Target-Oriented Content and Sentiment Analysis

by

Shuai Wang
M.S., Chinese University of Hong Kong, 2013
B.E., Guangdong University of Foreign Studies, 2012

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2019

Chicago, Illinois

Defense Committee:
Bing Liu, Chair and Advisor
Philip S. Yu
Cornelia Caragea
Sherry L. Emery, University of Chicago
Pradipto Das, Rakuten Inc.

This thesis is dedicated to my wife Xiaohui (Jess) Lin.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my advisor Prof. Bing Liu. It was my great honor to be his Ph.D. student, through which I learned not only how to conduct research, but more importantly, how to think and solve problems in a fundamental manner. I sincerely appreciate his supervision, encouragement, patience, support, and trust.

I also want to thank Prof. Philip S. Yu, Prof. Cornelia Caragea, Prof. Sherry Emery and Dr. Pradipto Das for taking their valuable times to serve as my committee. Their comments and feedbacks make important contributions to the accomplishment of this thesis.

I also want to thank my colleagues: Zhiyuan Chen, Huayi Li, Geli Fei, Yueshen Xu, Nianzu Ma, Lei Shu, Xu Hu, Sahisnu Mazumder, Jiahua Chen, Guangyi Lv, Hao Wang, Xue Zhao, Chenwei Zhang, Zhan Shi, Mao Li, Hongwei Jin, Yingyi Ma, and so on, who have helped my Ph.D. studies in various ways.

Last but not least, I owe many thanks to my family, including my parents, sister, brother-in-law, grandparents and all other relatives, without whom my Ph.D. studies will not even be possible. My greatest gratitude goes to my wife Xiaohui, who has been taking perfect care of me. Her unconditional love, support, and trust motivate me to work hard every day. Her company is always my strongest weapon to fight against all difficulties and toughnesses.

SW

CONTRIBUTION OF AUTHORS

Chapter 2 presents a published manuscript (Wang et al., 2016a) for which I was the primary author. Zhiyuan Chen, Geli Fei, my advisor Professor Bing Liu, and Sherry Emery contributed to the discussions about the preliminary ideas and the assistance in revising the manuscript. Sherry Emery also provided the social media datasets for the experiments.

Chapter 3 presents a published manuscript (Wang et al., 2018b) for which I was the primary author. Sahisnu Mazumder, Mianwei Zhou, my advisor Professor Bing Liu, and Yi Chang contributed to the discussions about the preliminary ideas and the assistance in revising the manuscript.

Chapter 4 presents a published manuscript (Wang et al., 2018a) for which I was the primary author. Guangyi Lv, Sahisnu Mazumder, Geli Fei, and my advisor Professor Bing Liu contributed to the discussion about the preliminary ideas and the assistance in revising the manuscript. Guangyi Lv also assisted in implementing and running two baseline models.

Chapter 5 presents a published manuscript (Wang et al., 2016b) for which I was the primary author. Zhiyuan Chen and my advisor Professor Bing Liu contributed to the discussion about the preliminary ideas and the assistance in revising the manuscript.

TABLE OF CONTENTS

CHAPTER		PAGE
1	**INTRODUCTION**	1
1.1	Target-Oriented Sentiment Analysis	2
1.2	Target-Oriented Content Analysis	3
1.3	Lifelong Machine Learning for Target-Oriented Analysis	4
2	**TARGETED TOPIC MODELING FOR TARGET-FOCUSED ANALYSIS**	6
2.1	Introduction	6
2.2	Proposed Method	10
2.2.1	Generative Process	13
2.2.2	Targeted Modeling with Sparsity	15
2.2.2.1	Biased Sparsity	16
2.2.2.2	Targeted Modeling	17
2.2.2.3	Spike-and-Slab Prior	19
2.2.3	Inference	20
2.3	Experiment	22
2.3.1	Experimental Setup	22
2.3.2	Baseline Models for Comparison	24
2.3.3	Quantitative Evaluation	25
2.3.3.1	Precision in Setting One	27
2.3.3.2	Precision in Setting Two	29
2.3.4	Qualitative Evaluation	31
2.3.4.1	Example One: E-Cigarette and Children	32
2.3.4.2	Example Two: Camera, Screen and Weight	34
2.4	Related Work	36
2.5	Summary	38
3	**TARGET-SENSITIVE MEMORY NETWORK FOR TARGET-BASED SENTIMENT CLASSIFICATION**	39
3.1	Introduction	39
3.2	Memory Network for ASC	42
3.3	Problem of the above Model for Target-Sensitive Sentiment	44
3.4	Proposed Approaches	47
3.5	Experiments	52
3.5.1	Candidate Models for Comparison	53
3.5.2	Evaluation Measure	54
3.5.3	Training Details	54

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

3.5.3.1 Result Analysis 55
3.5.4 Effect of TCS Interaction for Identifying Target-Sensitive Sentiment 58
3.6 Related Work 60
3.7 Summary 62

4 LIFELONG LEARNING MEMORY NETWORK FOR TARGET-BASED SENTIMENT CLASSIFICATION 63
4.1 Introduction 63
4.2 ASC Memory Network 69
4.3 Lifelong Learning Algorithm 71
4.4 lifelong learning memory network (L2MN) 77
4.5 Experiments 81
4.5.1 Candidate Models for Comparison 81
4.5.2 Experimental Setup 83
4.5.3 Result Analysis 86
4.5.4 Case Study 91
4.6 Related Work 93
4.7 Summary 95

5 LIFELONG ASPECT SENTIMENT TOPIC MODELING FOR MINING TARGET-SPECIFIC SENTIMENT 96
5.1 Introduction 96
5.2 JAST Model 102
5.3 LAST Model 107
5.3.1 LAST Learning Algorithm 107
5.3.2 Proposed Gibbs Sampler for LAST 110
5.3.2.1 Pólya Urn Model 110
5.3.2.2 Promotion Matrix Estimation 111
5.3.2.3 Inference 113
5.3.3 Discussion 114
5.4 Evaluation 115
5.4.1 Candidate Models for Comparison 115
5.4.2 Experiment Setup 115
5.4.3 Topic Coherence 117
5.4.4 Topic Quality Evaluation 119
5.4.4.1 Opinion Precision 119
5.4.4.2 Opinion Specificity 121
5.4.4.3 Aspect Precision 123
5.5 Related Work 124
5.6 Summary 129

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

6 CONCLUSIONS 130

APPENDICES 132

CITED LITERATURE 136

VITA 148

LIST OF TABLES

TABLE		PAGE
I	Definitions of notations	12
II	Five datasets, targeted aspects, and initial documents (tweets or review sentences)	23
III	Precisions of setting one. The last two rows are (a) the average scores of all targeted aspects of all topics and (b) the improvement achieved by TTM over other models respectively.	28
IV	Precisions of setting two. The last two rows are (a) the average scores of all targeted aspects of all topics and (b) the improvement achieved by TTM over other models respectively.	30
V	Topics of aspect *children* under E-Cig. Errors are italicized and marked in red.	32
VI	Topics of two aspects *screen* and *weight* under Camera. Errors are italicized and marked in red.	35
VII	Definition of notations	43
VIII	Statistics of datasets	53
IX	Results of all models on two datasets. Top three F1-Macro scores are marked in bold. The first nine models are 1-hop memory networks. The last nine models are 3-hop memory networks.	56
X	Results with Recurrent Attention	57
XI	Sample Records and Model Comparison between MN and TMN	58
XII	Statistics of the datasets in Gold Data.	84
XIII	Binary classification results on first three datasets of Gold Data.	86
XIV	Three-class classification results on first three datasets of Gold Data.	88
XV	Results on laptop.	89
XVI	Knowledge examples.	90
XVII	Definition of notations	103
XVIII	Opinion words for Battery and Shipping&Order aspects. Incorrect opinion words are italicized and marked in red. Non-specific opinion words are italicized and marked in blue.	121
XIX	Example aspect words for Battery and Shipping&Order. Errors are marked in red.	124

LIST OF FIGURES

FIGURE		PAGE
1	The graphical model of TTM	11
2	Sample data with aspects and aspect-sentiment labels.	64
3	Lifelong Learning Memory Network (L2MN).	77
4	Attention and sentiment logit on case 1.	91
5	Attention and sentiment logit on case 2.	92
6	The graphical model of JAST	102
7	Average topic coherence of each model over 50 domains.	117
8	Opinion evaluation - models in each figure from left to right are LAST, JAST and ASUM	120
9	Aspect precision - models in each figure from left to right are LAST, JAST and ASUM	123

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
LML	Lifelong Machine Learning
LDA	Latent Dirichlet Allocation
TTM	Targeted Topic Model
ASC	Aspect Sentiment Classification
MN	Memory Network
TMN	Target-sensitive Memory Network
L2MN	Lifelong Learning Memory Network
JAST	Joint Aspect-based Sentiment Topic model
LAST	Lifelong Aspect-based Sentiment Topic model

SUMMARY

Sentiment analysis aims to analyze people's opinions, sentiments, emotions, attitudes, and other related concepts, which has been an active research topic in the fields of natural language processing (NLP) and data mining. Content analysis is a broader concept or can be viewed as a more general task, which aims to discover various types of useful knowledge or information given a text corpus (in addition to sentiment), for example, to find out the topics under discussion with a given text corpus. With the rapid development of the Web, a huge amount of user-generated content is publicly accessible. Content analysis, as well as sentiment analysis, thus plays a more and more vital role in this big data era.

To comprehensively understand the content and sentiment, simply relying on coarse-grained or full analysis techniques is insufficient and of limited use. Instead, we need *target-oriented content and sentiment analysis* as a further step. For example, given a text corpus, it is more informative to understand what topics or aspects are discussed and what sentiments are carried in each topic or aspect.

