoencoders in a semi-supervised setting to predict sentiment distributions of sentences. Glorot et al. [2011] propose another deep learning approach based on stacked denoising autoencoders. They evaluate their approach for domain adaptation on a multi-domain dataset.

Unsupervised Approach Turney [2002] proposes an unsupervised approach for document-level sentiment classification. It first extracts phrases based on a set of predefined POS patterns. Then, it calculates semantic orientation based on the Pointwise Mutual Information (PMI) between the extracted phrases and two reference words *excellent* and *poor*. Finally, it classifies the sentiment of a review based on the average semantic orientation of its phrases. Hu and Liu [2004] investigate the possibility of using sentiment lexicons for unsupervised sentiment classification. They build a sentiment lexicon using a small set of sentiment words and WordNet

[Miller, 1995]. To predict the sentiment of a sentence, they use the dominant sentiment of words to determine the sentiment of the sentence. Similar ideas are also used in [Kim and Hovy, 2004; Ding et al., 2008].

Most of topic modeling-based sentiment analysis approaches [Jo and Oh, 2011; Lin et al., 2012; Kim et al., 2013] have the ability of unsupervised sentiment classification. These approaches learn distributions of aspects and sentiments based on word co-occurrence in documents. The learned distributions can be used for inferring sentiments of documents, sentences, or words. A major drawback of these approaches is that they infer sentiments without considering the changes of sentiments within or between sentences and clauses. Recently, Lazaridouet al. [2013] propose a topic modeling approach using discourse relations among EDUs. We have compared ReNew with their approach and discussed the results in Section 2.8.7.

2.9.2 Sentiment Lexicon Generation

Another important task in sentiment analysis is building sentiment lexicons. The goal of this task is to build a lexicon containing words typically used for conveying sentiments. Sentiments of these individual words provide basic key clues for revealing sentiments of longer texts, such as sentences or documents.

Researchers have proposed many approaches including manually approach and automated approach. The manually approach is labor intensive and time consuming. A well-known manually generated lexicon is Affective Norms for English Words (ANEW) [Bradley and Lang, 1999]. The lexicon contains a set of normative emotional ratings for 1,034 words in English. The ratings are labeled by a group of students with psychology major.

The automated approach commonly generates large sentiment lexicons by extending existing dictionaries or sentiment lexicons. Hu and Liu [2004] manually collect a small set of sentiment words and expand it iteratively by searching synonyms and antonyms in WordNet [Miller, 1995]. Similar approaches are used by Valitutti [2004] and Kim and Hovy [2004]. Rao and Ravichandran [2009] formalize the problem of sentiment detection as a semi-supervised label propagation problem in a graph. Each node represents a word, and a weighted edge between any two nodes indicates the strength of the relationship between them. Esuli and Sebastiani [2006] use a set of classifiers in a semi-supervised fashion to iteratively expand a manually defined lexicon. Their lexicon, named SentiWordNet, comprises the synset of each word obtained from WordNet. Each synset is associated with three sentiment scores: positive, negative, and objective. Qiu et al.[2011] use a bootstrapping framework that exploits a dependency parser to expand a small manually generated sentiment lexicon.

2.10 Conclusions

The leading lexical approaches to sentiment analysis from text are based on fixed lexicons that are painstakingly built by hand. There is little a priori justification that such lexicons would port across application domains. In contrast, our ReNew approach seeks to automate the building of domain-specific lexicons beginning from a general sentiment lexicon and the iterative application of linear-chain Conditional Random Fields. Our results are promising. ReNew greatly reduces the human effort by only using 20 labeled review and can generate a high-quality sentiment lexicon together with a classification model. In future work, we plan to apply ReNew to additional sentiment analysis problems such as review quality analysis and sentiment summarization.

Chapter 3

Arch: A Probabilistic Model of Author-Based Sentiment Aspect Discovery

3.1 Introduction

With the rapid development of social media, more and more people express their sentiments publicly, generating a large amount of opinionated text on a variety of topics. Opinionated text contains an author's compliments or criticisms of one or more entities, such as hotels and restaurants. Importantly, different authors may express sentiments with respect to different aspects of the same or similar entities. For example, when reviewing a hotel, a business traveler may consider aspects such as *Internet access*, *Concierge services*, and *Room*, whereas a tourist may consider aspects such as *Nearby area*, *Dining*, and *Room*. Even for the same aspect, authors may vary the emphasis they place on it.

Capturing the similarities and differences among authors can help in developing user-based applications. For example, we might produce recommendations for services that are better aligned with the expectations of a particular user. Such alignment may be inferred from other users who consider the same aspects important. Therefore, a linguistic model that can jointly discover sentiment and aspect at the granularity of authors would be valuable.

Natural Language Processing (NLP) has focused on developing probabilistic topic models [Hofmann, 1999; Blei et al., 2003]. Such models provide an unsupervised way to learn latent constructs from text, which enable NLP tasks that rely on human annotations. Although existing models [Rosen-Zvi et al., 2004; Kim et al., 2012; Diao and Jiang, 2013] can capture the association of texts with entities, such as authors or organizations, they focus only on topic discovery without investigating the sentiment expressed in texts. Leading sentiment analysis

approaches [Jo and Oh, 2011; Kim et al., 2013; Lin et al., 2012] can jointly model aspects and sentiments but fail to consider author information, which not only limits their applicability to user-based services but also hurts their performance. That is, a gap exists between the topic modeling literature on author-specific topic discovery (but ignoring sentiments) and the sentiment analysis literature of aspect-specific sentiments (but ignoring authors).

To bridge the gap, we propose Arch, an unsupervised probabilistic model for discovering author-based sentiments and aspects from opinionated text. Arch automatically generates interpretable author profiles that describe preferences in terms of sentiments and aspects. We evaluate Arch using four datasets from two domains. We find that Arch successfully discovers aspects associated with sentiments. Its author profiles are well correlated with ground truth. To exhibit the prospects for potential applications, we demonstrate the effectiveness of Arch for authorship attribution and sentiment classification.

The rest of chapter is organized as follows. Section 3.2 presents the graphical representation of Arch and illustrates its generative process. Section 3.3 compares Arch with two related probabilistic models. Section 3.4 describes an inference algorithm. Section 3.5 evaluates Arch over four datasets. Section 3.6 discuss the most relevant literature. Section 3.7 concludes with a discussion of the ramifications of Arch and outlines possible future directions.

3.2 Model

Arch seeks to extract author-based aspects and sentiments from opinionated text with minimal human effort. We adopt probabilistic topic models since they support unsupervised learning. To capture associations between authors and sentiment-aspect pairs, Arch generates a mixture over sentiments and aspects for each author. Arch assumes that reviews are mixtures of sentiments. To generate a segment for a review written by an author, we first select a sentiment. Then, given the sentiment and the author's profile represented as multinomial distributions over aspects, we select an aspect. Finally, we generate the words in the segment given the selected aspect and sentiment.

Before presenting the formal definition of the generative process model, let us discuss the representation for documents in Arch. Most existing topic models, including Latent Dirichlet Allocation (LDA) and JST, use the bag-of-words representation for documents. That is, they rely on word co-occurrences at the document level, which is problematic when applied to opinionated text [Titov and McDonald, 2008b; Wallach, 2006]. In reviews, words associated with the same aspect and sentiment are likely to appear in the same sentence or clause. To alleviate this problem, ASUM [Jo and Oh, 2011] extends JST and enforces a constraint that words appearing in a sentence belong to the same aspect and sentiment. As shown in Section 1.2, this is often not the case in opinionated text. Same as ReNew, Arch uses the segments as the unit

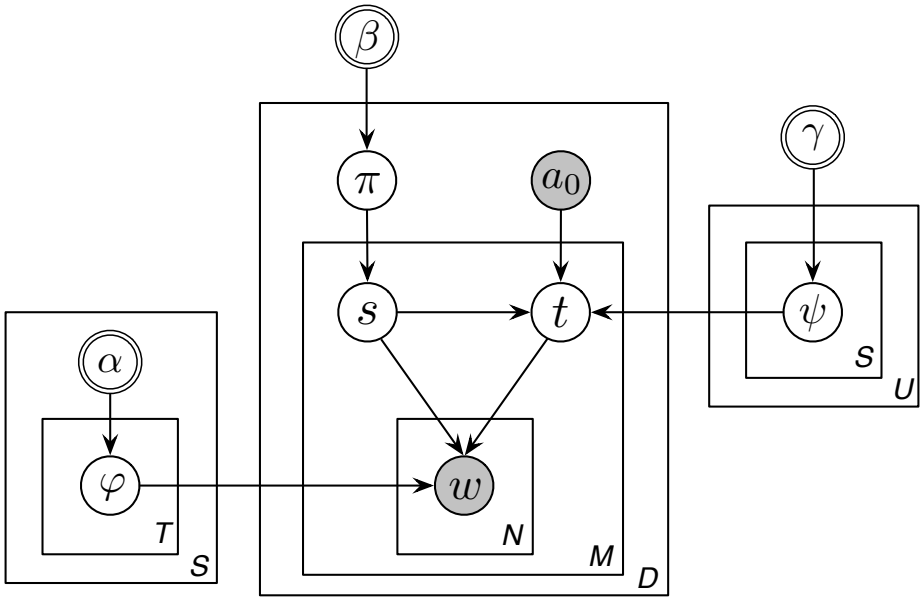


Figure 3.1: A graphical representation of Arch.

for a sentiment-aspect pair. We show experimentally that modifying ASUM to use segments improves its performance.

Given a set of D reviews written by a set of U authors with regards to T aspects and S sentiments, the generative process in Arch is as follows.

1. For each pair of aspect t and sentiment s, draw a word distribution $\varphi_{t,s} \sim Dir\ (\alpha)$.
2. For each author a_0 and each sentiment s, draw an aspect distribution $\psi_{s,a_0} \sim Dir\ (\gamma)$.
3. Given a review d written by author a_0, draw a sentiment distribution $\pi_d \sim Dir\ (\beta)$, and for each segment in d,
   - Choose a sentiment $s \sim Multi\ (\pi_d)$.
   - Given s, choose an aspect $t \sim Multi\ (\psi_{s,a_0})$.
   - Given t and s, sample words $w \sim Multi\ (\varphi_{t,s})$.

Figure 3.1 shows Arch's model graphically. Table 3.1 lists our notation.

3.3 Comparison with Previous Generative Models

Recently, however, there has been a growing interest in modeling texts using topic models. One of the earliest work is Probabilistic Latent Semantic Indexing (PLSI) proposed by Thomas

Table 3.1: Summary of notations and variables.

Notation	**Description**
D	number of reviews
M	number of segments in a review
N	number of words in a segment
T	number of aspects
S	number of sentiments
U	number of authors
φ	multinomial distribution over words
ψ	multinomial distribution over aspects
π	multinomial distribution over sentiments
α	Dirichlet prior for φ
β	Dirichlet prior for π
γ	Dirichlet prior for ψ
t	aspect assignment of a segment
s	sentiment assignment of a segment
w	word
a_0	the author of a review
$Dir\ (\cdot)$	Dirichlet distribution
$Multi\ (\cdot)$	multinomial distribution

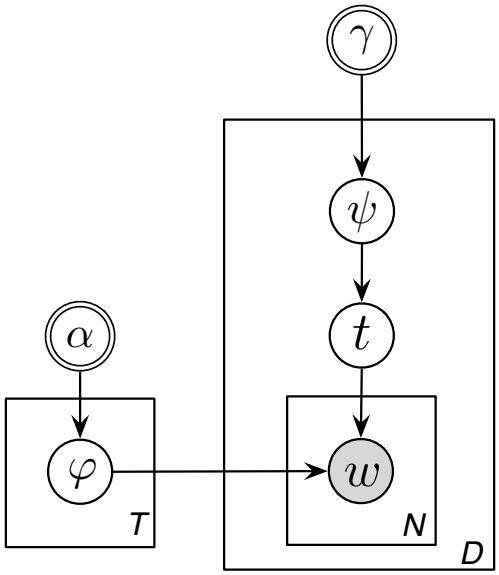


Figure 3.2: A graphical representation of LDA.

Hofmann in which he introduced a probabilistic framework that uses a latent variable to model the co-occurrence information between documents and words. Then, Blei et al.proposes LDA to address the shortcoming of PLSI for modeling unseen documents. In LDA, a document contains a multinomial distribution over a fixed number of topics, each of which is a multinomial distribution over words. Let φ denotes a multinomial distribution over words and ψ denotes a multinomial distribution over topics. The generative process of a set of D documents is as follows.

1. For each topic t, draw a word distribution $\varphi_t \sim Dir\ (\alpha)$.
2. For each document d, draw a topic distribution $\psi_d \sim Dir\ (\gamma)$.
3. For each word w in d,
   - Choose a topic $t \sim Multi\ (\psi_d)$.
   - Given t, sample a word $w \sim Multi\ (\varphi_t)$.

Figure 3.2 shows a graphical representation of the above process. LDA does not directly apply to our task since it does not include any parameter related to sentiments or authors. JST [Lin et al., 2012] and ASUM [Jo and Oh, 2011] extend LDA by modeling a review as two multinomial distributions, over topics and sentiments, respectively. They condition the probability of generating a word on both topic and sentiment. Let φ denotes a multinomial

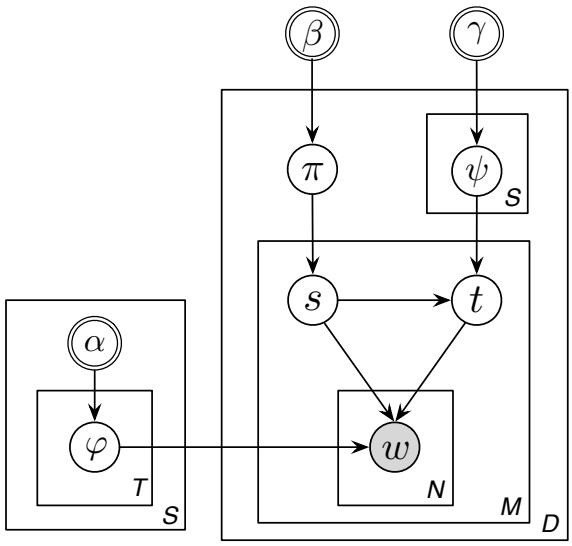


Figure 3.3: A graphical representation of ASUM.

distribution over words, ψ denotes a multinomial distribution over topics, and π denotes a multinomial distribution over sentiments. The generative process of a set of D documents is as follows.