A specific problem of target-oriented content and sentiment analysis is defined in a more fine-grained or focused setting. Here a target could be an object of interest, such as a topic, an entity, or an aspect (i.e., product feature). For example, given a set of online reviews about the product camera, a user may be particularly interested in the target aspect *screen*. More specifically, the user wants to have a focused analysis and figure out what specific topics are related to the target aspect *screen*, i.e., what people mostly care about the screen, like its resolution,

its picture quality, and whether the menu (displayed on the screen) is easy to use. This is a typical task of target-oriented content analysis, and we have proposed a novel topic model to address it (Wang et al., 2016a). Notice that in the literature (Liu, 2012), we often use the term aspect to generally represent broader concepts like entities, attributes, sub-components. We follow such manner and will use the term *aspect*, *target aspect* and *target* interchangeably in this thesis. In term of target-oriented sentiment analysis, the aspect sentiment classification (a.k.a., aspect-based sentiment classification) is a core task, which is to infer the sentiment conditioned on the target. We have proposed several alternative approaches to addressing it and making the sentiment prediction sensitive on different targets (Wang et al., 2018b).

While these two works can basically demonstrate the usefulness of encoding target information in content and sentiment analysis, they only consider the data in one domain independently, which somewhat limits their model performance. As discussed above, with the rapid development of the Web and the surge of social media, we now have a good opportunity to use massive data. However, the challenge is how to use them in an effective way, especially the unlabeled data. While the exploitation of (big) labeled data can intuitively help achieve better results, the exploration of (big) unlabeled data is more realistic and promising, because labeling big data is a costly, time-consuming, labor-intensive, and daunting task. In practice, it is also hard to scale up to multiple domains, where a domain generally means a specific type of product like camera, cellphone, and laptop. To tackle the issue of better model learning using big data, our solution is *lifelong machine learning* (LML) or *lifelong learning* for short.

SUMMARY (Continued)

Two of my Ph.D. studies will be illustrated as follows to show the effectiveness of using LML for target-oriented content and sentiment analysis. More specifically, one study is to address the aspect sentiment classification problem, by learning and incorporating target-specific attention and sentiment knowledge (Wang et al., 2018a). Another one is to holistically identify aspect terms and aspect-specific opinion terms, with the knowledge involved in the topic modeling process. Both tasks have successfully employed the big (unlabeled) data from multiple domains in a lifelong learning setting.

Based on this thesis, we believe: (1) Target-oriented content and sentiment analysis is useful and practical in many real-world applications; (2) Lifelong machine learning (LML) can play an important role in target-oriented content and sentiment; (3) Many and more different types of knowledge can be learned and used in the future development of target-oriented analysis.

CHAPTER 1

INTRODUCTION

With the rapid development of the Web and social media platforms, analyzing the content from massive amounts of data becomes more and more important. In regard to the Web content, a sea of information is user-generated, which contains opinions, sentiments, emotions, stands and so on, where the sentiment analysis and opinion mining also becomes an important task. To some extent, sentiment analysis can be viewed as a mini natural language processing (NLP) problem (Liu, 2012). That is, if we have a set of effective approaches to addressing its sub-problems like aspect extraction, aspect categorization, and sentiment classification, we will be very likely to extend them to more general NLP tasks like named entity recognition, topic summarization, and category classification. This motivates me to select the sentiment analysis, as well as content analysis (which is a broader concept), to be my research focus in my Ph.D. studies.

In this thesis, I focus on a more specific problem named target-oriented content and sentiment analysis. Here a target generally stands for an object of a user's interest, such as an entity, an aspect (i.e., a product feature), a topic and an event. The objective is to discover target-oriented information, such as topics, opinion words, and sentiment polarities that are particularly related to a specific target. Let us say there is an online review sentence "the screen is very clean but the battery life is too short". When the target is *screen* its sentiment should be positive, but when the target is *battery life*, its sentiment would be negative. This

motivating example shows the key idea of target-oriented analysis, which is to pinpoint target dependent information in a more focused setting. This is quite different from traditional coarse-grained or full analysis research. In this thesis, I will first introduce two important tasks, namely, target-oriented sentiment analysis, and target-oriented topic modeling. To address them, I proposed novel solutions and conducted experiments for their evaluation, which will be discussed in details in the following sections. They are two of the research works I have made during my Ph.D. studies.

My further works on these two tasks are to use the idea of lifelong machine learning (LML) for performance enhancement. The intuition is that, when a system or learner performs tasks continuously, we want it to utilize the knowledge obtained from the past to help future tasks. To achieve this goal, we proposed two lifelong machine learning models for content analysis and sentiment analysis respectively.

1.1 Target-Oriented Sentiment Analysis

Sentiment analysis aims to analyze people's opinions, sentiments, emotions, attitudes, and other related concepts, which has been an active research topic in the fields of natural language processing (NLP) and data mining. A target-oriented sentiment analysis, also known as target-based or aspect-based sentiment analysis, aims to figure out the sentiment particularly towards a given target. In this type of analysis, a core task is the target sentiment classification, which is to infer the sentiment polarity on the given target, namely, positive, neutral, or negative. To address this task, a state-of-the-art machine learning model is memory network.

Memory network, a type of neural model, is very suitable for the target sentiment classification with three reasons. First, it can learn a set of target representations and a set of context representations. Second, given a target and a sentence, its attention mechanism enables it to discover important sentiment context from the sentence, using the interaction between internally learned target representations and context representations. Third, a trained memory network is domain-specific, which can capture the in-domain sentiment expressions without any external resources like sentiment lexicons.

However, we observed that this model cannot address the target-sensitive information well due to some fundamental problems in its model design. For example, it fails to correctly predict "the price is high" is negative given the *price* as the target and "the screen resolution is high" is positive given the target is *screen resolution* at the same time. I will discuss the cause in depth and introduce our proposed solutions in Chapter 3.

1.2 Target-Oriented Content Analysis

Content analysis is a broader concept or can be viewed as a more general task of sentiment analysis, which aims to discover various types of useful knowledge or information given a text corpus (in addition to sentiment). Target-oriented content analysis aims to learn target-based information in a more focused fashion, which is tightly related to the given target.

My research work in this thread aims to address the problem of generating target-oriented topics. That is, given a collection of documents, our goal is to find the topics (discussed in the documents) that are specifically related to a given target. For example, given a set of tweets

(from Twitter) about e-cigarette, and if our target is kids, we could find target-focused topics like fears, regulations, and health (concerning kids smoking e-cigarettes).

This task is very important in analyzing real-world data, especially for health and social scientists who always need to conduct a focused analysis. It is also useful for a common user or manufacturer to have the target-specific analysis of some important product features. Before my work, existing solutions failed to handle this task because of their limited or unsuitable model capabilities. To address it, I designed a new model specifically for this task, utilizing the theory of probability and sparse modeling, which will be introduced in Chapter 2.

1.3 Lifelong Machine Learning for Target-Oriented Analysis

Lifelong Machine Learning (LML) (Chen and Liu, 2016), also called lifelong learning, is proposed as a machine learning paradigm to make machines learn as humans do. When meeting a new task, humans naturally use obtained experience or accumulated knowledge from the past tasks to help deal with it. We also become more knowledgeable and capable to perform better with more and more knowledge learned. LML mimics this human learning capability and applies it to computational models.

While the above introduced two works can basically demonstrate the usefulness of encoding target information in content and sentiment analysis, they only consider the data in one domain independently, which somewhat limits their model performance. As discussed above, living in this bid date era, we now have a good opportunity to use massive text corpora, especially the unlabeled ones. However, the challenge is how to use them in a effective way. While the exploitation of (big) labeled data can intuitively help achieve better results, the exploration

of (big) unlabeled data is more realistic and promising, because labeling big data is usually time-consuming and labor-intensive to obtain. In practice, it is also hard to scale up to a lrge number of domains. To tackle the issue of better model learning using big data, lifelong machine learning is a natural choice because it can use the learned knowledge from the past (big) data to help.

Two of my Ph.D. studies will be illustrated in Chapter 4 and Chapter 5 to demonstrate the effectiveness of using LML for target-oriented content and sentiment analysis. More specifically, Chapter 4 tackles the aspect sentiment classification problem in a lifelong learning setting, by learning and incorporating target-specific attention and sentiment knowledge. Chapter 5 holistically mines aspect terms and aspect-specific opinion terms, with the knowledge involved in the topic modeling process.

CHAPTER 2

TARGETED TOPIC MODELING FOR TARGET-FOCUSED ANALYSIS

(This chapter includes and expands on my paper previously published in *Shuai Wang, Zhiyuan Chen, Geli Fei, Bing Liu, and Sherry Emery. "Targeted Topic Modeling for Focused Analysis". In* ***KDD 2016****.*)

2.1 Introduction

In this chapter, we introduce the task of target-focused analysis, which is a typical case of target-oriented content analysis. To address it, we proposed a novel model named targeted topic model.

One of the important NLP and data mining tasks is to discover the topics discussed in a collection of text documents (or a corpus). Topic modeling is one of the main techniques used for this purpose. So far numerous topic models have been proposed in the literature, which may mine topics only or jointly mine topics and other types of useful information, for example, sentiment information (Jo and Oh, 2011; Brody and Elhadad, 2010).

However, existing models typically perform full analysis on the entire corpus to discover all topics. This is certainly useful, but it is inevitably coarse. In practice we found that the user almost always also wants to perform deeper and more focused analysis on some specific aspects of the data, which we refer to as *targets*, or *targeted aspects* in this paper. For example, given a set of tweets about *e-cigarette*, the user (or researcher) wants to gain insight into topics that

have been discussed about *children*. Here *children* is the targeted aspect. If a topic model can find topics such as *regulations* and *fears* about children smoking e-cigarette that are specifically related to this target, it will be very useful. Formally, the proposed targeted analysis problem is defined as follows (note that we use targeted analysis and focused analysis in this paper interchangeably).

Problem Definition: Given a corpus C of documents of a broad area/domain, discover related topics T of a user-interested aspect (called targeted aspect) represented with a set of keywords S provided by the user.

To solve this problem, a natural approach to start with is to use a regular full-analysis topic model such as LDA (Blei et al., 2003). Following the previous example, we can see that e-cigarette is a broad area (with corpus C provided) and *children* is the targeted aspect. Note that the target *children* is represented by some keywords S (e.g., S = {"children," "kids"}). After applying LDA on the corpus C to produce a set of topics Q, we find those topics in Q that contain some keywords from S in their top ranked words and study them in order to find the target-related topics T ($T \subseteq Q$). However, this approach is often unsatisfactory due to a few reasons (or *issues*).

1. The user does not know all the keywords that can represent a targeted aspect. In the above example, if the user specifies "children" as the only keyword and miss out other related keywords such as "young" and "minors", he may lose some important topics.

2. It may not find any topics for the user-interested aspect. Because there may be many other more prevalent or dominating topics in the data, the model may not find the related topics

of some less frequent aspects. For example, when our targeted aspect is *weight* represented by the keyword "weight" in camera reviews, the word "weight" has a relatively low occurrence frequency as people often mention this concept implicitly in the sentences like "the lens is so heavy" or "its battery is light". A full-analysis topic model such as LDA may not find any topic about *weight* at all.

3. Even if the keywords are frequent, one still may not find good coherent topics due to two reasons: (1) *Topic suppression*: Since the targeted aspect is only one of many aspects discussed in a broad area and a full-analysis topic model generates all topics for all aspects, the related topics of the targeted aspect may be suppressed. Many general words may be ranked at the top. For example, a topic about *children* (the targeted aspect) with topical words like "children", "kids", "young" is not informative. (2) *Word intrusion*: Words from other non-targeted aspects' topics may be *intruder* words appearing in the related topics of the targeted aspect, which makes the detection and understanding of the target-related topics difficult.

The cause of the above problems is related to some properties of topic modeling. First, useful information may not be easily detected under the condition of data sparsity or small data size. That is because classic topic models are unsupervised and governed by the phenomena called *higher order co-occurrence* (Heinrich, 2009). As a result, some informative but infrequent words may be ranked low or even can not be correctly grouped. Second, the existing models are not targeted towards any user interest. As identified in (Chang et al., 2009), the objective functions of topic models may not correlate well with human judgments and needs. For example, a given broad corpus often cover a large number of topics, the topics for the user-interested

aspect (i.e., target) may be mixed up with other non-target related topics and become incoherent. Although increasing the number of topics may help, the problem remains and other problems like information fragmentation may show up (the problems will be further discussed in Section 2.3.4). One may use knowledge-based models for targeted modeling (Mukherjee and Liu, 2012; Wang et al., 2016b), but these models only try to put words related to the user specified keywords in the same topic. They do not distill the topics related to target aspect represented by keywords as we do. They still suffer from the aforementioned issues.

Another intuitive approach to solving the proposed problem is to select a subset of documents from the corpus C that contains at least one keyword $s \in S$ (denoted by C') and apply LDA to the resulting documents in C'. Clearly, this approach has the problem as illustrated in *Issue* 1. It also has *Issue* 2 but manifested differently. For example, when the keyword set S is {"weight"} the number of document C' might be so small that a topic model is unable to produce many good topics. Moreover, since it discards many potentially relevant documents, it diminishes the quality of topics and also loses many potentially related topics.

Recently, topic modeling with sparsity has been proposed. Sparse topic models such as those in (Chen et al., 2012; Williamson et al., 2010; Lin et al., 2014) can (a) identify focused topics of a document, or (b) extract focused words of a topic. However, they are still full-analysis models and not for targeted modeling, and thus still suffer from the aforementioned issues. More discussions will be given in Section 2.4. Inspired by them, we employ the sparsity idea in designing our new model for targeted or focused analysis.

To address the proposed problems, we designed a new model called the *targeted topic model* (TTM). It is used for focused analysis as it can directly generate related topics of a given targeted aspect. The novelty of TTM is that it models using the entire corpus C while targeted at the user-specified aspect. This enables TTM to discover more related topics and also improve the topic quality because it can better exploit the information from other relevant documents in C that do not contain the given keywords in S.

In summary, this paper makes the following contributions:

1. It proposes the new problem of targeted topic modeling to discover only topics that are related to a user-specified aspect. To the best of our knowledge, no existing topic model can perform this task. Such targeted/focused analysis is important because not everyone is interested in everything in a corpus. When one is interested in a particular aspect, he/she often wants to perform deeper and focused analysis.

2. It proposes a new probabilistic topic model called *Targeted Topic Model* (TTM) that is able to perform the proposed focused analysis, which is also the first such model.

3. Our experimental results using five real-life datasets and a set of aspects show the effectiveness of the proposed model. It outperforms state-of-art baseline models markedly.

2.2 Proposed Method

As discussed in the introduction section, our problem statement is that given a corpus C of a broad area, our proposed model can generate topics of the targeted aspect specified by the user using a set of keywords S.

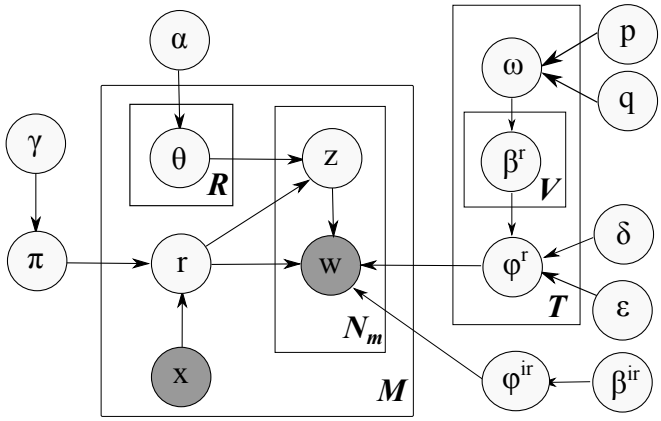


Figure 1: The graphical model of TTM

Since the proposed model needs to discover fine-grained topics (also called topics of the targeted aspect), we treat each sentence as a document in topic modeling like several other fine-grained models (Brody and Elhadad, 2010; Titov and McDonald, 2008a). Following previous work in (Titov and McDonald, 2008a; Brody and Elhadad, 2010; Zhao et al., 2010; Mukherjee and Liu, 2012; Jo and Oh, 2011), we also assume that each sentence focuses on only one aspect. Since we regard each sentence as a document in modeling, each document focuses on only one aspect. Although it might not always be correct, it holds up well in practice (Jo and Oh, 2011; Zhao et al., 2010) and generates good results (shown in Section 2.3). The graphical model of the proposed TTM is given in Figure 1. Since a document talks about one aspect, when the targeted aspect is specified, there can be two possible statuses for a document, that is, relevant or irrelevant to the targeted aspect. The status variable is denoted by r.

T	the number of topics
M	the number of documents
N_m	the number of words in document m
V	the number of words (or terms) in vocabulary
R	the number of the relevance status
m, z, v	document, topic, word
w_i, z_i,	word in position i (word i), topic of word i
x, r	keyword indicator, relevance status
p, q	beta prior for ω
δ, ϵ	smoothing prior, weak smoothing prior
β^r	term selector (value of term selector $\in$ {0,1})
θ	multinomial distribution over topics
π	bernoulli distribution over relevance status
ω	bernoulli distribution over term selector
φ^r	multinomial distribution over topical words
φ^{ir}	multinomial distribution over irrelevant topical words
α, β^{ir}, γ	Dirichlet prior for θ, φ^{ir}, Beta prior for π
x_m, r_m,	keyword indicator, relevance status of document m
β_v^r, $\beta_{t,v}^r$	term selector of term v under relevance status r, term selector of term v in topic t
$\boldsymbol{w}$, $\boldsymbol{z}$, $\boldsymbol{r}$	all words, assigned topics and relevance status
$\boldsymbol{z}^{-i}$	all assigned topics except the one for word i
$\boldsymbol{\beta^{r(-v)}}$	all selected terms except the term v under relevance status r
$\boldsymbol{\beta_t^{r(-v)}}$	all selected terms except the term v in topic t
$f_{r,m,v}$	the frequency of vocabulary term v in document m under relevance status r
$C_r^{R(-m)}$	the number of documents under relevance r except document m
$C_{r,\,v}^{RW}$	the number of words of vocabulary term v under relevance status r
$C_{r,m,t}^{RMT(-i)}$	the number of words under relevance r and topic t in document m except word i
$C_{r,\,t,\,v}^{RTW}$	the number of words of vocabulary term v under relevance r and topic t

TABLE I: Definitions of notations

2.2.1 Generative Process

Here we present and illustrate the generative process:

1. Draw $\varphi^{ir} \sim Dirichlet(\boldsymbol{\beta^{ir}})$ as a word distribution of a irrelevant topic to the targeted aspect;
2. For each target-relevant topic $t \in \{1, 2, ..., T\}$:
   (a) Draw a prior distribution $\omega_t \sim Beta(p, q)$;
   (b) For each term $v \in \{1, 2, ..., V\}$:
      i. Draw a term selector $\beta^r_{t,\,v} \sim Bernoulli(\omega_t)$;
   (c) Draw a word distribution $\varphi^r_t \sim Dirichlet(\boldsymbol{\beta^r_t \delta} + \boldsymbol{\epsilon})$;
3. For each document $m \in \{1, 2, ..., M\}$:
   (a) Draw a prior distribution $\pi_m \sim Beta(\boldsymbol{\gamma})$;
   (b) Draw relevance status r based on keyword indicator x and $Bernoulli(\pi_m)$;
   (c) If the document is relevant to the targeted aspect, i.e., $r = 1$:
      i. Draw a topic $z \sim Multinomial(\theta^r)$;
      ii. Emit a word $w_i \sim Multinomial(\varphi^r_z)$.
   (d) If the document is irrelevant to the targeted aspect, i.e., $r = 0$:
      i. Emit a word $w_i \sim Multinomial(\varphi^{ir})$

For better understanding, we use three sample documents from the *e-cigarette* (*e-cig* for short) domain for illustration:

($d1$) *e-cig is a gateway to smoking for children.*

($d2$) *it explores gateway effect of e-cig for kids.*

($d3$) *I saw a woman smoking e-cig on the street.*

As we assumed above, a document talks about one aspect. Then when the targeted aspect is given by a user, a document can be identified as relevant or irrelevant to the targeted aspect. r represents this relevance. $r \in \{0, 1\}$, where r=1 means the document is relevant to the target and r=0 is irrelevant. Another related variable is x, which represents whether a document contains at least one keyword $s \in S$. $x \in \{0, 1\}$, where x=1 indicates the document contains the keyword(s) and x=0 indicates it does not. For example, when S is {"children"} and a document m says "e-cigarette is a gateway to smoking for *children*" (i.e., example $d1$), the keyword indicator x_m=1 because the document m contains the keyword "children". In this case (x_m=1), m is regarded as relevant (r_m=1) because it is unlikely that a short sentence contains the word "children" and is not talking about *children*. However, this is a soft constraint that can be relaxed by adjusting a control factor λ (presented in Equation Equation 2.1) and $0 \leqslant \lambda \leqslant 1$, i.e., λ controls how much we believe a document contains a keyword is actually relevant. When it comes to the opposite situation that there is no keyword found in a documents m (i.e., x_m=0), it is a different case because the document can be either relevant or irrelevant. For instance, the above example $d2$ is clearly relevant to target *children* while example $d3$ is not and they both do not contain the keyword "children". We will discuss how to handle this case (x_m=0) in the following sub-sections.

After the relevance r of a document is drawn, we see how a word w_i is generated. As discussed above, there are two types of relevance status for each document. When r=1, a topic t is chosen from θ^r. The total number of topics is $|T|$ and these topics are the topics related to the target. After that a word is emitted from the selected topic by φ_t^r. When r=0 the

generative process is similar but TTM does not further generate multiple topics of the non-targeted aspects because they are not related to the user-interested aspect. Thus a word w_i is emitted directly from φ^{ir}. In other words, the words in the irrelevant documents are drawn from only one (irrelevant) topic.

2.2.2 Targeted Modeling with Sparsity

This sub-section presents how TTM achieves the targeted modeling by exploiting the idea of sparsity. Traditionally, sparsity in topic modeling indicates that a topic usually focuses on a narrow range of words rather than a wide range of them in the vocabulary, or a document focuses on a very small number of topics (Wang and Blei, 2009; Lin et al., 2014). Similarly, in TTM we use a similar concept called *aspect sparsity*, which consists of two parts: *document-aspect sparsity* and *targeted-aspect sparsity*, which are detailed below.

Since a sentence is regarded as a document, TTM already assumes that each document focuses on only one aspect. The document-aspect sparsity is then naturally achieved. Additionally, the idea of targeted modeling with sparsity is based on two importance observations.

First, the targeted aspect may only be a small part or a minority among all aspects in a given corpus. For example, *Children* is only one aspect in the e-cigarette domain, which has a large number of other aspects such as *Elderly*, *Vaporizer* and *Health*. This observation gives raise to the targeted-aspect sparsity.

Second, since each document is coupled to one aspect (using document-aspect sparsity), if we can better represent the targeted aspect, we should be able to better extract its relevant documents. If we can extract more relevant documents for the targeted aspect, we are likely

to discover other more important words to better represent the target. As a result, better and more topics are more likely to be generated because of better representative words and more relevant documents about the target are identified. Thus, we believe jointly modeling of relevance status r and the targeted aspect can benefit the topic discovery.

As introduced in section 2.4, sparsity has been used to represent the skewed data distribution in topic modeling. It provides a possible way to represent the property of minority (the first observation above) and to model the targeted aspect with document relevance in a unified framework (the second observation above). We therefore follow this direction and propose the idea of *biased sparsity* to help achieve the targeted modeling and the aspect sparsity.

2.2.2.1 Biased Sparsity

Note that a target (aspect) is essentially represented by words. So the problem is how to automatically discover representative words of the targeted aspect in a joint modeling manner. As discussed above, a targeted aspect is sparse among all aspects. From it we can further posit that (in most cases) a targeted aspect is also sparse compared to the combination of all non-targeted aspects. When combining this statement with the scenario that *a targeted aspect is represented by words*, we can conclude that the number of important words for distinguishing the targeted aspect from non-targeted aspects is in a narrow range. That is, the representative words for the targeted aspect are sparse. These words are denoted by $V_{r=1}$. However, the representative words from the combination of all other non-targeted aspects are probably not sparse because that combination needs to contain almost all possible words for describing all

other aspects. These words are denoted by $V_{r=0}$. Now we introduce the proposed *biased sparsity*, which is used for realizing the above idea.

The proposed *biased sparsity* approach is to make the word sparsity biased much more towards the targeted aspect (i.e., $|V_{r=1}| \ll |V|$) and keep the non-targeted aspects almost non-sparse (i.e., $|V_{r=0}| \approx |V|$). In other words, the word distribution of the targeted aspect (demoted by φ^r) only focuses on a small number of representative words. But the word distribution of non-targeted aspects (denoted by φ^{ir}, where ir means a unified irrelevant topic) contains almost all possible words. With this setting, only those words that simultaneously satisfy both the following two conditions can be selected as representative words for the targeted aspect: (a) they are semantically correlated with the known words of the target (achieved by the power of topic modeling) and (b) they can distinguish the targeted aspect from non-targeted aspects (constrained by *biased sparsity*).

2.2.2.2 Targeted Modeling

This sub-section demonstrates the targeted modeling in TTM, with the incorporation of biased sparsity. Here we follow the previous examples in section 2.2.1 to illustrate. Recall that S={"children"} and the document $d1$ containing the keyword (x_{d1}=1) is known to be relevant (r_{d1}=1). Although the word "children" is not in document $d2$, the word "gateway" (also in $d1$) serves as a bridge to connect the words in $d2$ via topic modeling. If the "gateway" has been identified as a discriminative word for target *children*, it makes $d2$ more probable to be relevant (r_{d2}=1) as $d2$ also contains the word "gateway". In this case, even though $d2$ has no keyword "children" it still has a high probability to be relevant. In contrast, although "smoking" appears

in both documents $d1$ and $d3$, if "smoking" is not identified as a discriminative word for aspect *children*, $d3$ will be mostly treated as irrelevant (r_{d3}=0).

Biased sparsity is the key to identify discriminative words for the targeted aspect, which is done by imposing the sparse constraint ($|V_{r=1}| \ll |V|$ and $|V_{r=0}| \approx |V|$) discussed above. Specifically, when the target is *children*, "gateway" will be a word included in φ^r and "smoking" will not be. An explanation is that "smoking" is also widely used by other non-targeted aspects like *Elderly* and *Health*, while "gateway" is used in *Children* more often. Thus, "gateway" is more discriminative for *children* but "smoking" is too genreal.

Another crucial factor is the relevance status r. When more discriminative words (like "gateway") are included in φ^r, they in return increase the overall probability of the documents (like $d2$) that contain those words (like "gateway") to be relevant (r=1). When those documents are identified as relevant, new representative words (like "kids" in $d2$) for the target aspect are also found and added to φ^r. Note that the growing of words in φ^r will not be an endless process because of the sparsity constraint. After that these new words (like "kids") in φ^r will help detect other relevant documents. This process shows the strength of joint modeling of the target aspect (using biased sparsity) and the relevance status r in TTM. It also explains why TTM can finally shape better topics. Because it is able to exploit the information in other relevant documents in C even without knowing the additional keywords such as "kids", "young" and "minors", as they are automatically found and involved in the modeling process.

In a nutshell, the relevance variable r and the biased-sparsity related variable φ^r and φ^{ir} function together to ensure the property of aspect sparsity. Specifically, the encoding of φ^r and φ^{ir} to achieve *biased sparsity* is based on the implementation of spike-and-slab prior.

2.2.2.3 Spike-and-Slab Prior

The spike-and-slab prior is incorporated in the probabilistic topic model to realize a switcher-like "on" and "off" selector. This prior is first introduced in (Mitchell and Beauchamp, 1988) and is recently reported as an effective way to reduce the model uncertainty (Ishwaran and Rao, 2005). It can also be used to decouple the sparsity and smoothing problem in Dirichlet distribution (Bengio et al., 2011). It has been used in topic modeling for word sparsity (Wang and Blei, 2009; Lin et al., 2014). In brief, "spike" controls the selection of a word, while "slab" smooths the word selected by "spike".

The spike-and-slab prior and other related components are together formulated as random variables in our model for the implementation of the biased sparsity. The related variables are ω, p, q, β^r, δ, ϵ and φ^r. For every word $v \in V$, $\beta_v^r \in \{0, 1\}$ is the specific word selector. When β_v^r=1 a word is selected and β_v^r=0 a word is not selected. Note that in TTM, for a keyword $s \in S$ the β_s^r is set to 1. It is intuitive as those keywords are given by the user for specifying the targeted aspect. Different from β^r, β^{ir} is a predefined prior. It is symmetric for all other words except the known keywords $s \in S$, because the known keywords given by the user is very unlikely to be irrelevant to the target.

Unlike non-probabilistic models, one knotty issue for probabilistic models is the decoupling of sparsity and smoothing (Wang and Blei, 2009). Motivated by (Lin et al., 2014), we incorporate a weak smoothing prior ϵ in addition to the regular smoothing prior δ.

Double Sparsity: In addition to the *target sparsity*, the words for a specific topic related to the targeted aspect is also sparse. That is reasonable because TTM aim at discovering more fine-grained topics and the words for describing such a topic are naturally sparse. Likewise, the $\varphi^r_{t,v}$ where $t \in T$ and $v \in V$ is also encoded with a slpike-and-slab prior in TTM. Together with the target sparsity, the sparsity of topical words enables the model to generate more coherent topics of the targeted aspect.