1. For each pair of aspect t and sentiment s, draw a word distribution $\varphi_{t,s} \sim Dir\ (\alpha)$.
2. For each document d,
   - Draw a sentiment distribution $\pi_d \sim Dir\ (\beta)$.
   - For each sentiment, draw an aspect distribution $\psi_s \sim Dir\ (\gamma)$.
3. For each segment in d,
   - Choose a sentiment $s \sim Multi\ (\pi_d)$.
   - Given s, choose an aspect $t \sim Multi\ (\psi_s)$.
   - Given t and s, sample words $w \sim Multi\ (\varphi_{t,s})$.

Figure 3.3 represents a graphical representation of ASUM. JST is similar to ASUM with only one difference in basic sentiment expressing units. JST uses words while ASUM uses sentences. Neither JST nor ASUM captures associations between authors and reviews.

Rosen-Zvi et al.'s Author Topic model (AT) [2004] can capture the associations between authors and reviews. Figure 3.4 shows a graphical representation of AT. It adds an aspect distribution π for each author. When generating a word in a document, the probability of its

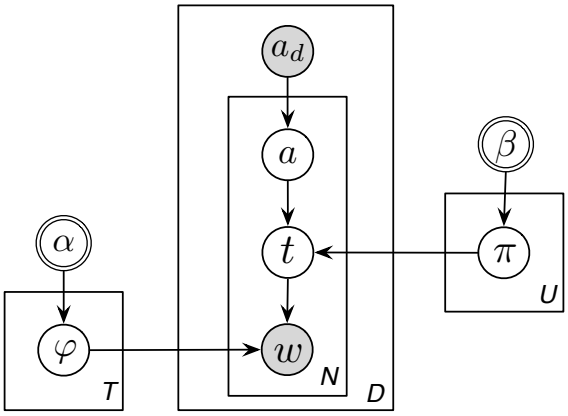


Figure 3.4: A graphical representation of AT.

aspect assignment is conditioned on the author of this document. Kim et al.'s Entity Topic model (ETM) [2012] assumes the probability of generating a word in a document is based on both topic and author. These models work well for detecting topics, but do not consider sentiment.

One may argue that we could address the problem of modeling author information by applying either JST or ASUM on reviews grouped by authors. In this way, we can obtain a model for each individual. However, this would face two challenges. First, we lose information about the co-occurrence information of aspects, sentiments, and words that exist across all authors, which would prevent comparing similarities and differences among authors. Second, training topic models require considerable amounts of data. In practice, it is almost impossible to obtain enough reviews for each individual authors.

3.4 Inference

Arch seeks to estimate $p(\boldsymbol{s}, \boldsymbol{t}|\boldsymbol{w}, a_0)$, the posterior of two latent variables, sentiments $\boldsymbol{s}$ and aspects $\boldsymbol{t}$, given all reviews written by author a_0. To this end, we adopt Liu's collapsed Gibbs sampling algorithm [1994]. To begin, we factorize the joint probability of the assignments of sentiments, aspects, and words for author a_0 as follows.

$$p(\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{w}|a_0, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = p(\boldsymbol{w}|\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{\alpha}) p(\boldsymbol{t}|\boldsymbol{s}, a_0, \boldsymbol{\gamma}) p(\boldsymbol{s}|\boldsymbol{\beta}) \tag{3.1}$$

By integrating out Φ in the first term in Equation 3.1, we obtain

$$
\begin{aligned}
p(\boldsymbol{w}|\boldsymbol{s},\boldsymbol{t},\boldsymbol{\alpha}) &= \int p(\boldsymbol{w}|\boldsymbol{s},\boldsymbol{t},\Phi)p(\Phi|\boldsymbol{\alpha})\,\mathrm{d}\Phi \\
&= \left(\frac{\Gamma(\sum_{w=1}^{W}\alpha_w)}{\prod_{w=1}^{W}\Gamma(\alpha_w)}\right)^{T*S} \times \prod_{t=1}^{T}\prod_{s=1}^{S}\frac{\prod_{w=1}^{W}\Gamma(n_{t,s}^{w}+\alpha_w)}{\Gamma\big[\sum_{w=1}^{W}(n_{t,s}^{w}+\alpha_w)\big]}
\end{aligned}
\tag{3.2}
$$

where $\Phi = \{\boldsymbol{\psi_i}\}_{i=1}^{T\times S}$; W is the size of the vocabulary; $n_{t,s}^{w}$ counts words w being assigned with sentiment s and aspect t; $n_{t,s}$ counts words being assigned with s and t in all reviews; and $\Gamma(\cdot)$ is the gamma function.

Next, by integrating out Ψ_{a_0} in the second term in Equation 3.1, we obtain

$$
\begin{aligned}
p(\boldsymbol{t}|\boldsymbol{s},\boldsymbol{\gamma},a_0) &= \int p(\boldsymbol{t}|\boldsymbol{s},\Psi_{a_0},a_0)p(\Psi_{a_0}|\boldsymbol{\gamma})\,\mathrm{d}\Psi_{a_0} \\
&= \left(\frac{\Gamma(\sum_{t=1}^{T}\gamma_t)}{\prod_{t=1}^{T}\Gamma(\gamma_t)}\right)^{S} \times \prod_{s=1}^{S}\frac{\prod_{t=1}^{T}\Gamma(n_{s,a_0}^{t}+\gamma_t)}{\Gamma\big[\sum_{t=1}^{T}(n_{s,a_0}^{t}+\gamma_t)\big]}
\end{aligned}
\tag{3.3}
$$

where $\Psi_{a_0} = \{\boldsymbol{\psi_i}\}_{i=1}^{S}$; n_{s,a_0}^{t} counts segments in author a_0's reviews being associated with sentiment s and aspect t; and n_{s,a_0} counts segments being associated with s in reviews written by a_0.

Similarly, for the third term in Equation 3.1, by integrating out Θ, we have

$$
\begin{aligned}
p(\boldsymbol{s}|\boldsymbol{\beta}) &= \int p(\boldsymbol{s}|\Theta)p(\Theta|\boldsymbol{\beta})\,\mathrm{d}\Theta \\
&= \left(\frac{\Gamma(\sum_{s=1}^{S}\beta_s)}{\prod_{s=1}^{S}\Gamma(\beta_s)}\right)^{D} \times \prod_{d=1}^{D}\frac{\prod_{s=1}^{S}\Gamma(n_{d}^{s}+\beta_s)}{\Gamma\big[\sum_{s=1}^{S}(n_{d}^{s}+\beta_s)\big]}
\end{aligned}
\tag{3.4}
$$

where $\Theta = \{\boldsymbol{\pi_i}\}_{i=1}^{D}$; D is the number of reviews; n_d^s is the number of times that a segment from review d being associated with sentiment s; and n_d counts segments in review d.

We can obtain the conditional probability used in Gibbs sampling by multiplying and canceling the terms in Equations 3.2–3.4. For author a_0, we obtain

$$
\begin{aligned}
p(s_i = s, t_i = t | \boldsymbol{s_{-i}}, \boldsymbol{t_{-i}}, \boldsymbol{w}, a_0) \\
&\propto \frac{n^t_{s,a_0,-i} + \gamma_t}{\sum_{t=1}^{T}(n^t_{s,a_0} + \gamma_t)} \times \frac{n^s_{d,-i} + \beta_s}{\sum_{s=1}^{S}(n^s_d + \beta_s)} \\
&\times \frac{\prod_{v_{i,j}} \prod_{c=0}^{C_{v_{i,j}}-1}(n^{v_{i,j}}_{t,s,-i} + \alpha_{v_{i,j}} + c)}{\prod_{c=0}^{C_i-1}(n_{t,s,-i} + \sum_{w=1}^{W} \alpha_w + c)}
\end{aligned}
\tag{3.5}
$$

where n^t_{s,a_0} counts segments from reviews written by author a_0 being associated with sentiment s and aspect t; n_{s,a_0} counts segments associated with sentiment s in reviews written by a_0; n^s_d counts segments from review d associated with sentiment s; n_d counts segments in d; $C_{v_{i,j}}$ counts words $v_{i,j}$ appearing in segment i; C_i counts words in segment i; $n^{v_{i,j}}_{t,s}$ counts words $v_{i,j}$ assigned with sentiment s and aspect t; $n_{t,s}$ counts words assigned sentiment s and aspect t in all reviews; and an index $-i$ means that we exclude segment i from the count.

The approximate probability of word w being associated with sentiment s and aspect t is

$$
\varphi_{s,t,w} = \frac{n^w_{s,t} + \alpha_w}{\sum_{w=1}^{W}(n^w_{s,t} + \alpha_w)} \tag{3.6}
$$

For author a_0, the approximate probability of a segment being associated with sentiment s and aspect t is

$$
\psi_{s,t,a_0} = \frac{n^t_{s,a_0} + \gamma_t}{\sum_{t=1}^{T}(n^t_{s,a_0} + \gamma_t)} \tag{3.7}
$$

Finally, the approximate probability of a segment in document d being assigned with sentiment s is

$$
\pi_{d,s} = \frac{n^s_d + \beta_s}{\sum_{s=1}^{S}(n^s_d + \beta_s)} \tag{3.8}
$$

3.5 Experiments

To assess the generalizability of Arch, we prepare four online review datasets from two domains. TripUser, TripType, and TripLoc are collections of hotel reviews from TripAdvisor. TripUser contains 28,165 reviews posted by 202 randomly selected reviewers, each of whom contributes

at least 100 hotel reviews. TripType is a subset of TripUser in which reviews are associated with trip types (specialized by reviewers) including business, couple, family, friend, and solo. TripLoc contains a total of 136,446 reviews about seven US cities, split approximately equally. YelpUser, selected from [Yelp, 2014], contains 23,874 restaurant reviews posted by 144 users, each of whom contributes at least 100 reviews.

We remove stop words and HTML tags, expand typical abbreviations, and mark special named entities using a rule-based algorithm (e.g., replace a URL by #LINK# and replace a monetary amount $78.99 by #MONEY#) and Stanford named entity recognizer [Finkel et al., 2005]. We use Porter's [1980] stemmer algorithm. To handle negation, for any word pair whose first word is *no*, *not*, or *nothing*, we simply replace the word pair by attaching prefix *not* to its second word. Finally, we split each review into segments using the rule-based algorithm introduced in Section 2.4.1. Table 3.2 summarizes our datasets.

Table 3.2: Summary of the evaluation datasets.

Property	**TripUser**	**TripType**	**TripLoc**	**YelpUser**
Number of reviews	28,165	22,984	136,446	23,874
Number of segments	429,091	359,601	1,667,028	373,546
Average segments per review	15	16	12	16
Average words per segment	6	6	6	5

3.5.1 Parameter Settings

Arch, as a topic model, includes three manually tuned hyperparameters that influence its sampling. Hyperparameter α is the Dirichlet prior of the word distribution φ. Jo and Oh [2011] and Lin et al. [2012] show that, for sentiment and aspect discovery, models using an asymmetric prior can produce better results than those using a symmetric prior. In Arch, prior knowledge includes a sentiment word list shown in Table 3.3. This list extends the one used in Turney and Littman's [2003] work. For any word in the positive list, α is set to 0 if this word appears in a segment assigned as negative, and is set to 1 if this word appears in a segment assigned as positive, and conversely for words in the negative list. For all the other words, α is set to 0.01. Hyperparameter β is the Dirichlet prior of the sentiment distribution π. We set it to the approximate ratio of reviews with high ratings (≥ 3) to reviews with low ratings (< 3). Hyperparameter γ is the Dirichlet prior of the aspect distribution ψ. For LDA, ASUM, and ASUM using segments (ASUM-S), ψ is aspect distribution of a review. We set it to $50/T$. For Arch,

ψ is aspect distribution of all the reviews posted by an author. We set it to $50/T \times R$, where R is the average number of reviews for authors. We set the number of sentiments, S, to two including positive and negative, although we can modify parameter settings for more sentiment categories.

Table 3.3: Lists of sentiment words.

Positive
good, nice, excel, posit, fortun, correct, love, fantast attract, awesom, best, comfort, enjoi, amaz, favorit fun, glad, great, happi, impress, superior, like, perfect satisfi, worth, upgrad, not_bad, recommend, free thank, yum
Negative
bad, nasti, poor, neg, unfortun, wrong, inferior, annoi hate, junk, mess, not_good, not_lik, not_recommend unaccept, upset, wast, worst, worthless, small, nois complaint, terribl, troubl, problem, regret, complain old, dislik, sorri, not_worth, disappoint, dirti

3.5.2 Sentiment Aspect Discovery

Our first experiment shows how Arch discovers sentiment-aspect pairs. We apply Arch to data from two domains, hotels (TripUser) and restaurants (YelpUser). The number of aspects, T, affects the granularity and redundancy of the generated sentiment-aspect pairs. We vary the number of aspects and find that Arch produces minimal redundancy when running on TripUser and YelpUser with 20 and 30 aspects, respectively. We manually assign an aspect for each cluster of words. For TripUser, we omit *hotel* and *room* to reduce clutter.