2.2.3 Inference

We use Gibbs Sampling (Griffiths and Steyvers, 2004) for model inference. The conditional distributions are shown in Equations Equation 2.1, Equation 2.2, Equation 2.3, Equation 2.4 and Equation 2.5 (see Table VII for the meanings of notations). The * symbol means the summation of all instances of the corresponding variable, for example, $C^{RTW(-i)}_{r,\,t,\,*}$ is the number of all words under relevance status r and topic t except the word in position i.

First, we sample the relevance status r for every document m, where $r \in R$ and $m \in M$.

$$P(r_m = c | x_m = d, \mathbf{\Theta}), \mathbf{\Theta} = \{\boldsymbol{r_m^{-i}}, \boldsymbol{w}, \boldsymbol{\pi}, \boldsymbol{\beta^r}, \boldsymbol{\beta^{ir}}, \boldsymbol{\gamma}, \delta, \epsilon\}$$
$$\propto \begin{cases} g(c, \mathbf{\Theta}) & d = 0, c = 0 \; or \; 1 \\ \lambda \times g(c, \mathbf{\Theta}) & d = 1, c = 1 \\ (1 - \lambda) \times g(c, \mathbf{\Theta}) & d = 1, c = 0 \end{cases} \tag{2.1}$$

$$g(c, \boldsymbol{\Theta}) =$$

$$\begin{cases} \frac{C_c^{R(-m)}+\gamma}{C_*^{R(-m)}+|R|\gamma} \times \frac{\prod_v^V \Gamma(\beta_v^r C_{c,\,v}^{RW(-m)}+\beta_v^r\delta+\epsilon+f_{c,m,v})}{\Gamma(\sum_v^V (\beta_v^r C_{c,\,v}^{RW(-m)}+f_{c,m,v})+|\beta_*^r|\delta+|V|\epsilon)} \\ c = 1 \\ \frac{C_c^{R(-m)}+\gamma}{C_*^{R(-m)}+|R|\gamma} \times \frac{\prod_v^V \Gamma(C_{c,\,v}^{RW(-m)}+\beta_v^{ir}+f_{c,m,v})}{\Gamma(\sum_v^V C_{c,\,v}^{RW(-m)}+f_{c,m,v}+\beta_j^{ir})} \\ c = 0 \end{cases} \tag{2.2}$$

Second, we sample the term selector β_v^r, where $v \in V$. $|\beta_*^r|$ is the sum of the values of all such term selectors.

$$P(\beta_v^r = b|\boldsymbol{\beta^{r(-v)}}, \boldsymbol{r}, \boldsymbol{w}, \delta, \epsilon, p, q) \propto$$

$$\begin{cases} \Gamma(C_{r=1,\,v}^{RW} + \delta + \epsilon) \times \Gamma(|\beta_*^{r(-v)}|\delta + |V|\epsilon + C_{r=1,\,*}^{RW(-v)}) \\ \times \Gamma(|\beta_*^{r(-v)}|\delta + \delta + |V|\epsilon) \times (p + |\beta_*^{r(-v)}|) & b = 1 \\ \Gamma(\delta + \epsilon) \times \Gamma(|\beta_*^{r(-v)}|\delta + \delta + |V|\epsilon + C_{r=1,\,*}^{RW(-v)}) \\ \times \Gamma(|\beta_*^{r(-v)}|\delta + |V|\epsilon) \times (q + |V| - |\beta_*^{r(-v)}| - 1) & b = 0 \end{cases} \tag{2.3}$$

Third, we sample a topic of the word in position i. We do it for all words in the corpus. $|\beta^r_{t,*}|$ is the sum of the values of all such term selectors in topic t.

$$P(z_i = t|\boldsymbol{z}^{-i}, \boldsymbol{r}, \boldsymbol{w}, \alpha, \boldsymbol{\beta^r}, \boldsymbol{\beta^{ir}}, \delta, \epsilon) \propto$$
$$\begin{cases} \frac{C^{RMT(-i)}_{r_i,m,t}+\alpha}{C^{RMT(-i)}_{r_i,m,*}+|T|\alpha} \times \frac{(\beta^r_{t,v} C^{RTW(-i)}_{r_i,\ t,\ w_i}+\beta^r_{t,v}\delta+\epsilon)}{C^{RTW(-i)}_{r_i,t,*}+|\beta^r_{t,*}|\delta+|V|\epsilon} & r_i = 1 \\ \frac{C^{RMT(-i)}_{r_i,m,t}+\alpha}{C^{RMT(-i)}_{r_i,m,*}+|T|\alpha} \times \frac{(C^{RTW(-i)}_{r_i,\ t,\ w_i}+\beta^{ir}_{t,v})}{C^{RTW(-i)}_{r_i,t,*}+|\beta^{ir}_{t,*}|} & r_i = 0 \end{cases} \tag{2.4}$$

Last, we sample the term selector $\beta_{t,v}$, where $t \in T$ and $v \in V$.

$$P(\beta^r_{t,v} = s|\boldsymbol{\beta^{r(-v)}_t}, \boldsymbol{z}, \boldsymbol{r}, \boldsymbol{w}, \delta, \epsilon, p, q) \propto$$
$$\begin{cases} \Gamma(C^{RTW}_{r=1,\ t,\ v} + \delta + \epsilon) \times \Gamma(|\beta^{r(-v)}_{t,*}|\delta + |V|\epsilon + C^{RTW(-v)}_{r=1,\ t,\ *}) \\ \times \Gamma(|\beta^{r(-v)}_{t,*}|\delta + \delta + |V|\epsilon) \times (p + |\beta^{r(-v)}_{t,*}|) & s = 1 \\ \Gamma(\delta + \epsilon) \times \Gamma(|\beta^{r(-v)}_{t,*}|\delta + \delta + |V|\epsilon + C^{RTW(-v)}_{r=1,\ t,\ *}) \\ \times \Gamma(|\beta^{r(-v)}_{t,*}|\delta + |V|\epsilon) \times (q + |V| - |\beta^{r(-v)}_{t,*}| - 1) & s = 0 \end{cases} \tag{2.5}$$

2.3 Experiment

2.3.1 Experimental Setup

Data and targeted aspects: Five real-world data sets in different domains are used in our experiments, namely, E-Cigarette, Cigar, Camera, Cell-Phone and Computer. The first two data sets are tweets collected from Twitter in October 2014. Specifically, E-Cigarette and

Cigar are two types of tobacco-related products, which are the research areas of the last author from health science. The last three datasets are product reviews of three popular electronic products. We crawled the reviews from Amazon.com. More detailed information about the data sets is presented in Table XII.

Dataset (Area)	**Source**	**Size**	**Targeted Aspects**
E-Cigarette	Twitter	50k	Children, Vape, Health
Cigar	Twitter	50k	Event, Box, Horse[1]
Camera	Amazon	10k	Lens, Screen, Weight
Cell-Phone	Amazon	10k	Sound, Case, Battery
Computer	Amazon	10k	Monitor, Software, Warranty

TABLE II: Five datasets, targeted aspects, and initial documents (tweets or review sentences)

Three targeted aspects are picked from each domain for targeted analysis. The aspects cover a wide range of diverse areas: Some of them are typical or frequent aspects in its domain like *Children* in E-Cigarette and *Lens* in Camera. Some are small or infrequent aspects like *Weight*, *Warranty* and *Horse*[1]. Note that one infrequent topic is specially chosen for each domain, listed as the last one in Table XII.

---

[1]A special topic: there was an well-known race horse called "Cigar" who died in October 2014. It is covered in the *Cigar* dataset as people talked about it in social media using its name. It is an evidently infrequent aspect in the Cigar data.

Parameter Setting: For the hyper-parameter setting, we place: $q = p = 1$ for a uniform Beta; $\gamma = 1$, $\alpha = 1$, $\beta^{ir} = \delta = 0.001$ and ϵ =1.0×10^{-7}. The models for comparison are also with the same setting. λ in Equation Equation 2.1 is set to 1 for TTM.

2.3.2 Baseline Models for Comparison

To evaluate our proposed targeted topic model TTM, we compare it with the following baseline models:

LDA: LDA is a well-known topic model (Blei et al., 2003). It finds all topics in a corpus. To identify topics that are relevant to the targeted aspect, we manually inspect all resulting topics from LDA and find the subset of relevant topics. Note that the targeted aspect may be split into multiple topics by LDA. This is a labor-intensive and tedious process if there is a large number of topics from LDA.

LDA*: We still use LDA for topic generation, but instead of manual inspection to find related topics, we use keywords in S to search for relevant topics. We refer to this as the *search strategy*, which eliminates the tedious manual process. In this approach the number of topics T can be set large because only the retrieved topics will be analyzed. We search only the top 20 words of each topic.

DS-LDA: This is a state-of-art probabilistic sparse topic model (Lin et al., 2014) that models both the sparsity of topic mixtures (mining salient topics of a document) and topical words (mining representative words of a topic). We follow the implementation in (Lin et al., 2014) and refer to this as dual-sparse topic model, DS-LDA, in this paper. Like that for LDA, the relevant topics for DS-LDA are found via manual inspection.

DS-LDA*: Like LDA*, we use keywords S and adopt the *search strategy* to find possible relevant topics from all topics generated by DS-LDA.

SS-LDA: We also use a single-sparse topic model named SS-LDA in this paper for comparison, to see whether the direct injection of the word sparsity can help our task, because it enables the topical word to be more focused. The realization is similar to DS-LDA but it only addresses the sparsity of topical words. Likewise, it requires manual inspection.

SS-LDA*: Like DS-LDA*, a *search strategy* is adopted to find possible relevant topics from all topics.

LDA-PD: This model runs LDA only on the documents in each dataset that contains one or more keywords from S. For instance, to find topics about targeted aspect *Children*, we use the keyword "children" to search for tweets in the *E-Cigarette* tweet corpus. After that we run LDA on the resulting tweets to find topics. We name this approach LDA-PD (where PD means Partial Data) for short.

2.3.3 Quantitative Evaluation

Since our goal is to discover topics for a given targeted aspect but the correct number of topics and the number of terms/words under each topic are unknown, a natural evaluation metric is to give the precision results at different rank position n, called $Precision@n$ (or $P@n$ for short).

For the evaluation of the first two datasets, two experts from our health science collaborator's team who are specialized in the tobacco-related products and social media were invited to judge

the results. Two human labelers who are familiar with Amazon product reviews labeled the results from the other three datasets. Cohen's Kappa scores of the two genres are 0.792 and 0.822 respectively.

Evaluation Measure: We use a normalized form of precision (defined in Equation Equation 2.6) that can evaluate both the correctness of the topical words and the number of detected topics in a unified manner as both are important. Specifically, after the human judgment of all models results, the number of Maximum Unique Topics (MUT) can be obtained. For example, when the target is *X*, *Model-1* finds one unique topic A (which is correct), *Model-2* finds two unique topics A, B and *Model-3* finds three unique topics A, B, C then MUT is 3. In another case, when *Model-1* finds two unique topics A, B, *Model-2* finds two unique topics B, C and *Model-3* finds topics B, C then the MUT is also 3.

$$P_{(i)}@n = \frac{\sum_{st}^{ST} \#C_{(i)st}(correct@n)}{\sum_{mt}^{MUT} \#C_{mt}(words@n)} \tag{2.6}$$

In Equation Equation 2.6, $P_{(i)}@n$ indicates the $precision@n$ for model (i), given the targeted aspect. $\#C_{(i)st}(correct@n)$ is the number of correct words found in the topic st, given that there are ST topics found by model i. $\#C_{mt}(words@n)$ is the maximum number of correct words from all models.

This evaluation measure is fair and reasonable because a model may only find one correct topic with high topical word precision but miss some correct topics. Note that if there are more than one identical topics generated by a model, their average score is used. If there are multiple

topics mixed in a single topic generated by a model, we use the best topic based on the number of relevant words in the top 20 words.

Two Comparison Settings: Two different experiment settings are used for comparison due to different properties of the candidate models. The reasons and differences will be elaborated in the following sub-sections.

2.3.3.1 Precision in Setting One

Here we compare LDA, DS-LDA, SS-LDA and LDA-PD with TTM. In this setting the *search strategy* is not used. Instead, the annotators were asked to go through all top (20) words in the generated topics. For LDA-PD, we use the target keyword itself (e.g., "children") to extract documents. For TTM, we also use the target keyword for targeted modeling. The topic number T in LDA-PD and TTM is set to 5 or 10 because we have no prior information about the number of target-related topics but we know that it intuitively depends on the prevalence of the targeted aspect (e.g., in the Camera domain the targeted aspect *Screen* is likely to be frequent and have more topics than the infrequent aspect *Weight*). In general, T is set to 5 for *infrequent* targets and to 10 for other more *frequent* targets. Although T could be 5 or 10 according to the targeted aspect, the same value of T is used for the comparison of results for both LDA-PD and TTM. Likewise, for LDA, DS-LDA and SS-LDA, T is set to 15 or 30 (and choose the one produces higher precision results). The numbers are larger because they do not directly generate topics for the targeted aspect like LDA-PD and TTM. They also produce topics for other non-targeted aspects.

Domain	Aspect	LDA			DS-LDA			SS-LDA			LDA-PD			TTM		
		P@5	P@10	P@20	P@5	P@10	P@20	P@5	P@10	P@20	P@5	P@10	P@20	P@5	P@10	P@20
E-Cig	children	0.15	0.14	0.13	0.15	0.10	0.07	0.15	0.11	0.09	0.48	0.35	0.27	0.73	0.59	0.41
	vape	0.23	0.30	0.24	0.16	0.10	0.09	0.36	0.32	0.22	0.59	0.42	0.22	0.76	0.57	0.30
	health	0.48	0.39	0.30	0.32	0.19	0.18	0.38	0.25	0.17	0.38	0.28	0.20	0.55	0.42	0.30
Cigar	box	0.46	0.35	0.22	0.15	0.13	0.08	0.20	0.23	0.15	0.53	0.51	0.34	0.65	0.54	0.41
	event	0.30	0.28	0.31	0.24	0.14	0.09	0.36	0.18	0.15	0.30	0.39	0.33	0.64	0.58	0.45
	horse	0.27	0.20	0.26	0.15	0.10	0.06	0.55	0.43	0.24	0.73	0.46	0.34	0.88	0.61	0.37
Camera	screen	0.51	0.35	0.35	0.12	0.15	0.13	0.30	0.25	0.15	0.44	0.31	0.27	0.74	0.61	0.40
	lens	0.25	0.16	0.13	0.21	0.17	0.12	0.31	0.27	0.19	0.36	0.34	0.30	0.53	0.36	0.34
	weight	0.20	0.20	0.15	0.04	0.02	0.04	0.16	0.10	0.08	0.56	0.44	0.25	0.72	0.60	0.44
Cellphone	sound	0.58	0.40	0.30	0.27	0.17	0.14	0.50	0.35	0.21	0.52	0.37	0.29	0.75	0.60	0.40
	case	0.56	0.59	0.53	0.14	0.18	0.15	0.23	0.16	0.14	0.61	0.44	0.38	0.66	0.56	0.40
	battery	0.41	0.30	0.20	0.30	0.20	0.14	0.16	0.12	0.12	0.48	0.38	0.25	0.60	0.42	0.42
Computer	monitor	0.72	0.60	0.42	0.11	0.11	0.06	0.57	0.39	0.26	0.66	0.54	0.43	0.70	0.56	0.45
	software	0.48	0.39	0.35	0.24	0.21	0.14	0.36	0.30	0.33	0.36	0.34	0.30	0.60	0.39	0.35
	warranty	0.31	0.23	0.19	0.03	0.03	0.04	0.23	0.13	0.08	0.47	0.38	0.31	0.63	0.53	0.43
average score		0.39	0.32	0.27	0.18	0.13	0.10	0.32	0.24	0.17	0.50	0.40	0.30	0.68	0.53	0.39
improvement by TTM		+0.28	+0.21	+0.12	+0.50	+0.40	+0.29	+0.35	+0.29	+0.22	+0.18	+0.13	+0.09	n/a	n/a	n/a

TABLE III: Precisions of setting one. The last two rows are (a) the average scores of all targeted aspects of all topics and (b) the improvement achieved by TTM over other models respectively.

The precision results at the rank position of 5, 10 and 20 are reported in Table III and we observe the following:

1. TTM significantly outperforms other models. The average scores and the improvements by TTM are presented in the last two rows in Table III. Among them, LDA-PD obtains the second best scores. Compared to LDA-PD, TTM has two key advantages, which are also the main reasons for the higher scores. (a) TTM can identify and use other relevant documents in modeling the topics of the targeted aspect with the power of biased sparsity, while LDA-PD can only generate topics from the subset of documents selected by user-specified keywords. (b) TTM can find more or better topics for the targeted aspect than

LDA-PD because LDA-PD discards many relevant documents, resulting in information loss. Concrete examples will be further given in sub-section 2.3.4.

2. LDA-PD is the second best because it rules out the irrelevant documents that can interfere with good topic identification. But then it also loses relevant information as we discussed above. Thus, it results in inferior performance to TTM. Moreover, LDA-PD works poorly when it comes to some *infrequent* targets because relevant documents about these targets are not easy to extract as they may not contain the keywords.
3. LDA is neither as good as TTM nor LDA-PD. But it is better than both DS-LDA and SS-LDA. The DS-LDA that models both the sparsity of topic mixture and topical words has the worse performance. We will analyze the reason in the next sub-section.

One might argue that we can keep increasing T to a larger number to make further improvement. That is possible, but it becomes rather messy and impractical (labor intensive and time consuming) for the user to manually inspect all generated topics. An alternative is to apply the *search strategy* with increased T, which leads to our following evaluation of setting two.

2.3.3.2 Precision in Setting Two

This subsection compares LDA*, DS-LDA* and SS-LDA* against TTM. Different from setting one, here the annotators utilize the *search strategy* to identify relevant topics from all models except TTM. That is, the targeted aspect keyword is used to search in the top 20 topical words in each topic to find possibly relevant topics to the target. Only those resulting topics are evaluated. Since search is used, the number of topics T can be large. For LDA* we set T to 15, 30 and 50. For DS-LDA* and SS-LDA* we follow the same setting but their results are

Domain	Aspect	LDA*(T-15)			LDA*(T-30)			LDA*(T-50)			DS-LDA*(T-50)			SS-LDA*(T-50)		
		P@5	P@10	P@20	P@5	P@10	P@20	P@5	P@10	P@20	P@5	P@10	P@20	P@5	P@10	P@20
E-Cig	children	0.08	0.05	0.04	0.13	0.13	0.11	0.20	0.13	0.08	0.05	0.30	0.04	0.16	0.12	0.08
	vape	0.12	0.14	0.13	0.17	0.14	0.17	0.28	0.30	0.23	0.08	0.06	0.06	0.36	0.32	0.19
	health	0.13	0.13	0.12	0.17	0.12	0.06	0.32	0.23	0.16	0.14	0.11	0.11	0.13	0.15	0.8
Cigar	box	0.30	0.25	0.20	0.20	0.24	0.20	0.43	0.26	0.16	0.08	0.06	0.08	0.33	0.29	0.18
	event	0.16	0.08	0.12	0.24	0.20	0.18	0.19	0.18	0.14	0.12	0.08	0.05	0.8	0.60	0.70
	horse	0.05	0.10	0.11	0.25	0.18	0.24	0.50	0.42	0.31	0.05	0.04	0.03	0.5	0.28	0.16
Camera	screen	0.08	0.06	0.08	0.08	0.06	0.06	0.08	0.05	0.06	0.23	0.20	0.14	0.15	0.11	0.06
	lens	0.13	0.10	0.08	0.18	0.16	0.12	0.12	0.11	0.10	0.17	0.13	0.11	0.27	0.16	0.11
	weight	0.08	0.08	0.08	0.16	0.14	0.10	0.12	0.10	0.07	0.04	0.04	0.03	0.13	0.1	0.13
Cellphone	sound	0.20	0.15	0.13	0.47	0.37	0.27	0.63	0.45	0.28	0.17	0.12	0.10	0.3	0.2	0.15
	case	0.38	0.35	0.31	0.55	0.51	0.36	0.39	0.37	0.30	0.13	0.11	0.21	0.2	0.2	0.18
	battery	0.16	0.12	0.08	0.32	0.22	0.16	0.32	0.20	0.16	0.16	0.08	0.06	0.12	0.12	0.11
Computer	monitor	0.59	0.56	0.40	0.61	0.51	0.38	0.61	0.46	0.38	0.20	0.10	0.09	0.31	0.24	0.18
	software	0.32	0.24	0.19	0.12	0.11	0.09	0.18	0.15	0.11	0.08	0.04	0.04	0.16	0.12	0.1
	warranty	0.10	0.08	0.05	0.30	0.23	0.18	0.33	0.30	0.20	0.07	0.03	0.03	0.23	0.18	0.11
average score		0.19	0.17	0.14	0.26	0.22	0.18	0.31	0.25	0.18	0.12	0.10	0.08	0.28	0.21	0.22
impr. by TTM		+0.49	+0.37	+0.26	+0.41	+0.31	+0.22	+0.36	+0.28	+0.22	+0.56	+0.43	+0.32	+0.40	+0.32	+0.18