Table 3.4 shows that Arch successfully discovers distinguishable aspects associated with sentiments from hotel reviews. For example, for aspect *Breakfast*, on the positive side, Arch discovers words describing the variety of food, such as *fruit*, *cereal*, and *juice* along with positive words such as *good* and *fresh*. In contrast, on the negative side, Arch discovers negative words such as *small* and *only*. For aspect *Internet*, the positive side contains words for praise, such as *free* and *good*, whereas the negative side has words for blame, such as *not_work* and *slow*.

Some aspects are only associated with one sentiment; e.g., aspect *Noise* is only negative.

It contains words such as *noise*, *hear*, and *loud*. We imagine that reviewers use these words to complain about noisy rooms. In addition, Arch discovers special markers in some aspects: *#LOC#* ranks high in aspect *View* and *NearbyArea*, and *#MONEY#* in aspect *Cost*, indicating that reviewers tend to mention monetary amounts when expressing their displeasure of the price.

We observe that aspects associated with restaurants are more complex than those for hotels because of the variety of cuisines. Titov and McDonald's [2008b] MG-LDA model performs well for hotel reviews but discovers only few ratable aspects from restaurant reviews, which they suggest is caused by the relatively small occurrences of words describing aspects for specific cuisines, such as *Italian*, and for general categories, such as *Meat*, compared with the words describing major aspects, such as *Service* or *Atmosphere*.

Arch yields promising results for restaurant reviews. Table 3.5 shows words describing specific cuisines such as aspect *Pizza*, *Asian* and *Dessert*. Arch discovers words describing general categories such as *Meat*. The positive side of *Meat* contains words describing tasting well, whereas the negative side involves words *dry*, *bland*, and *not_good*. We see similar comparisons for *Size* and *Service*.

3.5.3 Qualitative Evaluation

Predictive metrics, such as perplexity, are commonly used to quantitatively evaluate topic models since they do not require any human supervision or external knowledge. The basic idea of these metrics is to evaluate a model's ability of estimating the probability of unseen held-out documents. Therefore, a better model can yield a higher probability of held-out testing datasets. Recent work [Chang et al., 2009] has shown that predictive metrics are negatively correlated with human judgments of topic quality. In fact, whether topics (word clusters) are semantic cohesive is an important factor for human judges to assess a topic model. However, these metrics do not contain any variables for capturing semantic information. They just measure how well a model fits the data.

Mimno et al. [2011] introduce a new automated metric, topic coherence, for topic model evaluation. They show that the topic coherence well correlates with human judgments. Thus, we exploit this metric to quantitatively assess Arch. Given a sentiment-aspect pair p and its top N words $\{w_1^p, \ldots, w_N^p\}$, we compute the topic coherence score of p as follows.

This is caused by the redundancy in aspects. If the specified number of aspects is larger than the data truly supports, Arch mistakenly clusters some function or non-sentiment words into an aspect with a sentiment.

$$C(p) = \sum_{i=2}^{N} \sum_{j=1}^{i} \log \frac{D(w_i^p, w_j^p) + 1}{w_j^p} \tag{3.9}$$

where $D(w_i^p, w_j^p)$ counts the number of reviews containing both w_i^p and w_j^p, $D(w_j^p)$ counts the number of reviews containing w_j^p, and a count of 1 is to avoid the logarithm of zero.

For comparison Arch, we include three other topic models: LDA, ASUM, and ASUM-S. We evaluate on two datasets including TripUser and YelpUser. For each aspect number, we first randomly select 50% of each dataset ten times. Then we conduct a paired t-test for each of the pairwise comparisons.

Figure 3.5 shows average topic coherence scores of each model on TripUser given different number of aspects. Table 3.6 shows average topic coherence scores with standard deviation. We see LDA performs worst among all models. This may be caused by the undesirable mixture of words with different sentiments in same aspects. ASUM outperforms LDA. ASUM-S consistently and significantly yields better topic coherence score than ASUM with p-value less then 0.001, which indicates the benefit of using segments as the basic sentiment expressing units. Arch further improves ASUM-S and yields the highest topic coherence scores for all of the numbers of aspects. The improvements are statistically significant with p-values less than 0.05. Figure 3.6 and Table 3.7 shows results on YelpUser. Similar conclusions hold except that Arch and ASUM-S achieve similar topic coherence scores when the number of aspects equal to 50 and 60. Table A.1 and A.2 show p-values.

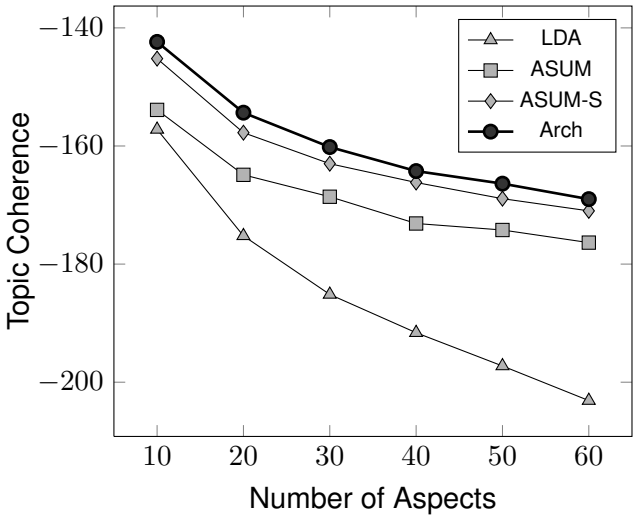


Figure 3.5: Topic coherence score on hotel reviews with different numbers of aspects.

Table 3.6: Average topic coherence scores with standard deviation on hotel reviews

	Arch	ASUM-S	ASUM	LDA
10	-142.38±2.02	-145.21±1.31	-153.89±2.91	-157.20±4.24
20	-154.36±2.52	-157.75±1.83	-164.89±2.33	-175.22±3.69
30	-160.18±1.62	-163.00±0.89	-168.60±2.39	-185.14±2.53
40	-164.25±1.39	-166.19±1.54	-173.13±1.63	-191.63±2.53
50	-166.38±2.54	-168.93±2.78	-174.23±2.56	-197.26±2.73
60	-169.00±2.41	-170.99±1.82	-176.37±1.78	-203.12±2.68

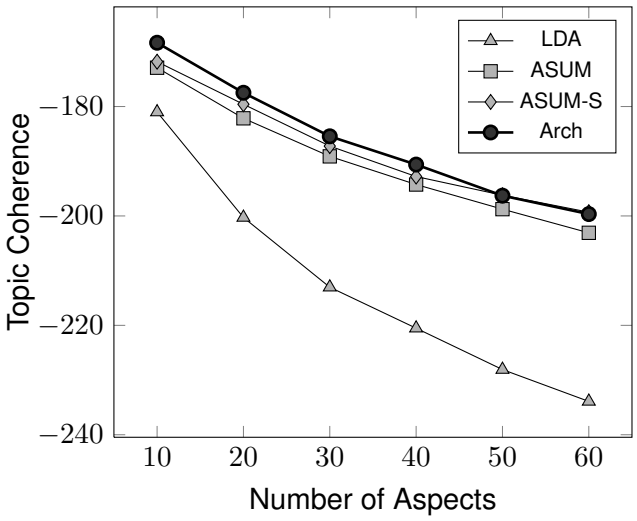


Figure 3.6: Topic coherence score on restaurant reviews with different numbers of aspects.

Table 3.7: Average topic coherence scores with standard deviation on restaurant reviews

	Arch	ASUM-S	ASUM	LDA
10	-168.34±2.49	-171.81±1.78	-172.91±2.07	-181.01±3.62
20	-177.49±1.46	-179.59±1.72	-182.16±2.51	-200.28±3.24
30	-185.45±2.25	-187.24±1.83	-189.12±2.91	-213.03±2.00
40	-190.61±2.07	-192.80±2.31	-194.26±1.76	-220.53±2.36
50	-196.30±1.72	-196.22±2.00	-198.76±2.22	-228.08±3.26
60	-199.66±2.34	-199.32±2.06	-203.09±2.22	-233.91±3.07

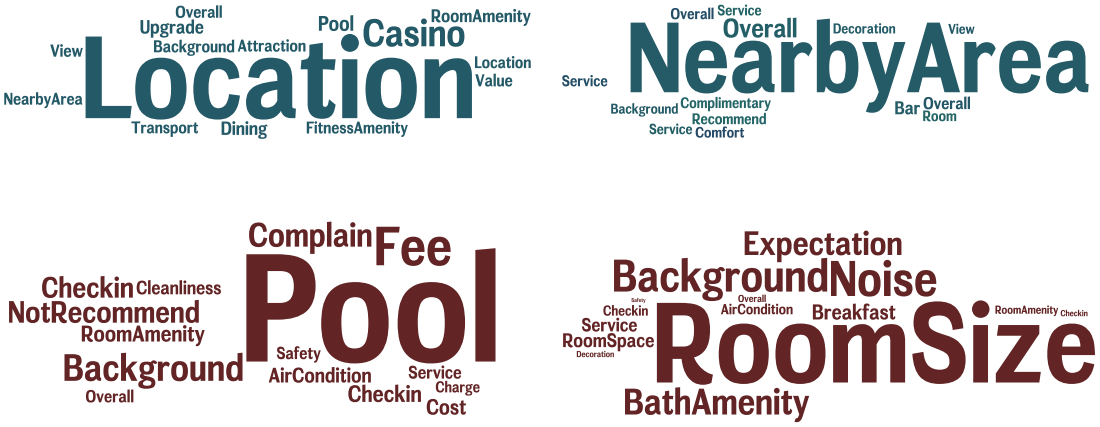
3.5.4 Author Preference Profile

Summarization

An important output of Arch is interpretable author profiles. Such profiles summarize authors' preferences in terms of sentiments and aspects when they write reviews. To evaluate author profiles, we run Arch on datasets created by using different attributes of reviews. TripLoc contains reviews associated with seven cities: we consider reviews of each city as written by one author. TripType contains reviews related to five trip types; we consider reviews of each type as written by one author. For TripUser, we consider individual authors. We compare the profiles generated by Arch with respect to their characteristics.

We visualize author profiles as aspect-clouds using the top 15 aspects in each sentiment. The size of the aspect corresponds to its score. We compute the aspect score using ψ learned from the data. The computing method is similar to term scores [Srivastava and Sahami, 2009, p. 75]. The aspect score is low for an aspect that has a high probability under a sentiment for all authors. In this way, the most representative aspects score higher than common aspects.

Figure 3.7 shows the preference profiles of four US cities generated by Arch. We include top five aspects of Boston, Chicago, and Orlando in Table 3.8. Appendix A.2 provides a visualization of preference profiles for all seven cities. These profiles provide a salient summary of reviews in regard to each city. For example, *Location* and *Casino* are the top two positive aspects for Las Vegas, a resort city for gambling. An inspection of the reviews reveals that most of the hotels with high ratings are located on the Strip, an important route. For Boston, Chicago, and New York, *NearbyArea* ranks on top on the positive side. Most hotels in these cities are located in the metropolitan area, so proximity to shopping, restaurants, and attractions is appealing. For the negative side, we see *Expectation* appears in the top five. These three cities are among the most expensive cities in the US ranking by daily hotel room rates [Statista, 2014]. Generally,



Las Vegas *New York*

Los Angeles *Miami*

Figure 3.7: An aspect-cloud visualization of U.S. cities (positive aspects above; negative aspects below).

consumers expect more when they pay more. We infer from the figure that a failed expectation could be caused by room size, especially for New York, which accords with the fact that room sizes in New York tend to be smaller than elsewhere in the US [NYC, 2014]. For Miami, aspect *Location* is the most important positive aspect. Miami is known as the "Cruise Capital of the World", and many travelers stay there before or after taking their cruise. Therefore, *Location* and *Transportation* are attractive aspects.

Figure 3.8 shows the results for TripType. We observe that *Internet*, *Service*, and *Decoration* are most likely to lead to a negative sentiment for business travelers. *Attraction* is most positive for both couples and families, and *View* additionally for couples. Solo travelers, business or tourist, express most opinions toward both *Attraction* and *Location*.

Table 3.8 (bottom rows) lists the top five aspects for TripUser. For Author 1, it is clear that *Room* and *Service* are the most attractive aspects and *Cleanliness* and *Checkin* are most likely to lead to a negative sentiment. Author 1 tends to summarize positive opinions when writing reviews. We also observe that *Service* is the most important aspect for Author 2, who expresses most opinions toward *Service* on both positive and negative sides. Authors 3 and 4 are similar in that *Business*, *NearbyArea*, and *Transportation* are positive and *RoomAmenity* and *Pool* are negative for both of them.

Similarity

Preference profiles can be used not only for summarization, but also for measuring similarities among authors, which can support user-based applications such as recommender systems. We adapt the Jensen-Shannon distance (JSD) [Endres and Schindelin, 2003] as the metric to measure similarity between authors.

JSD is a metric defined as the square root of the JensenShannon divergence. Given author profiles P and Q, we can compute their JSD, D_{JS}, as follows.

$$\begin{aligned} D_{JS} &= \sqrt{\frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M)} \\ M &= \frac{1}{2}(P+Q) \end{aligned} \tag{3.10}$$

where D_{KL} is the Kullback-Leibler (KL) divergence. Given two probability distributions $P = \{p_1, \ldots, p_n\}$ and $Q = \{q_1, \ldots, q_n\}$. We calculate their KL divergence, $D_{KL}(P||Q)$, as follows.