TABLE IV: Precisions of setting two. The last two rows are (a) the average scores of all targeted aspects of all topics and (b) the improvement achieved by TTM over other models respectively.

not good as LDA*. Due to the limited space we show the T=50 for DS-LDA* and SS-LDA* only.

The precision results at the rank position of 5, 10 and 20 are reported in Table IV and we observe the following:

1. TTM again outperforms the other models by a large margin. We can see that the results from the search strategy is also poor. TTM generates better topics without human invention.
2. The increase of T for LDA* helps improve the performance but the enhancement is in a decreasing trend. Although the precision scores improve with increased T, the improve-

ment from LDA*(T-30) to LDA*(T-50), which are 5.0%, 2.5%, 0.3% for P@5, P@10, P@15, drops from LDA*(T-15) to LDA*(T-30), which are 7.1%, 5.5%, 3.7% respectively. One reason is that with the increment of T, although more possible relevant topics may be generated, the *search strategy* can only find a subset of the relevant topics because the complete set of keywords for the targeted aspect is unknown. For instance, for aspect *Children* there might be new topics formed about children with a larger T but they do not contain the keyword "children". Instead, the new topics may contain other related keywords such as "kids", "young", and "minors".

3. Neither SS-LDA nor SS-LDA* can produce better results than LDA or LDA*. The reason is that when the word sparsity can find focused words to represent a topic, the number of related topics (to the target) found by SS-LDA is fewer than that by LDA. In other words, SS-LDA might form better topics but they are less related to the targeted aspect. DS-LDA has the worse results because it may find more focused topics for individual documents but not for the targeted aspect of user interest.

2.3.4 Qualitative Evaluation

This section presents the qualitative evaluation. We show several resulting topics in Table V and Table VI to give a flavor of each system. The domain and the targeted aspect are shown at top of the tables above the model names and topic names (given by us). Incorrect words of a topic are italicized and marked in red. Notice that the targeted aspect keyword itself is also shown but it is excluded from the computation of precision in the previous section because it is already known.

Domain: E-Cigarette					
Targeted aspect: Children					
TTM		**LDA-PD**		**LDA* (T-50)**	
fears	**regulation**	**fears**	**regulation**	**fear**	**regulation**
children	states	children	buy	fears	buy
fears	buy	fears	states	*research*	legally
nicotine	live	smoked	children	concerns	children
nightmare	legally	sale	live	*cloud*	kids
sales	*atlanta*	*real*	legally	*parents*	purchase
data	young	*report*	*online*	*fulled*	state
age	smoked	easy	*press*	*batteries*	live
gateway	*online*	*figures*	*washington*	experimenting	*nocotine*
reason	laws	*top*	*smokeless*	*progress*	*nightmare*
fight	safe	*make*	*post*	smoked	*press*
LDA		**DS-LDA**		**SS-LDA**	
fears	**regulation**	**fears**	**regulation**	**fears**	**regulation**
fears	buy	children	*vaping*	*lacks*	regulation
city	kids	*evidence*	regulation	fears	*trending*
ban	legally	nicotine	*vaporizer*	*report*	*two*
council	children	*pour*	*view*	*stories*	*kanavape*
media	minors	*starter*	*prisoner*	*watch*	washington
social	*columbia*	*council*	*cut*	prison	safer
public	*district*	*fire*	*part*	*slapped*	*drinking*
study	*states*	*television*	washington	*world*	*thing*
addition	purchase	virus	state	evaporation	*cells*
bans	live	*business*	*social*	*set*	data

TABLE V: Topics of aspect *children* under E-Cig. Errors are italicized and marked in red.

2.3.4.1 Example One: E-Cigarette and Children

E-Cigarette (e-cig) is a key area studied by our collaborators from the health science and *Children* is one aspect that they are highly interested in and thus want to know its topics discussed on Twitter. Table V shows the topical words discovered by different models. The models are attached with the setting in the previous section so the topic number T is not explicitly given in the table. We explain the results below.

(1) Topic *fears*: It means the fears or concerns about children using e-cig. By comparison, we see that TTM generates a clearer and more reasonable topic. While the other methods generate incoherent topics with many wrong words, the words in TTM are more informative and interpretable, because the words like "nicotine", "gateway", and "sale" can better indicate the reasons of the fears as well as what people are actually concerned about for the children. That is, people worry about the "nicotine" in e-cig that is bad for kids; they are afraid that e-cig becomes a "gateway" for their kids to smoking, and the increasing "sale" to the young.

(2) Topic *regulation*: It is about the regulation of e-cig for children, mostly about the policy of purchase. Similar to topic *fears*, TTM finds more informative words like "laws" and "safe". Particularly, the word "young" is also included in the topic *regulation* of TTM, which is meaningful. It can infer that other documents that contain the unknown/unprovided keyword "young" are probably identified by TTM as relevant to the targeted aspect children (i.e., r=1) and thus involved in the topic generation of the target. Because one short sentence is unlikely to mention the keyword "children" and unknown keyword "young" at the same time (they are semantically similar). The documents containing the word "young" is unlikely to also contain "children" and vice versa. This is an advantage that LDA-PD cannot achieve. Those documents (containing "young") identified by TTM as relevant help generate a more coherent topic for the targeted aspect.

Here we also analyze the topics formed by LDA and LDA* (T-50) to demonstrate some aforementioned problems. (a) Although the topic *regulation* in LDA looks good with many good words grouped, the topic is actually not very informative as the words "children, kids,

minors" all ranked high but they do not tell anything about a topic related to e-cig and children. (b) Since these words are grouped together, it may lead to missing of some good topics of the targeted aspect. (c) When comes to LDA*, although T is set to 50 this method actually finds only 2 topics that contain "children" by using the *search strategy*. (d) Also in the *regulation* topics, one may notice that the word "minors" which is in LDA (where its topic number is 15 or 30) is not included in LDA*(T-50) any more. However, since we did not previously know all those keywords (e.g.,"minors", "young"), the topic is not found with $T = 50$, i.e., a related topic is unfortunately lost.

2.3.4.2 Example Two: Camera, Screen and Weight

The previous sub-section presents the example from tweets, which mainly reveals the trend of a discussed topic in Twitter. Now let us take a look at online reviews in the Camera domain. The data is from Amazon.com and the analysis is also related to opinion mining and sentiment analysis.

Since a full/comprehensive comparison has been done in the E-Cig domain, this subsection shows two different aspects (a popular aspect and an infrequent aspect) instead of repeating the full comparison with all models. Thus we only select some good models for comparison. Since LDA-PD achieves the second best score most of the time, it is included in our comparison. In addition, we pick one additional model that can also find the same topic. We analyze the two aspects *screen* and *weight* in the camera domain.

(1) Aspect *Screen*: When the target is screen, we first pick up the topic of *picture* for analysis. It discusses the features of the picture displayed on the *Screen*. From Table VI we

Domain: Camera					
Targeted aspect: Screen					
TTM			**LDA-PD**		**LDA**
picture	**menu**	**imaging**	**picture**	**menu**	**picture**
picture	easy	image	picture	screen	picture
color	control	shutter	photo	*touch*	photo
shot	manual	speed	*computer*	control	*easy*
sharp	photo	stabilization	video	menu	*video*
feature	function	shooting	black	button	feature
clear	menu	*lens*	*show*	easy	clear
resolution	condition	picture	full	*small*	pic
grainy	macro	frame	*turn*	*comcorder*	*fun*
action	*learn*	*feature*	*record*	*change*	*wonderful*
video	*situation*	produce	*print*	*lcd*	*beautiful*
Targeted aspect: Weight					
TTM			**LDA-PD**		**DS-LDA***
lens	**battery**	**carrying**	**lens**	**battery**	**-**
lens	battery	easy	weight	weight	*picture*
weight	weight	carry	lens	battery	compact
canon	light	*picture*	*zoom*	light	*stabilization*
size	*life*	compact	*extra*	*image*	weight
heavy	travel	case	size	*easy*	*place*
smaller	*rechargeable*	small	*add*	*shot*	*absolute*
long	lightweight	*extra*	light	*life*	*ease*
digital	*video*	big	*carry*	*control*	*instruction*
zoom	*image*	light	heavy	*focus*	*average*
trap	*control*	bulky	*battery*	*flash*	*release*

TABLE VI: Topics of two aspects *screen* and *weight* under Camera. Errors are italicized and marked in red.

observe that TTM produces a good result. LDA also has a topic about *picture* but it has an issue (though it might look good at the first glance). The problem is that it groups the words "picture", "photo", "pic" together as they are synonyms but the topic becomes vague because these words may belong to different fine-grained topics. In addition, while LDA finds some more general features like "wonderful", "beautiful" and "fun" (these words can be regarded as

general opinion words) because they actually can modify many different aspects) but TTM finds more specific (or coherent) features like "sharp" and "clear". This is an important property in opinion mining because people always want to know the specific reasons for opinions. Further illustration about this property in opinion mining can be found in (Wang et al., 2016b).