Business *Couple*

Solo *Family*

Figure 3.8: An aspect-cloud visualization of different trip types (positive aspects above; negative aspects below).

$$D_{KL}(P||Q) = \sum_i p_i \log \frac{p_i}{q_i} \tag{3.11}$$

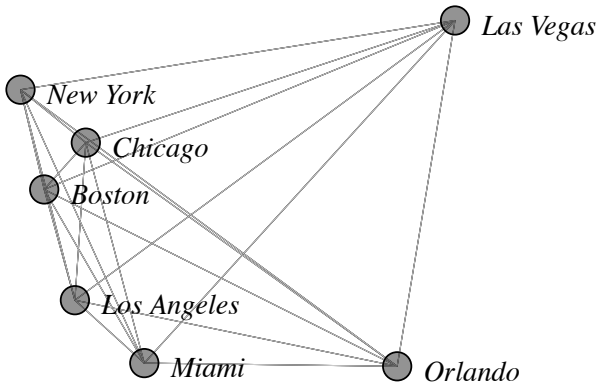


Figure 3.9: Similarities among cities as force fields.

Figure 3.9 shows the similarities among the profiles of the seven cities (here dissimilarity corresponds to distance). Boston, Chicago, and New York are close to each other; Los Angeles and Miami are close to each other; Las Vegas is far away from any of the others; Orlando is far away from the others, except Miami. These are reasonable results. As we discussed earlier, hotels in Boston, Chicago, and New York have common characteristics; Las Vegas differs strongly from the others because it is a resort city and most hotels there are combined with casinos; Orlando is a tourism destination but differs from Las Vegas in that it is famous for local attractions, such as theme parks. Orlando's profile exhibits that travelers there tend to be more aware of aspect *Attraction* than elsewhere. An interesting pair is Los Angeles and Miami. We see that *Location* appears as the most important aspect on the positive side for both of them. Such a similarity could be partially explained by the role of the city. Both Los Angeles and Miami serve as locations for taking cruises. Also, the common aspect *Safety* (negative for both) could increase their similarity.

Figure 3.10 shows similarities among the five trip types. Business is far from others but is closer to Friend and Solo than Couple and Family. Couple, Family, and Friend are close to each other. Business reviewers attend to different aspects from others. Further, Solo and Friend contain reviews of business trips, although the authors did not select Business as the trip type. However, this situation does not happen for Couple and Family.

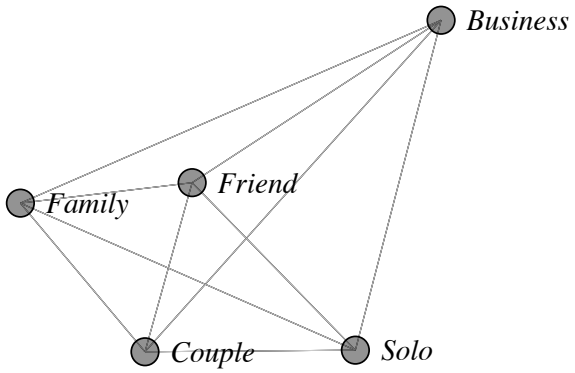


Figure 3.10: Similarities among trip types as force fields.

Figure 3.11 shows similarities among 20 authors, including four authors we mentioned earlier. We can see that Authors 1 and 2 are far away from each other. On the contrary, Authors 3 and 4 are close to each other. By using the relative distance among these authors and k-nearest neighbors algorithm ($k = 1$), we split them into four clusters for display.

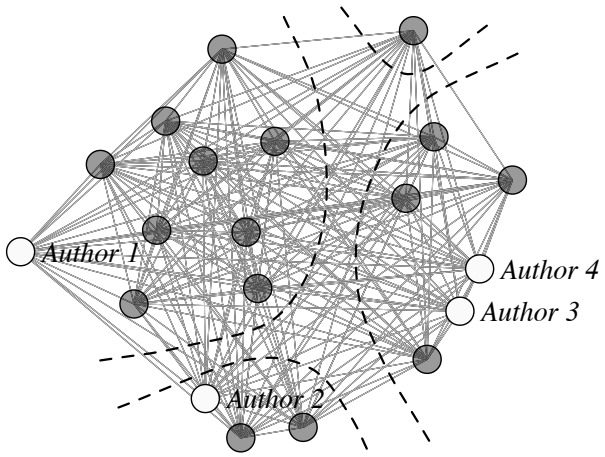


Figure 3.11: Similarities among 20 authors as a force field.

3.5.5 Authorship Attribution

Authorship attribution is a research topic with a long history [Mendenhall, 1887; Mosteller and Wallace, 1963]. A typical task is to identify the author of a text of unknown authorship given a collection of text samples written by a set of candidate authors. Existing studies [Zheng et al., 2006; Savoy, 2013; Nasir et al., 2014] focus on exploring and selecting various types

of features that can capture distinctive traits of authors, such as writing style or topic focus. For opinionated text, the discriminating traits of authors are their preferences with respect to sentiments and aspects. In this experiment, we demonstrate the benefits of using author profiles for authorship attribution.

To determine the author of query texts, we compute perplexity using a set of candidate author profiles. Perplexity introduced by Bahl et al. [1983] is a measurement of the ability of a probabilistic model to generalize unseen testing data. Given a training dataset D^{train} and a testing dataset D^{test}. We compute the perplexity as follows.

$$perplexity = \exp(-\frac{\log\ p(D^{test}|D^{train})}{\sum_{d=1}^{D^{test}} n_d}) \tag{3.12}$$

where n_d is the number of words in review d. In our experiment, we approximate $p(D^{test}|D^{train}$ as the harmonic mean [Griffiths and Steyvers, 2004] of a set of $p(D^{test}|M)$ sampled from $p(M|D^{train})$, where M refers to the set of parameters in a model. We obtain 20 samples of M using Gibbs sampling.

Given a candidate author a_0, we compute $p(D^{test}|M_{a_0})$ as follows.

$$\begin{aligned} p(D^{test}|M_{a_0}) &= \prod_{d=1}^{D} p(\boldsymbol{w}_d|M, a_0) \\ &= \prod_{d=1}^{D}\prod_{w=1}^{W}(\sum_{t=1}^{T}\sum_{s=1}^{S}\varphi_{s,t,w}\psi_{s,t,a_0}\pi_{d,s})^{n_d^w} \end{aligned} \tag{3.13}$$

where D is the number of reviews in query texts. n_d^w is number of times that word w appears in review d.

The author predicted is the one who yields the lowest perplexity. We randomly select 20 authors from TripUser and YelpUser. We use 80% of their reviews as training datasets. By varying the percentage of the remaining 20% query texts, we build five testing datasets for each domain. This simulates identifying authorship by observing reviews incrementally.

Table 3.9: Description of character Features used in our baseline model.

Character Features	**Description**
Number of characters	
Number of letters	A–Z, a–z
Number of numeric characters	0–9
Number of white space	
Number of tab stops	
Number of upper-case letters	A–Z
Frequency of 26 letters	
Frequency of punctuations	@, #, $, %, ˆ, &, *, -, _, =,+, <,>, ˜, |, \, /

Table 3.10: Description of word Features used in our baseline model.

Word Features	**Description**
Number of words	
Number of short words	Words consist of less than four letters
Average length of words	
Number of different words	
Average characters per sentence	
Average words per sentence	
Brunets W measure	Brunet's measure of lexical richness
Honores R measure	Honore's measure of lexical richness
Sichels S measure	Sichel's measure of lexical richness
Simpsons D measure	Simpson's measure of lexical richness
Yules K measure	Yule's measure of lexical richness
Hapax legomena	Frequency of words that occur once
Hapax dislegomena	Frequency of words that occur twice
Frequency of words with different length	Length ranges from 1 to 20

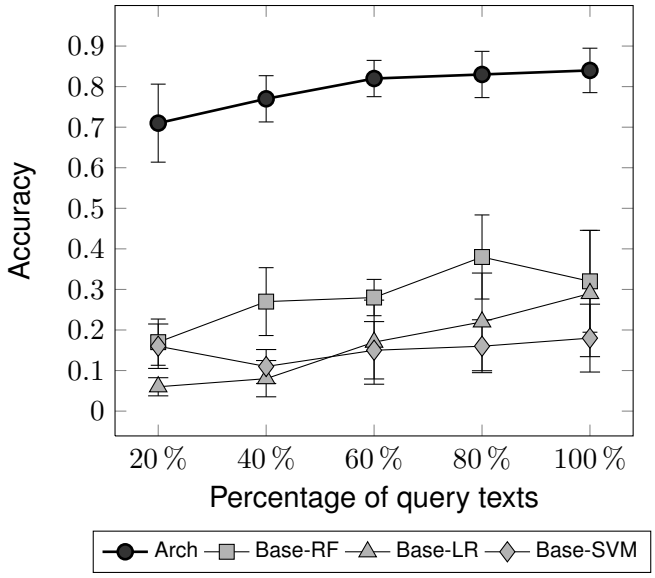


Figure 3.12: Accuracy of authorship attribution on hotel reviews with different sizes of query texts.

We build baseline models using a collection of features associated with writing styles. Table 3.9 and 3.10 list character and word features, respectively. We train three classifiers in WEKA [Hall et al., 2009]: random forests (Base-RF), logistic regression (Base-LR), and support vector machine (Base-SVM). Note that this task is challenging even for humans since all the reviews describe the same global topic *hotel* with 20 possible labels for each prediction, so random guessing would yield accuracy of 5%.

Figure 3.12 shows the prediction accuracy on hotel reviews, with error bars representing the standard deviations of five-fold cross-validation. All baseline models perform poorly on this task. For Arch, we observe that as the size of query texts increases, its predictive power improves. Predicting on all query texts yields the highest average accuracy of 84%.

Figure 3.13 shows the prediction accuracy on restaurant reviews. Similar to the results obtained with hotel reviews, we observe that Arch achieves the highest accuracy among all models. Base-RF yields the best prediction accuracy among baseline models.

3.5.6 Sentiment Classification

We now evaluate the effectiveness of Arch for document-level binary sentiment classification. We use two datasets, TripUser and YelpUser.

To collect ground-truth labels, we use integer ratings from [1, 5] associated with reviews: four and five (positive), one and two (negative), and ignore the rest. For both TripUser and

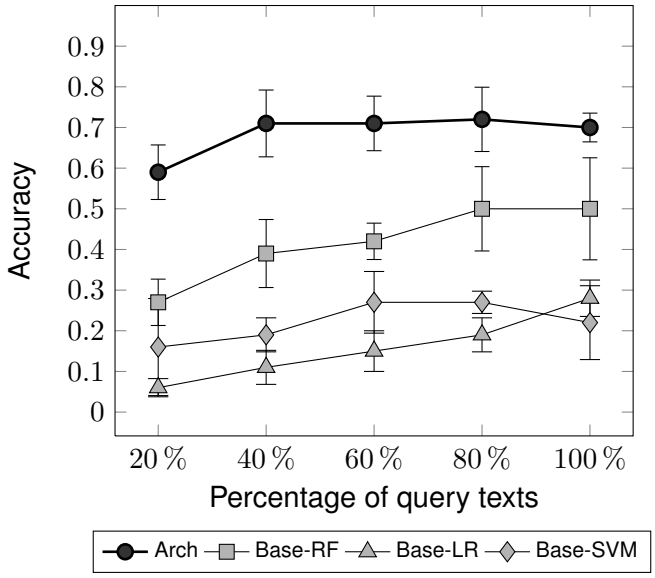


Figure 3.13: Accuracy of authorship attribution on restaurant reviews with different sizes of query texts.

YelpUser, we use 80% of each author's reviews as training datasets. To build balanced testing datasets, we sample equal number of positive and negative reviews from the remaining 20% reviews of each author.

We compare Arch with ASUM, ASUM using segments (ASUM-S), and LingPipe [Alias-i, 2008] using unigrams (Ling-Uni) and bigrams (Ling-Bi). Ling-Uni and Ling-Bi are supervised models. Arch, ASUM, and ASUM-S use the same hyperparameter settings. Although we didn't tune the hyperparameters as such, the values chosen are compatible with those in the literature and yield better results (for ASUM and Arch) than other values we considered. We use the following strategy to assign sentiments to reviews. In Arch, we assign a sentiment to a review as the sentiment that occurs most commonly over its segments. For ASUM and ASUM-S, we assign sentiments using sentiment distributions associated with reviews, as suggested by Jo and Oh.

Figure 3.14 reports the classification accuracy on hotel reviews, with error bars indicating the standard deviations of five-fold cross-validation. Arch outperforms all unsupervised models and the supervised model Ling-Uni with average gains of 10%. ASUM-S is consistently better than ASUM, indicating that using segments as the basic sentiment units is effective. When the number of aspects is equal to 10, Arch yields comparable accuracy to Ling-Bi. As the number of aspects increases, the accuracy decreases for all unsupervised models. This is caused by the redundancy in aspects. If the specified number of aspects is larger than the data truly supports, Arch mistakenly clusters some function or non-sentiment words into an aspect with a sentiment.

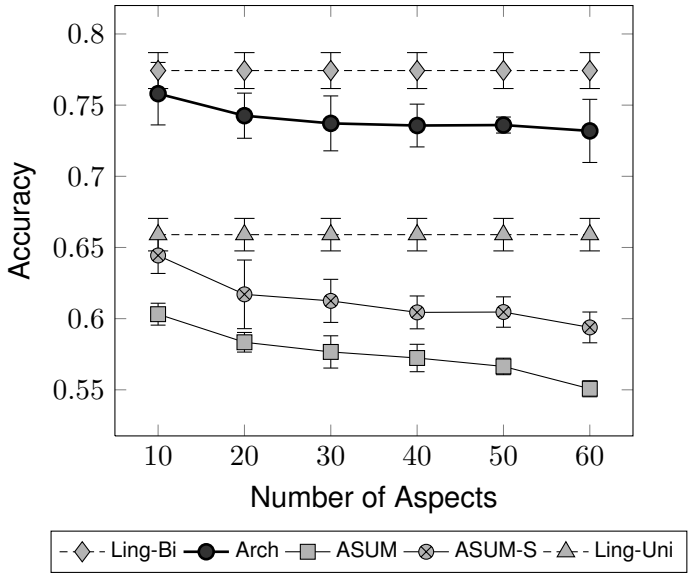


Figure 3.14: Accuracy of sentiment classification on hotel reviews with different numbers of aspects.

Figure 3.15 shows the accuracy of sentiment classification on restaurant reviews, with error bars indicating the standard deviations of five-fold cross-validation. We obtain similar results to those obtained with hotel reviews. Arch outperforms all unsupervised models and the supervised model, Ling-Uni, with average gains of 8%. Note that Arch achieves the best average classification accuracy of 71% with aspect number equal to 30. This is because of the fact that there are more aspects mentioned in restaurant reviews.

3.6 Related Work

In this section, we survey related work in two areas including sentiment aspect discovery and opinion summarization.

3.6.1 Sentiment Aspect Discovery

The problem of sentiment and aspect discovery has been extensively studied in recent years. Approaches based on Latent Dirichlet Allocation (LDA) [Blei et al., 2003] have been successfully applied to analyze the content of textual information. In LDA, a document is represented as a mixture of topics, each a multinomial distribution over words. The learning process approximates the topic and word distributions based on their co-occurrence in documents. Titov and McDonald's [2008b] model handles global and local topics involved in documents, and their

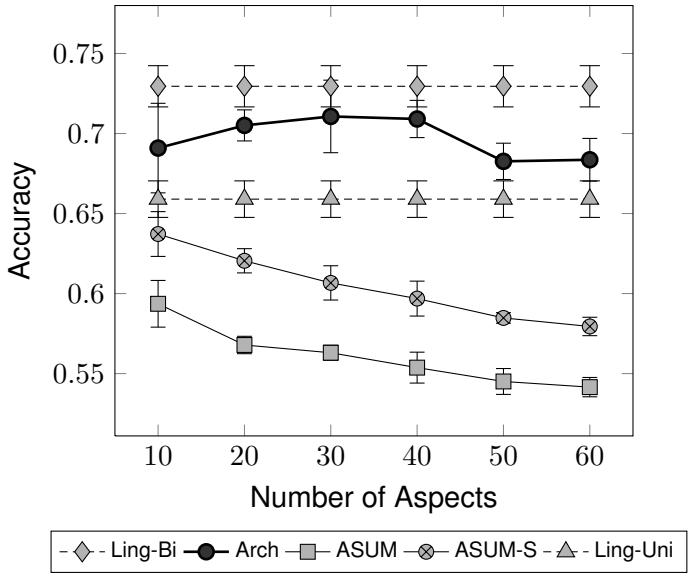


Figure 3.15: Accuracy of sentiment classification on restaurant reviews with different numbers of aspects.

[2008a] framework discovers topics using aspect ratings provided by reviewers. JST [Lin et al., 2012] and ASUM [Jo and Oh, 2011] model a review as two multinomial distributions, over topics and sentiments, respectively. They condition the probability of generating words on both topics and sentiments. Kim et al. [2013] extend ASUM by allowing its probabilistic model to discover a hierarchical structure of aspect-based sentiments. Although the above models produce reasonable results, they omit author information, which is naturally associated with opinionated text.

Rosen-Zvi et al.'s Author Topic model (AT) [2004] can capture such associations by building a topic distribution for each author. When generating a word in a document, AT conditions the probability of the topic assignment on the author of the document. Similarly, Kim et al. [2012] propose a topic model that captures entities mentioned in documents. In their model, the probability of generating a word is conditioned on both entity and topic. Diao and Jiang's [2013] model jointly considers topics, events, and users on Twitter. Although the above models capture authors associated with texts, none of them can handle sentiments.

Mukherjee et al. [2014] propose JAST, a model that jointly considers authors, sentiments, and topics. However, JAST requires ratings information for training. We have observed that, in online review datasets, authors generally have different ratings distributions. Such a diversity of ratings not only impacts learning by JAST but also limits the applications of the authors' preferences generated. For example, we cannot directly measure similarity between two authors if they have different ratings distributions.

3.6.2 Sentiment Summarization

Give a collection of reviews, we often desire to generate a summary of sentiments. Such a summary can help understand how people feel about certain entities or aspects without laboriously go through all of the reviews. Sentiment summarization has attracted increasing attention over the past few years.

As a pioneer work, Hu and Liu [2004] propose a summarization system for product reviews. Their system uses association mining to extract product features from reviews. These product features are then used to select and order sentences. To label sentiments for sentences, they build a sentiment lexicon by iteratively expanding a small set of sentiment words using synonyms and antonyms in WordNet. Similar approaches are also used in [Liu et al., 2005] and [Popescu and Etzioni, 2005].

Ontologies are commonly used in sentiment summarization systems to guide the extraction process. Tata and Di Eugenio [2010] develop a summarization system for song reviews. Given a set of reviews related to a song. Their system first uses a set of heuristics to extract feature terms from each sentence. Then it uses these feature terms to regenerate each sentence in a compact way. After that, it groups sentences based on common n-grams and sentiments. Finally it selects representative sentences using a music ontology to generate a summary of reviews. Lu et al.'s [2010] approach first maps sentences of reviews to a set of aspects using ontologies. It then selects aspects based on size, opinion coverage, and conditional entropy. After that, it generates summaries using aspects ordered by a coherence measure.

Ganesan et al. [2010] propose a graph-based summarization system. It formulates sentences as graphs and uses properties of graphs to generate abstractive summaries. However, their system does not contain any component for sentiment analysis. Therefore, the generated summaries may contain conflicting information. For example, one aspect could be associated with two sentiments.

Compared with the above approaches, Arch generates sentiment summaries without relying on any predefined heuristics or ontologies. The only prior knowledge is a list of 64 sentiment words. Therefore, Arch is more general and can be easily applied to different domains.

3.7 Conclusions

Arch provides an unsupervised way to discover sentiments and aspects from opinionated text. By incorporating authors as a factor, Arch captures the association of sentiments and aspects with authors and generates interpretable author profiles describing their preferences in terms of sentiments and aspects. Our experiments show that the author profiles provide salient authors' preference summaries that are well correlated with ground truth. We also demonstrate

the effectiveness of using the author profiles for authorship attribution. Finally, we show that Arch achieves better sentiment classification accuracy than a state-of-the-art topic modeling approach for document-level sentiment classification. In future work, we plan to apply Arch to recommender systems based on the similarity among authors.

Table 3.4: Top words discovered for sentiment-aspect pairs from hotel reviews.

Service	Service	Breakfast	Breakfast	Room	Room	Cost
P	N	P	N	P	N	N
staff	servic	breakfast	coffe	clean	bed	park
friendli	staff	egg	breakfast	comfort	small	#MONEY#
help	would	fruit	water	bed	bathroom	charg
desk	stai	cereal	small	nice	pillow	car
servic	problem	juic	food	well	size	pai
front	onli	coffe	tea	good	doubl	fee
checkin	could	bread	servic	bathroom	bit	cost
nice	thing	good	bar	larg	space	valet
effici	properti	fresh	drink	spaciou	sheet	night
good	place	hot	onli	quiet	onli	onli
pleasant	manag	buffet	bottl	modern	littl	lot
recept	guest	chees	order	new	side	taxi
great	#LOC#	cold	tabl	great	mattress	internet
welcom	time	waffl	ask	size	standard	hour
excel	need	sausag	even	decor	hard	#LOC#
Internet	Internet	Decoration	Cleanliness	View	NearbyArea	Noise
P	N	P	N	P	P	N
free	internet	like	old	view	walk	nois
internet	work	look	carpet	pool	locat	could#
wifi	onli	nice	need	nice	restaur	door
park	not_work	modern	bathroom	area	#LOC#	hear
work	desk	lobbi	bit	great	shop	night
lobbi	wifi	decor	look	floor	minut	quiet
access	connect	design	small	larg	good	#TIME#
good	phone	area	date	good	street	window
comput	plug	feel	furnitur	#LOC#	distanc	street
well	could	style	littl	balconi	within	light
avail	tv	build	wall	love	area	loud
wireless	time	#LOC#	worn	look	close	next
breakfast	slow	beauti	dirti	overlook	right	traffic
center	busi	larg	floor	locat	great	onli
busi	comput	comfort	clean	beauti	awai	morn

Table 3.5: Top words discovered for sentiment-aspect pairs from restaurant reviews.

Meat	Meat	Pizza	Dessert	Asian	Drink	Interjection
P	N	P	P	P	P	N
flavor	bit	pizza	chocol	chicken	beer	not
like	littl	good	dessert	roll	wine	oh
good	tast	order	cake	rice	drink	well
sauc	flavor	egg	like	dish	good	ye
tast	dry	like	good	order	select	no
nice	sauc	chees	order	good	like	yeah
cook	bland	sausag	bread	fri	glass	lol
fresh	realli	breakfast	sweet	soup	order	ok
meat	could	crust	top	noodl	#MONEY#	know
sweet	not_good	sandwich	delici	sauc	tap	why
well	chees	love	butter	beef	bottl	said
bread	ok	thin	cooki	spici	tea	wtf
chicken	lack	pepperoni	cream	like	great	sorri
littl	kind	sauc	tast	thai	ic	ugh
realli	thought	bacon	flavor	curri	margarita	digress
Size	Size	Service	Service	Atmosphere	Deal	Overall
P	N	P	N	P	P	P
portion	small	servic	servic	nice	#MONEY#	would
#MONEY#	salad	great	littl	patio	free	again
good	order	food	bit	tabl	card	place
size	plate	friendli	place	bar	coupon	definit
like	menu	good	slow	like	onli	time
meal	#MONEY#	staff	onli	seat	off	good
half	side	nice	small	place	meal	like
sandwich	portion	place	realli	insid	good	return
price	onli	atmospher	loud	area	like	food
could	serv	clean	time	room	place	worth
food	size	price	park	dine	tip	soon
larg	rice	realli	restaur	restaur	drink	order
huge	sandwich	excel	pretti	decor	time	visit
order	would	fast	food	look	deal	look
enough	item	attent	music	outsid	gift	star

Table 3.8: Top five aspects discovered by Arch.

Boston (P)	**Boston (N)**	**Chicago (P)**	**Chicago (N)**
NearbyArea	Expectation	NearbyArea	Expectation
Overall	NearbyArea	Overall	Parking
Background	RoomSize	Overall	RoomSize
Decoration	Parking	Decoration	Background
Complimentary	Noise	Bar	Decoration
Orlando (P)	**Orlando (N)**	**Friend (P)**	**Friend (N)**
Attraction	Background	Location	Room
Pool	Cleanliness	Attraction	Breakfast
Dining	Pool	View	Fee
RoomAmenity	NearbyArea	Breakfast	Bathroom
Location	Complain	NearbyArea	Pool
Author 1 **(P)**	**Author** 1**(N)**	**Author** 2 **(P)**	**Author** 2 **(N)**
Overall	Cleanliness	Service	Service
Service	Checkin	Background	Service
Room	Dining	Attraction	Pool
Background	Background	Value	Breakfast
Recommend	Service	Decoration	Bathroom
Author 3 **(P)**	**Author** 3 **w(N)**	**Author** 4 **(P)**	**Author** 4 **(N)**
Bathroom	RoomAmenity	Transportation	Bathroom
Business	Breakfast	NearbyArea	RoomAmenity
NearbyArea	Checkin	Recommend	Dining
Transportation	Comfort	Business	Background
Room	Pool	Dining	Pool

Chapter 4

Conclusions and Future Work

Sentiment analysis and opinion mining is a challenging and intriguing research field. Although impressive advances have been made over the past few years, we are still far from a satisfactory solution. Most of the challenges we face stem from the complexity and diversity of linguistic representations. In this dissertation, we identify four challenges that may impede current research progress and focus on developing general and robust sentiment analysis systems that require minimum human efforts. Accordingly, we propose the concept of segments and develop two segment-based sentiment analysis approaches, ReNew and Arch.

ReNew is a semi-supervised sentiment analysis framework that leverages unlabeled opinionated texts to automatically generate a domain-specific sentiment lexicon and train a sentiment classifier. The domain-specific sentiment lexicon uses dependency relation pairs as basic elements to capture contextual information surround words. The sentiment classifier leverages the relationships between consecutive segments to infer their sentiments. We evaluate the effectiveness of ReNew using hotel reviews from TripAdvisor. Empirical results show that our framework can greatly reduce the human effort for building a domain-specific sentiment lexicon with high quality. Specifically, in our evaluation, working with just 20 manually labeled reviews, it generates a domain-specific sentiment lexicon that yields weighted average F-Measure gains of 3%. Our sentiment classification model achieves approximately 1% greater accuracy than a state-of-the-art approach based on elementary discourse units.

Arch is an unsupervised sentiment analysis model that leverages unlabeled opinionated texts to automatically discover sentiment-aspect pairs in terms of word-clusters of words. Arch generates interpretable author preference profiles that can be used for (1) summarizing authors preferences in terms of sentiments and aspects and (2) measuring similarities among authors. The learned probabilistic model can be used for (1) sentiment classification at document and segment level, (2) aspect classification at segment level, and (3) authorship attribution. We evaluate Arch using four datasets from two domains. We find that Arch successfully discov-

ers word-clusters describing aspects associated with sentiments. The generated author profiles are well correlated with ground truth. To exhibit the prospects for potential applications, we demonstrate the effectiveness of Arch for authorship prediction and sentiment classification.

4.1 Future Work

The results of ReNew and Arch open up many interesting directions for future work. In this section, we briefly introduce three directions.

Segmentation

Finding an optimal segmentation of a review is challenging due to the freedom of writing styles and the diversity of linguistic representations. Although our rule-based segmentation algorithm produces promising results, we can still easily observe that many reviews are either over-segmented or mis-segmented. These mistakes hurts the performance of sentiment analysis systems. For example, in ReNew, all of the mistakes propagate through the bootstrapping process since segmenting is the first step in each iteration. In future, we would like to explore more advance techniques to identify boundaries of segments. For example, we can use Semi-Markov Conditional Random Fields [Sarawagi and Cohen, 2004] to jointly perform segmenting and sentiment labeling for reviews.

High-order Relationships Learner

So far, ReNew has exploited two learners to learn forward and backward relationships between segments. Existing learners are limited to capture first-order relationships, relationships between adjacent segments. Therefore, ReNew cannot capture the discourse relationships that may span over more than two segments. For example, an elaboration relation generally consists of a segment of statement and a group of satellite segments for supporting the statement. In future, we would like to investigate the possibility of adding high-order relationship learners to capture relationships among segments that are not adjacent. In this way, we could take the advantage of using a more rich set of relationships for sentiment analysis.

Content-based Recommendation System

Researchers have extensively studied the problem of building recommendation systems. Most existing approaches generate user preference profiles by exploiting ratings and usage data associated with users without taking advantage of user generated contents, such as comments and reviews. In fact, user generated contents play an important role in revealing user preferences in many domains, such as hotel and book reviews. Building a component for analyzing user

generated contents is not a trivial task. It often requires domain knowledge and a considerable amount of labeled data for training. Arch provides an effective unsupervised solution for generating user preference profiles by analyzing user generated contents . We have demonstrated a way of using these profiles for measuring similarities among users. A very interesting future direction is to build a content-based recommendation system based on Arch.

REFERENCES

Alias-i. LingPipe 4.1.0. http://alias-i.com/lingpipe, 2008. Accessed: 23/05/2014.

Amazon. A review sentence from Amazon. http://www.amazon.com/review/RHENB70WRMML1/ref=cm, 2014. Accessed: 10/14/2014.

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, March 1983.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, February 2003.

Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

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

Johan Bollen, Huina Mao, and Xiao-Jun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, March 2011.

Margaret M. Bradley and Peter J. Lang. Affective norms for English words (ANEW): Instruction manual and affective ratings. Technical Report C-1, The Center for Research in Psychophysiology, University of Florida, Gainesville, FL, 1999.

Rebecca Bruce and Janyce Wiebe. Recognizing subjectivity: A case study of manual tagging. *Natural Language Engineering*, 5(2):187–205, June 1999.

Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms using different performance metrics. In *Proceeding of* 23^{rd} *International Conference on Machine Learning (ICML)*, pages 161–168, Pittsburgh, 2006.

Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of the* 23^{rd} *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 288–296, Vancouver, 2009.

Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z. Sui. Exploring millions of footprints in location sharing services. In *Proceedings of the* 5^{th} *International Conference on Weblogs and Social Media (ICWSM)*, Barcelona, 2011.

Yejin Choi and Claire Cardie. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the* 14^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 590–598, Singapore, 2009.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the* 25^{th} *International Conference on Machine Learning (ICML)*, pages 160–167, Helsinki, 2008.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the* 5^{th} *Conference on Language Resources and Evaluation (LREC)*, pages 449–454, Genoa, Italy, 2006.

Qiming Diao and Jing Jiang. A unified model for topics, events and users on Twitter. In *Proceedings of the* 18^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1869–1879, Seattle, 2013.

Xiaowen Ding, Bing Liu, and Philip S. Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the* 1^{st} *International Conference on Web Search and Web Data Mining (WSDM)*, pages 231–240, Palo Alto, 2008.

Cícero Nogueira dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the* 25^{th} *International Conference on Computational Linguistics (COLING)*, pages 69–78, Dublin, 2014.

Dominik Maria Endres and Johannes E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, July 2003.

Andrea Esuli and Fabrizio Sebastiani. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the* 5^{th} *Conference on Language Resources and Evaluation (LREC)*, pages 417–422, Genoa, Italy, 2006.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the* 43^{rd} *Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, 2005.

Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric K. Ringger. Pulse: Mining customer opinions from free text. In *Proceedings of the* 6^{th} *International Symposium on Intelligent Data Analysis (IDA)*, pages 121–132, Madrid, 2005.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the* 23^{rd} *International Conference on Computational Linguistics (COLING)*, pages 340–348, Beijing, 2010.

Walter R. Gilks, Sylvia Richardson, and David J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC, London, 1996.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceeding of* 28^{th} *International Conference on Machine Learning (ICML)*, pages 513–520, Bellevue, Washington, 2011.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Supplement 1):5228–5235, April 2004.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations Newsletter*, 11(1):10–18, November 2009.

Vasileios Hatzivassiloglou and Janyce Wiebe. Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the* 18^{th} *International Conference on Computational Linguistics (COLING)*, pages 299–305, Saarbrücken, Germany, 2000.

Yulan He. Learning sentiment classification model from labeled features. In *Proceedings of the* 19^{th} *ACM Conference on Information and Knowledge Management (CIKM)*, pages 1685–1688, Toronto, 2010.

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of* 15^{th} *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 289–296, Stockholm, 1999.

Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the* 10^{th} *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 168–177, Seattle, 2004.

Yohan Jo and Alice Haeyun Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the* 4^{th} *ACM International Conference on Web Search and Data Mining (WSDM)*, pages 815–824, Hong Kong, 2011.

Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the* 11^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–363, Sydney, 2006.

Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125, May 2006.

Hyungsul Kim, Yizhou Sun, Julia Hockenmaier, and Jiawei Han. ETM: Entity topic models for mining documents associated with entities. In *Proceedings of the* 12^{th} *IEEE International Conference on Data Mining (ICDM)*, pages 349–358, Brussels, 2012.

Soo-Min Kim and Eduard H. Hovy. Determining the sentiment of opinions. In *Proceedings of the* 20^{th} *International Conference on Computational Linguistics (COLING)*, pages 1367–1373, Geneva, 2004.

Suin Kim, Jianwen Zhang, Zheng Chen, Alice H. Oh, and Shixia Liu. A hierarchical aspect-sentiment model for online reviews. In *Proceedings of the* 27^{th} *AAAI Conference on Artificial Intelligence (AAAI)*, pages 804–812, Bellevue, 2013.

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the* 41^{st} *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 423–430, Sapporo, Japan, 2003.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the* 18^{th} *International Conference on Machine Learning (ICML)*, pages 282–289, San Francisco, 2001.

Angeliki Lazaridou, Ivan Titov, and Caroline Sporleder. A Bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *Proceedings of the* 51^{st} *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1630–1639, Sofia, Bulgaria, 2013.

Chenghua Lin, Yulan He, Richard Everson, and Stefan M. Rüger. Weakly supervised joint sentiment-topic detection from text. *IEEE Transactions on Knowledge and Data Engineering*, 24(6):1134–1145, June 2012.

Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, San Rafael, CA, 2012.

Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the* 14^{th} *international conference on World Wide Web (WWW)*, pages 342–351, Chiba, Japan, 2005.

Jun S. Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966, September 1994.

Yue Lu, Huizhong Duan, Hongning Wang, and ChengXiang Zhai. Exploiting structured ontology to organize scattered online opinions. In *Proceedings of the* 23^{rd} *International Conference on Computational Linguistics (COLING)*, pages 734–742, Beijing, 2010.

William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, January 1988.

Ryan T. McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeffrey C. Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the* 45^{th} *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 432–439, Prague, 2007.

Yelena Mejova and Padmini Srinivasan. Exploring feature definition and selection for sentiment classifiers. In *Proceedings of the* 5^{th} *International Conference on Weblogs and Social Media (ICWSM)*, Barcelona, 2011.

Thomas Corwin Mendenhall. The characteristic curves of composition. *Science*, 9(214):237–246, March 1887.

George A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38 (11):39–41, November 1995.

David M. Mimno, Hanna M. Wallach, Edmund M. Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the* 16^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 262–272, Edinburgh, 2011.

Frederick Mosteller and David L. Wallace. Inference in an authorship problem. *Journal of the American Statistical Association*, 58(302):275–309, June 1963.

Subhabrata Mukherjee, Gaurab Basu, and Sachindra Joshi. Joint author sentiment topic model. In *Proceedings of the* 14^{th} *International Conference on Data Mining (SDM)*, pages 370–378, Philadelphia, 2014.

Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. Dependency tree-based sentiment classification using CRFs with hidden variables. In *Proceedings of the* 11^{th} *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 786–794, Los Angeles, 2010.

Jamal A. Nasir, Nico Görnitz, and Ulf Brefeld. An off-the-shelf approach to authorship attribution. In *Proceedings of the* 25^{th} *International Conference on Computational Linguistics (COLING)*, pages 895–904, Dublin, 2014.

Vincent Ng, Sajib Dasgupta, and S. M. Niaz Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the* 21^{st} *International Conference on Computational Linguistics and* 44^{th} *Annual Meeting of the Association for Computational Linguistics (COLING/ACL)*, pages 611–618, Sydney, 2006.

NYC. About New York hotels. http://www.nyc.com/visitor_guide/about_new_york_hotels.703528/editorial_review.aspx, 2014. Accessed: 10/14/2014.

Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the* 42^{nd} *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 271–278, Barcelona, 2004.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the* 7^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, Philadelphia, 2002.

Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the Human Language Technology Conference and the* 11^{th} *Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, 2005.

Martin F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. Opinion word expansion and target extraction through double propagation. *Computer Linguistics*, 37(1):9–27, March 2011.

Lawrence R. Rabiner. Readings in speech recognition. chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pages 267–296. Morgan Kaufmann, San Francisco, 1990.

Delip Rao and Deepak Ravichandran. Semi-supervised polarity lexicon induction. In *Proceedings of the* 12^{th} *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 675–682, Athens, 2009.

Vassiliki Rentoumi, George A. Vouros, Vangelis Karkaletsis, and Amalia Moser. Investigating metaphorical language in sentiment analysis: A sense-to-sentiment perspective. *ACM Transactions on Speech and Language Processing (TSLP)*, (3):6:1–6:31, November 2012.

Michal Rosen-Zvi, Thomas L. Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the* 20^{th} *Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 487–494, Banff, Canada, 2004.

Sunita Sarawagi and William W. Cohen. Semi-markov conditional random fields for information extraction. In *Proceedings of Neural Information Processing Systems (NIPS)*, Vancouver, 2004.

Jacques Savoy. Authorship attribution based on a probabilistic topic model. *Information Processing Management*, 49(1):341–354, January 2013.

Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and José San Pedro. How useful are your comments?: analyzing and predicting youtube comments and comment ratings. In *Proceedings of the* 19^{th} *International Conference on World Wide Web (WWW)*, pages 891–900, Raleigh, 2010.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the* 16^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–161, Edinburgh, 2011.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the* 18^{th} *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642, Seattle, 2013.

Ashok Srivastava and Mehran Sahami, editors. *Text Mining: Classification, Clustering, and Applications*. Data Mining and Knowledge Discovery Series. Chapman & Hall/CRC, London, 2009.

Statista. A ranking of the most expensive cities in the US. http://www.statista.com/statistics/214585/most-expensive-cities-in-the-us-ordered-by-hotel-prices-2010/, 2014. Accessed: 10/14/2014.

Amber Stubbs. MAE and MAI: Lightweight annotation and adjudication tools. In *Proceedings of the* 5^{th} *Linguistic Annotation Workshop*, pages 129–133, Portland, 2011.

Swati Tata and Barbara Di Eugenio. Generating fine-grained reviews of songs from album reviews. In *Proceedings of the* 48^{th} *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1376–1385, Uppsala, Sweden, 2010.

Yla R. Tausczik and James W. Pennebaker. The psychological meaning of words: LIWC and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, March 2010.

Ivan Titov and Ryan T. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the* 46^{th} *Annual Meeting on Association for Computational Linguistics (ACL)*, pages 308–316, Columbus, Ohio, 2008a.

Ivan Titov and Ryan T. McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the* 17^{th} *International Conference on World Wide Web (WWW)*, pages 308–316, Beijing, 2008b.

TripAdvisor. A review sentence from TripAdvisor. http://www.tripadvisor.com/ShowUserReviews-g60763-d93589-r10006597, 2007. Accessed: 10/14/2014.

TripAdvisor. A review sentence from TripAdvisor. http://www.tripadvisor.com/ShowUserReviews-g295419-d514516-r48015315, 2009. Accessed: 10/14/2014.

TripAdvisor. A review sentence from TripAdvisor. http://www.tripadvisor.com/ShowUserReviews-g189415-d491123-r58095543, 2010. Accessed: 10/14/2014.

TripAdvisor. A review sentence from TripAdvisor. http://www.tripadvisor.com/ShowUserReviews-g32655-d81765-r100000013, 2011. Accessed: 10/14/2014.

TripAdvisor. A review sentence from TripAdvisor. http://www.tripadvisor.com/ShowUserReviews-g34372-d615165-r229142999, 2014a. Accessed: 10/14/2014.

TripAdvisor. A review sentence from TripAdvisor. http://www.tripadvisor.com/ShowUserReviews-g60873-d240673-r229021630, 2014b. Accessed: 10/14/2014.

TripAdvisor. TripAdvisor fact sheet. http://www.tripadvisor.com/PressCenter-c4-Fact_Sheet.html, 2014c. Accessed: 10/14/2014.

Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty. In *Proceedings of the* 47^{th} *Annual Meeting of the Association for Computational Linguistics and the* 4^{th} *International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL/IJCNLP)*, pages 477–485, Singapore, 2009.

Andranik Tumasjan, Timm Oliver Sprenger, Philipp G. Sandner, and Isabell M. Welpe. Predicting elections with Twitter: What 140 characters reveal about political sentiment. In *Proceedings of the* 4^{th} *International Conference on Weblogs and Social Media (ICWSM)*, Washington, DC, 2010.

Peter D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the* 40^{th} *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 417–424, Philadelphia, 2002.

Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346, October 2003.

Alessandro Valitutti, Carlo Strapparava, and Oliviero Stock. Developing affective lexical resources. *PsychNology Journal*, 2(1):61–83, November 2004.

Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceeding of* 23^{rd} *International Conference on Machine Learning (ICML)*, pages 977–984, Pittsburgh, 2006.

Janyce Wiebe. Learning subjective adjectives from corpora. In *Proceedings of the* 17^{th} *National Conference on Artificial Intelligence (AAAI)*, pages 735–740, Austin, 2000.

Janyce Wiebe, Rebecca Bruce, and Thomas O'Hara. Development and use of a gold standard data set for subjectivity classifications ANN. In *Proceedings of the* 37^{th} *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 246–253, College Park, 1999.

Yelp. Yelp dataset challenge. https://www.yelp.com/datasetchallenge/, 2014. Accessed: 04/09/2014.

Guangchao Yuan, Pradeep K. Murukannaiah, Zhe Zhang, and Munindar P. Singh. Exploiting sentiment homophily for link prediction. In *Proceedings of the* 8^{th} *ACM Conference on Recommender Systems (RecSys)*, pages 17–24, Foster City, 2014.

Lei Zhang, Bing Liu, Suk Hwan Lim, and Eamonn O'Brien-Strain. Extracting and ranking product features in opinion documents. In *Proceedings of the* 23^{rd} *International Conference on Computational Linguistics (COLING)*, pages 1462–1470, Beijing, 2010.

Rong Zheng, Jiexun Li, Hsinchun Chen, and Zan Huang. A framework for authorship identification of online messages: Writing-style features and classification techniques. *Journal of the Association for Information Science and Technology*, 57(3):378–393, February 2006.

APPENDIX

Appendix A

Supplemental Texts of Arch

A.1 Model Inference Process

In this section, we describe the detailed derivation of model inference process of Arch. We start from the joint probability of the assignments of sentiments $\boldsymbol{s}$, aspects $\boldsymbol{t}$, and words $\boldsymbol{w}$ for documents written by author a_0.

$$p(\boldsymbol{s},\boldsymbol{t},\boldsymbol{w}|a_0,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) = p(\boldsymbol{w}|\boldsymbol{s},\boldsymbol{t},\boldsymbol{\alpha})p(\boldsymbol{t}|\boldsymbol{s},a_0,\boldsymbol{\gamma})p(\boldsymbol{s}|\boldsymbol{\beta}) \tag{A.1}$$

We integrate out Φ in the first term in Equation A.1 as follows.

$$\begin{aligned}
p(\boldsymbol{w}|\boldsymbol{s},\boldsymbol{t},\boldsymbol{\alpha}) &= \int p(\boldsymbol{w}|\boldsymbol{s},\boldsymbol{t},\Phi)p(\Phi|\boldsymbol{\alpha})\,\mathrm{d}\Phi \\
&= \int \prod_{s=1}^{S}\prod_{t=1}^{T}\prod_{w=1}^{W} \varphi_{s,t,w}^{n_{s,t}^{w}} \prod_{s=1}^{S}\prod_{t=1}^{T}\frac{1}{\Delta(\boldsymbol{\alpha})}\prod_{w=1}^{W}\varphi_{s,t,w}^{\alpha_w-1}\,\mathrm{d}\boldsymbol{\varphi_{s,t}} \\
&= \prod_{s=1}^{S}\prod_{t=1}^{T}\frac{1}{\Delta(\boldsymbol{\alpha})}\int\prod_{w=1}^{W}\varphi_{s,t,w}^{n_{s,t}^{w}+\alpha_w-1}\,\mathrm{d}\boldsymbol{\varphi_{s,t}} \\
&= \prod_{s=1}^{S}\prod_{t=1}^{T}\frac{\Delta(\boldsymbol{n_{s,t}}+\boldsymbol{\alpha})}{\Delta(\boldsymbol{\alpha})} \\
&= \left(\frac{\Gamma(\sum_{w=1}^{W}\alpha_w)}{\prod_{w=1}^{W}\Gamma(\alpha_w)}\right)^{S*T} \times \prod_{s=1}^{S}\prod_{t=1}^{T}\frac{\prod_{w=1}^{W}\Gamma(n_{s,t}^{w}+\alpha_w)}{\Gamma\left[\sum_{w=1}^{W}(n_{s,t}^{w}+\alpha_w)\right]}
\end{aligned} \tag{A.2}$$

where T is the number of aspects; S is the number of sentiments; W is the size of the vocabulary; $\Phi = \{\boldsymbol{\varphi_i}\}_{i=1}^{T\times S}$; $\boldsymbol{n_{s,t}} = \{n_{s,t}^{w}\}_{w=1}^{W}$; $n_{t,s}^{w}$ counts the number of times that words

w being assigned with sentiment s and aspect t; $\Gamma(\cdot)$ is the gamma function; and function $\Delta(\boldsymbol{x}) = \frac{\prod_{i=1}^{dim\boldsymbol{x}} \Gamma(x_i)}{\Gamma(\sum_{i=1}^{dim\boldsymbol{x}} x_i)}$. .

Next, we integrate out Ψ_{a_0} in the second term in Equation A.1 as follows.

$$
\begin{aligned}
p(\boldsymbol{t}|\boldsymbol{s}, \boldsymbol{\gamma}, a_0) &= \int p(\boldsymbol{t}|\boldsymbol{s}, \Psi_{a_0}, a_0) p(\Psi_{a_0}|\boldsymbol{\gamma}) \, \mathrm{d}\Psi_{a_0} \\
&= \int \prod_{s=1}^{S} \prod_{t=1}^{T} \psi_{s,t}^{n_{s,a_0}^t} \prod_{s=1}^{S} \frac{1}{\Delta(\boldsymbol{\gamma})} \prod_{t=1}^{T} \psi_{s,t}^{\gamma_t - 1} \, \mathrm{d}\boldsymbol{\psi_s} \\
&= \prod_{s=1}^{S} \frac{1}{\Delta(\boldsymbol{\gamma})} \int \prod_{t=1}^{T} \psi_{s,t}^{n_{s,a_0}^t + \gamma_t - 1} \, \mathrm{d}\boldsymbol{\psi_s} \\
&= \prod_{s=1}^{S} \frac{\Delta(\boldsymbol{n_{s,a_0}} + \boldsymbol{\gamma})}{\Delta(\boldsymbol{\gamma})} \\
&= \left(\frac{\Gamma(\sum_{t=1}^{T} \gamma_t)}{\prod_{t=1}^{T} \Gamma(\gamma_t)} \right)^S \times \prod_{s=1}^{S} \frac{\prod_{t=1}^{T} \Gamma(n_{s,a_0}^t + \gamma_t)}{\Gamma\big[\sum_{t=1}^{T} (n_{s,a_0}^t + \gamma_t)\big]}
\end{aligned}
\tag{A.3}
$$

where $\Psi_{a_0} = \{\boldsymbol{\psi_i}\}_{i=1}^{S}$; $\boldsymbol{n_{s,a_0}} = \{n_{s,a_0}^t\}_{t=1}^{T}$; and n_{s,a_0}^t counts the number of segments in author a_0's reviews being associated with sentiment s and aspect t.

For the third term in Equation A.1, we integrate out Θ as follows.

$$
\begin{aligned}
p(\boldsymbol{s}|\boldsymbol{\beta}) &= \int p(\boldsymbol{s}|\Theta) p(\Theta|\boldsymbol{\beta}) \, \mathrm{d}\Theta \\
&= \int \prod_{d=1}^{D} \prod_{s=1}^{S} \pi_{d,s}^{n_d^s} \prod_{d=1}^{D} \frac{1}{\Delta(\boldsymbol{\beta})} \prod_{s=1}^{S} \pi_{d,s}^{\beta_s - 1} \, \mathrm{d}\boldsymbol{\pi_d} \\
&= \prod_{d=1}^{D} \frac{1}{\Delta(\boldsymbol{\beta})} \int \prod_{s=1}^{S} \pi_{d,s}^{n_d^s + \beta_s - 1} \, \mathrm{d}\boldsymbol{\pi_d} \\
&= \prod_{d=1}^{D} \frac{\Delta(\boldsymbol{n_d} + \boldsymbol{\beta})}{\Delta(\boldsymbol{\beta})} \\
&= \left(\frac{\Gamma(\sum_{s=1}^{S} \beta_s)}{\prod_{s=1}^{S} \Gamma(\beta_s)} \right)^D \times \prod_{d=1}^{D} \frac{\prod_{s=1}^{S} \Gamma(n_d^s + \beta_s)}{\Gamma\big[\sum_{s=1}^{S} (n_d^s + \beta_s)\big]}
\end{aligned}
\tag{A.4}
$$

where D is the number of reviews; $\Theta = \{\boldsymbol{\pi_i}\}_{i=1}^{D}$; $\boldsymbol{n_d} = \{n_d^s\}_{s=1}^{S}$; and n_d^s counts the number of times that a segment from review d being associated with sentiment s.

Arch contains five latent variables including sentiment assignments $\boldsymbol{s}$, aspect assignments $\boldsymbol{t}$, word distribution for sentiment-aspect pairs Φ, aspect distribution under sentiments for authors

Ψ, and sentiment distribution for reviews Θ. The goal of inference is to learn the following posterior distribution given the observed data.

$$p(\boldsymbol{s}, \boldsymbol{t}, \Phi, \Psi, \Theta | \boldsymbol{w}, a_0, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \tag{A.5}$$

Exact inference for this distribution is intractable. However, we can exploit approximate inference techniques, such as Gibbs sampling. Gibbs sampling is an example of Markov Chain Monte Carlo (MCMC) methods [Gilks et al., 1996]. To approximate a distribution, it constructs a Markov chain by sequentially sampling variables conditioned on all other variables. The stationary distribution of this Markov chain is an approximation of the distribution. Rather than building a Gibbs Sampler for each of five latent variables, we adopt the collapsed Gibbs sampling algorithm [Liu, 1994] in which we can simply sample $\boldsymbol{s}$ and $\boldsymbol{t}$ by integrating out Φ, Ψ, and Θ. Consequently, we derive the following full conditional distribution (hyperparameters are omitted) used as the update equation in the sampling process.

$$p(s_i = s, t_i = t | \boldsymbol{s_{-i}}, \boldsymbol{t_{-i}}, \boldsymbol{w}, a_0) \tag{A.6}$$

where an index $-i$ means that we exclude segment i from the count. By applying Bayes' rule, we transform Equation A.6 as follows.

$$\begin{aligned} p(s_i = s, t_i = t | \boldsymbol{s_{-i}}, \boldsymbol{t_{-i}}, \boldsymbol{w}, a_0) &= \frac{p(\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{w} | a_0)}{p(\boldsymbol{s_{-i}}, \boldsymbol{t_{-i}}, \boldsymbol{w} | a_0)} \\ &= \frac{p(\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{w} | a_0)}{p(\boldsymbol{w_{-i}} | \boldsymbol{s_{-i}}, \boldsymbol{t_{-i}}, a_0) p(\boldsymbol{s_{-i}}, \boldsymbol{t_{-i}} | a_0) p(w_i)} \\ &\propto \frac{p(\boldsymbol{s}, \boldsymbol{t}, \boldsymbol{w} | a_0)}{p(\boldsymbol{s_{-i}}, \boldsymbol{t_{-i}}, \boldsymbol{w_{-i}} | a_0)} \\ &\propto \frac{\Delta(\boldsymbol{n_s} + \boldsymbol{\beta})}{\Delta(\boldsymbol{n_{s,-i}} + \boldsymbol{\beta})} \cdot \frac{\Delta(\boldsymbol{n_t} + \boldsymbol{\gamma})}{\Delta(\boldsymbol{n_{t,-i}} + \boldsymbol{\gamma})} \cdot \frac{\Delta(\boldsymbol{n_w} + \boldsymbol{\alpha})}{\Delta(\boldsymbol{n_{w,-i}} + \boldsymbol{\alpha})} \end{aligned} \tag{A.7}$$

and

$$\begin{aligned} \frac{\Delta(\boldsymbol{n_s} + \boldsymbol{\beta})}{\Delta(\boldsymbol{n_{s,-i}} + \boldsymbol{\beta})} &\propto \frac{\Gamma(n_d^s + \beta_s) \Gamma\big[\sum_{s=1}^{S} (n_{d,-i}^s + \beta_s)\big]}{\Gamma(n_{d,-i}^s + \beta_s) \Gamma\big[\sum_{s=1}^{S} (n_d^s + \beta_s)\big]} \\ &\propto \frac{n_{d,-i}^s + \beta_s}{\sum_{s=1}^{S} (n_d^s + \beta_s)} \end{aligned} \tag{A.8}$$

$$\begin{aligned}\frac{\Delta(\boldsymbol{n_t}+\boldsymbol{\gamma})}{\Delta(\boldsymbol{n_{t,-i}}+\boldsymbol{\gamma})} &\propto \frac{\Gamma(n^t_{s,a_0}+\gamma_t)\Gamma\big[\sum_{t=1}^{T}(n^t_{s,a_0,-i}+\gamma_t)\big]}{\Gamma(n^t_{s,a_0,-i}+\gamma_t)\Gamma\big[\sum_{t=1}^{T}(n^t_{s,a_0}+\gamma_t)\big]} \\ &\propto \frac{n^t_{s,a_0,-i}+\gamma_t}{\sum_{t=1}^{T}(n^t_{s,a_0}+\gamma_t)}\end{aligned} \tag{A.9}$$

$$\begin{aligned}\frac{\Delta(\boldsymbol{n_w}+\boldsymbol{\alpha})}{\Delta(\boldsymbol{n_{w,-i}}+\boldsymbol{\alpha})} &\propto \frac{\prod_{v_{i,j}}\Gamma(n^{v_{i,j}}_{s,t}+\alpha_{v_{i,j}})\Gamma\big[\sum_{w=1}^{W}(n^{v_{i,j}}_{s,t,-i}+\alpha_{v_{i,j}})\big]}{\prod_{v_{i,j}}\Gamma(n^{v_{i,j}}_{s,t,-i}+\alpha_{v_{i,j}})\Gamma\big[\sum_{w=1}^{W}(n^{w}_{s,t}+\alpha_{w})\big]} \\ &\propto \frac{\prod_{v_{i,j}}\prod_{c=0}^{C_{v_{i,j}}-1}(n^{v_{i,j}}_{t,s,-i}+\alpha_{v_{i,j}}+c)}{\prod_{c=0}^{C_i-1}(n_{t,s,-i}+\sum_{w=1}^{W}\alpha_w+c)}\end{aligned} \tag{A.10}$$

Therefore, we have

$$\begin{aligned}p(s_i=s,t_i=t|\boldsymbol{s_{-i}},\boldsymbol{t_{-i}},\boldsymbol{w},a_0) &= A.8\times A.9\times A.10 \\ &\propto \frac{n^s_{d,-i}+\beta_s}{\sum_{s=1}^{S}(n^s_d+\beta_s)}\times\frac{n^t_{s,a_0,-i}+\gamma_t}{\sum_{t=1}^{T}(n^t_{s,a_0}+\gamma_t)} \\ &\times \frac{\prod_{v_{i,j}}\prod_{c=0}^{C_{v_{i,j}}-1}(n^{v_{i,j}}_{t,s,-i}+\alpha_{v_{i,j}}+c)}{\prod_{c=0}^{C_i-1}(n_{t,s,-i}+\sum_{w=1}^{W}\alpha_w+c)}\end{aligned} \tag{A.11}$$

where n^t_{s,a_0} counts segments from reviews written by author a_0 being associated with sentiment s and aspect t; n^s_d counts segments from review d associated with sentiment s; $C_{v_{i,j}}$ counts words $v_{i,j}$ appearing in segment i; C_i counts words in segment i; $n^{v_{i,j}}_{t,s}$ counts words $v_{i,j}$ assigned with sentiment s and aspect t; and $n_{t,s}$ counts words assigned sentiment s and aspect t in all reviews.

Next, we derive Φ, Ψ, and Θ using the sentiment and aspect assignments $\boldsymbol{M}=\{\boldsymbol{s},\boldsymbol{t}\}$ obtained through the sampling process.

We compute the word distribution of sentiment s and aspect t as follows.

$$
\begin{aligned}
p(\boldsymbol{\varphi_{s,t}}|\boldsymbol{M},\boldsymbol{\alpha}) &= \frac{\prod_{n=1}^{N_m} p(w_n|\boldsymbol{\varphi_{s,t}})p(\boldsymbol{\varphi_{s,t}}|\boldsymbol{\alpha})}{\int\prod_{n=1}^{N_m} p(w_n|\boldsymbol{\varphi_{s,t}})p(\boldsymbol{\varphi_{s,t}}|\boldsymbol{\alpha})\mathrm{d}\boldsymbol{\varphi_{s,t}}} \\
&= \frac{1}{Z}\prod_{w=1}^{W} p(w|\boldsymbol{\varphi_{s,t}})^{n_{s,t}^{w}}\frac{1}{\Delta(\boldsymbol{\alpha})}p(w|\boldsymbol{\varphi_{s,t}})^{\alpha_w-1} \\
&= \frac{1}{\Delta(\boldsymbol{n_{s,t}}+\boldsymbol{\alpha})}\prod_{w=1}^{W} p(w|\boldsymbol{\varphi_{s,t}})^{n_{s,a_0}^{t}+\alpha_w-1} \\
&= Dir(\boldsymbol{\varphi_{s,t}}|\boldsymbol{n_{s,t}}+\boldsymbol{\alpha})
\end{aligned}
\tag{A.12}
$$

Given the above Dirichlet distribution, the approximate probability of word w being associated with sentiment s and aspect t is

$$
\varphi_{s,t,w} = \frac{n_{s,t}^{w}+\alpha_w}{n_{s,t}+\sum_{w=1}^{W}\alpha_w} \tag{A.13}
$$

We compute the aspect distribution of sentiment s for author a_0 as follows.

$$
\begin{aligned}
p(\boldsymbol{\psi_{s,a_0}}|\boldsymbol{M},\boldsymbol{\gamma}) &= \frac{\prod_{n=1}^{N_s} p(t_{n,a_0}|\boldsymbol{\psi_{s,a_0}})p(\boldsymbol{\psi_{s,a_0}}|\boldsymbol{\gamma})}{\int\prod_{n=1}^{N_s} p(t_{n,a_0}|\boldsymbol{\psi_{s,a_0}})p(\boldsymbol{\psi_{s,a_0}}|\boldsymbol{\gamma})\mathrm{d}\boldsymbol{\psi_{s,a_0}}} \\
&= \frac{1}{Z}\prod_{t=1}^{T} p(t|\boldsymbol{\psi_{s,a_0}})^{n_{s,a_0}^{t}}\frac{1}{\Delta(\boldsymbol{\gamma})}p(t|\boldsymbol{\psi_{s,a_0}})^{\gamma_t-1} \\
&= \frac{1}{\Delta(\boldsymbol{n_{s,a_0}}+\boldsymbol{\gamma})}\prod_{t=1}^{T} p(t|\boldsymbol{\psi_{s,a_0}})^{n_{s,a_0}^{t}+\gamma_t-1} \\
&= Dir(\boldsymbol{\psi_{s,a_0}}|\boldsymbol{n_{s,a_0}}+\boldsymbol{\gamma})
\end{aligned}
\tag{A.14}
$$

Given the above Dirichlet distribution, for author a_0, the approximate probability of a segment being associated with sentiment s and aspect t is

$$
\psi_{s,t,a_0} = \frac{n_{s,a_0}^{t}+\gamma_t}{n_{s,a_0}+\sum_{t=1}^{T}\gamma_t} \tag{A.15}
$$

Finally, We compute the sentiment distribution of document d as follows.

$$
\begin{aligned}
p(\boldsymbol{\pi_d}|\boldsymbol{M},\boldsymbol{\beta}) &= \frac{\prod_{n=1}^{N_d} p(s_{d,n}|\boldsymbol{\pi_d})p(\boldsymbol{\pi_d}|\boldsymbol{\beta})}{\int\prod_{n=1}^{N_d} p(s_{d,n}|\boldsymbol{\pi_d})p(\boldsymbol{\pi_d}|\boldsymbol{\beta})\mathrm{d}\boldsymbol{\pi_d}} \\
&= \frac{1}{Z}\prod_{s=1}^{S} p(s|\boldsymbol{\pi_d})^{n_d^s}\frac{1}{\Delta(\boldsymbol{\beta})}p(s|\boldsymbol{\pi_d})^{\beta_s-1} \\
&= \frac{1}{\Delta(\boldsymbol{n_d}+\boldsymbol{\beta})}\prod_{s=1}^{S} p(s|\boldsymbol{\pi_d})^{n_d^s+\beta_s-1} \\
&= Dir(\boldsymbol{\pi_d}|\boldsymbol{n_d}+\boldsymbol{\beta})
\end{aligned}
\tag{A.16}
$$

Given the above Dirichlet distribution, the approximate probability of a segment in document d being assigned with sentiment s is

$$
\pi_{d,s} = \frac{n_d^s+\beta_s}{n_d+\sum_{s=1}^{S}\beta_s} \tag{A.17}
$$

A.2 Additional Experimental Results

In this section, we show some additional experimental results of Arch. Table A.1 and A.2 show p-values of topic coherence comparison with different numbers of aspects on hotel and restaurant reviews, respectively. Table A.3 shows the similarities among the seven cities. Table A.4 shows the similarities among the five trip types. Table A.5 shows the similarities among four authors.

Table A.1: Statistics of topic coherence comparison on hotel reviews with different numbers of aspects

	10	20	30	40	50	60
Arch>ASUM-S	3.10E-3	4.94E-4	4.08E-4	3.50E-3	1.87E-2	3.83E-2
Arch>ASUM	9.62E-6	4.45E-7	1.35E-5	9.98E-7	1.78E-4	3.22E-5
Arch>LDA	5.87E-7	5.48E-8	6.80E-10	1.20E-10	1.78E-11	1.05E-12
ASUM-S>ASUM	4.19E-6	1.65E-5	8.04E-5	1.86E-6	4.43E-4	6.15E-6
ASUM>LDA	3.59E-2	4.48E-6	7.25E-8	3.39E-9	4.07E-8	1.04E-9

Table A.2: Statistics of topic coherence comparison on restaurant reviews with different numbers of aspects; (·) indicates no significance with p-value greater than 0.1

	10	20	30	40	50	60
Arch>ASUM-S	2.19E-4	4.98E-4	6.30E-3	1.36E-2	(5.40E-1)	(6.82E-1)
Arch>ASUM	1.20E-3	8.42E-5	8.07E-4	9.20E-4	1.66E-2	1.10E-2
Arch>LDA	2.36E-6	6.54E-9	3.32E-11	1.32E-10	9.12E-10	9.49E-10
ASUM-S>ASUM	6.35E-2	5.40E-3	1.04E-2	2.27E-2	1.58E-2	5.20E-3
ASUM>LDA	6.97E-5	1.53E-8	5.16E-10	9.70E-11	5.78E-11	1.81E-10

Table A.3: Similarity matrix among seven cities.

	Boston	Chicago	Las Vegas	Los Angeles	Miami	New York	Orlando
Boston	—	0.075	0.443	0.150	0.22	0.132	0.374
Chicago	0.075	—	0.439	0.178	0.238	0.112	0.371
Las Vegas	0.443	0.439	—	0.428	0.405	0.469	0.407
Los Angeles	0.150	0.178	0.428	—	0.142	0.225	0.340
Miami	0.220	0.238	0.405	0.142	—	0.286	0.338
New York	0.132	0.112	0.469	0.225	0.286	—	0.405
Orlando	0.374	0.371	0.407	0.340	0.338	0.405	—

Table A.4: Similarity matrix among five trip types.

	Business	Couple	Family	Friend	Solo
Business	—	0.184	0.194	0.166	0.159
Couple	0.184	—	0.096	0.088	0.111
Family	0.194	0.096	—	0.107	0.146
Friend	0.166	0.088	0.107	—	0.114
Solo	0.159	0.111	0.146	0.114	—

Table A.5: Similarity matrix among four authors.

	Author 1	Author 2	Author 3	Author 4
Author 1	—	0.597	0.631	0.674
Author 2	0.597	—	0.549	0.555
Author 3	0.631	0.549	—	0.339
Author 4	0.674	0.555	0.339	—

Figure A.1 represents the preference profiles of Boston, Chicago, Orlando, and Friend. Figure A.2 represents the preference profiles of four authors.

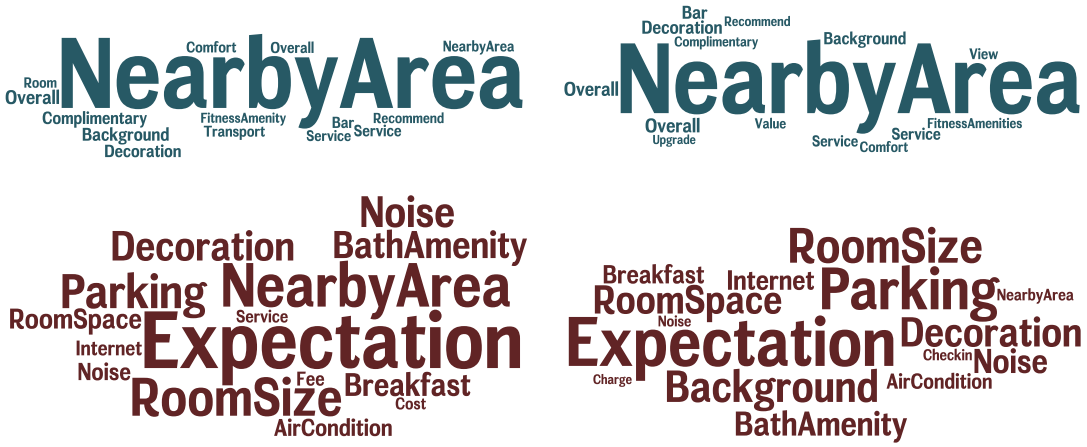
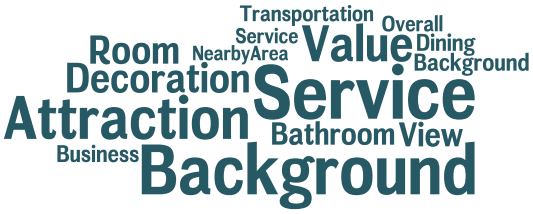
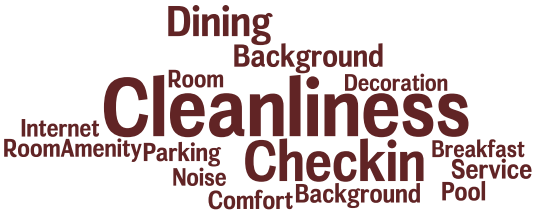


Figure A.1: An aspect-cloud visualization of Boston, Chicago, Orlando, and Friend (positive aspects above; negative aspects below).



Author 1



Author 2

Author 3

Author 4

Figure A.2: An aspect-cloud visualization of four authors (positive aspects above; negative aspects below).