The quality of commonly found topics like *menu* is also improved. Additionally, it is worth noting that TTM identifies a new unique topic called *imaging*, which is not found by any other model, but is in fact related to aspect *screen*.

(2) Infrequent Aspect *Weight*: Only LDA-PD and TTM find meaningful related topics for the aspect Weight. Here *lens* and *battery* are two detected topics for demonstration. Because people often complain the heavy weight of these two components. However, they usually mention it in an implicit manner like "the lens is so heavy" or "the heavy weight battery is annoying". On the contrary, the topic generated by DS-LDA*(T-50) is not so clear. Moreover, TTM solely detects other interesting topics like "carrying", which is also closely related to the aspect Weight.

2.4 Related Work

To our knowledge, there is no existing topic model that is able to perform the proposed targeted analysis as we do. Our work is, however, clearly related to the classic topic models such as PLSA (Hofmann, 1999) and LDA (Blei et al., 2003) and their variants. These models have been used to discover hidden thematic structures in a collection of documents or corpus. There are numerous existing models (Rosen-Zvi et al., 2004; Ramage et al., 2009; Zhao et al., 2010; Mukherjee and Liu, 2012; Chen et al., 2012; Griffiths and Tenenbaum, 2004; Wal-

lach, 2006). They either identify topics only or jointly identify both topics and other types of information. For example, while both LDA and PLSA only identify topics, (Wang et al., 2010; Moghaddam and Ester, 2013) jointly model both topics and ratings in reviews. (Ramage et al., 2009; Eisenstein et al., 2011) model labeled data with the class information. (Mei and Zhai, 2005; Hong et al., 2011) conduct time-series analysis of topics. However, as we indicated in the introduction section, all these models and their variants are full analysis models. They aim to find all topics in the corpus, and none of them is able to perform targeted analysis based on only a specific aspect that is of interest to users. Existing research also proposed several knowledge-based topic models, which can incorporate prior domain knowledge in topic modeling (Andrzejewski et al., 2009; Mukherjee and Liu, 2012; Chen and Liu, 2014a; Wang et al., 2016b) to generate better results. But they are also full-analysis models, and do not help discover related topics of the user interested aspect.

Our work is also related to sparse topic models. The modeling of sparsity can represent skewed distributions, e.g., a topic usually focuses on a narrow range of words instead of a wide range of them in the vocabulary (Lin et al., 2014). This type of models is inspired by the influential power of the asymmetric Dirichlet prior (Wallach et al., 2009). Existing sparse topic models can be categorized into two types: (1) those that discover the salient (focused) topics of a specific document (Chen et al., 2012; Williamson et al., 2010), and (2) those that discover representative (focused) terms of a particular topic (Wang and Blei, 2009). Researchers also tried to achieve both at the same time in a hybrid manner (Lin et al., 2014; Archambeau et al., 2015). However, it is important to note that focused topics or terms in sparse models are

entirely different from our targeted analysis because they aim to achieve very high probabilities for a small number of topics in a document or for a small set of words/terms in a topic. They are still full-analysis models and cannot focus their modeling only on a user-specified aspect.

Although not directly related, there are other works of non-probabilistic models with sparse coding (Zhu and Xing, 2012; Min et al., 2010; Chen et al., 2011; Wang et al., 2011). However, they function quite differently in the goal or the methodology compared to probabilistic generative models. For example, they do not need to tackle the knotty issue of decoupling sparsity and smoothing (Wang and Blei, 2009) for sparsity realization in probabilistic models. Furthermore, they have been shown with poorer performances as the number of topics increases and are inferior compared to the probabilistic sparse topic models reported in (Lin et al., 2014). Most importantly, like other models, they do not do targeted analysis like we do.

2.5 Summary

In this work, we studied the novel problem of targeted modeling. Instead of finding all topics from a corpus like existing models based on full modeling, the proposed model focuses on finding topics of a targeted aspect to help the user perform deeper or finer-grained analysis. This is motivated by real-life applications that researchers are often not interested in everything in a corpus but only some aspects of it in order to answer their research questions. Existing full models are not the most effective methods for such focused analysis because their results are often too coarse and they may not find topics that the user is really interested in and/or miss many details. Experimental results showed that this is indeed the case and the proposed new model outperforms the state-of-the-art existing models markedly.

CHAPTER 3

TARGET-SENSITIVE MEMORY NETWORK FOR TARGET-BASED SENTIMENT CLASSIFICATION

(This chapter includes and expands on my paper previously published in *Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. "Target-Sensitive Memory Networks for Aspect Sentiment Classification". In* ***ACL 2018****.*)

3.1 Introduction

In this chapter, we introduce the task of target/aspect-based sentiment classification, which is a fundamental task of target-oriented sentiment analysis. To address it, we proposed novel models named target-sensitive memory networks.

Aspect sentiment classification (ASC), or target/aspect-based sentiment classification, is a core problem of sentiment analysis (Liu, 2012). Given an aspect and a sentence containing the aspect, ASC classifies the sentiment polarity expressed in the sentence about the aspect, namely, positive, neutral, or negative. Aspects are also called *opinion targets* (or simply *targets*), which are usually product/service features in customer reviews. In this paper, we use aspect and target interchangeably. In practice, aspects can be specified by the user or extracted automatically using an aspect extraction technique (Liu, 2012). In this work, we assume the aspect terms are given and only focus on the classification task.

Due to their impressive results in many NLP tasks (Deng et al., 2014), neural networks have been applied to ASC (see the survey (Zhang et al., 2018)). Memory networks (MNs), a type of neural networks which were first proposed for question answering (Weston et al., 2015; Sukhbaatar et al., 2015), have achieved the state-of-the-art results in ASC (Tang et al., 2016). A key factor for their success is the attention mechanism. However, we found that using existing MNs to deal with ASC has an important problem and simply relying on attention modeling cannot solve it. That is, their performance degrades when the sentiment of a context word is sensitive to the given target.

Let us consider the following sentences:

(1) *The screen resolution is **excellent** but the price is **ridiculous**.*

(2) *The screen resolution is **excellent** but the price is **high**.*

(3) *The price is **high**.*

(4) *The screen resolution is **high**.*

In sentence (1), the sentiment expressed on aspect *screen resolution* (or *resolution* for short) is positive, whereas the sentiment on aspect *price* is negative. For the sake of predicting correct sentiment, a crucial step is to first detect the sentiment context about the given aspect/target. We call this step *targeted-context detection*. Memory networks (MNs) can deal with this step quite well because the sentiment context of a given aspect can be captured by the internal attention mechanism in MNs. Concretely, in sentence (1) the word "excellent" can be identified as the sentiment context when *resolution* is specified. Likewise, the context word "ridiculous" will be placed with a high attention when *price* is the target. With the correct targeted-context

detected, a trained MN, which recognizes "excellent" as positive sentiment and "ridiculous" as negative sentiment, will infer correct sentiment polarity for the given target. This is relatively easy as "excellent" and "ridiculous" are both target-independent sentiment words, i.e., the words themselves already indicate clear sentiments.

As illustrated above, the attention mechanism addressing the targeted-context detection problem is very useful for ASC, and it helps classify many sentences like sentence (1) accurately. This also led to existing and potential research in improving attention modeling (discussed in Section 3.6). However, we observed that simply focusing on tackling the target-context detection problem and learning better attention are not sufficient to solve the problem found in sentences (2), (3) and (4).

Sentence (2) is similar to sentence (1) except that the (sentiment) context modifying aspect/target *price* is "high". In this case, when "high" is assigned the correct attention for the aspect *price*, the model also needs to capture the sentiment interaction between "high" and *price* in order to identify the correct sentiment polarity. This is not as easy as sentence (1) because "high" itself indicates no clear sentiment. Instead, its sentiment polarity is dependent on the given target.

Looking at sentences (3) and (4), we further see the importance of this problem and also why relying on attention mechanism alone is insufficient. In these two sentences, sentiment contexts are both "high" (i.e., same attention), but sentence (3) is negative and sentence (4) is positive simply because their target aspects are different. Therefore, focusing on improving

attention will not help in these cases. We will give a theoretical insight about this problem with MNs in Section 3.3.

In this work, we aim to solve this problem. To distinguish it from the aforementioned targeted-context detection problem as shown by sentence (1), we refer to the problem in (2), (3) and (4) as the *target-sensitive sentiment* (or target-dependent sentiment) problem, which means that the sentiment polarity of a detected/attended context word is conditioned on the target and cannot be directly inferred from the context word alone, unlike "excellent" and "ridiculous". To address this problem, we propose *target-sensitive memory networks* (TMNs), which can capture the sentiment interaction between targets and contexts. We present several approaches to implementing TMNs and experimentally evaluate their effectiveness.

3.2 Memory Network for ASC

This section describes our basic memory network for ASC, also as a background knowledge. It does not include the proposed target-sensitive sentiment solutions, which are introduced in Section 3.4. The model design follows previous studies (Sukhbaatar et al., 2015; Tang et al., 2016) except that a different attention alignment function is used (shown in Equation 4.1). Their original models will be compared in our experiments as well. The definitions of related notations are given in Table VII.

Input Representation: Given a target aspect t, an embedding matrix A is used to convert t into a vector representation, v_t ($v_t = At$). Similarly, each context word (non-aspect word in a sentence) $x_i \in \{x_1, x_2, ...x_n\}$ is also projected to the continuous space stored in memory, denoted by m_i ($m_i = Ax_i$) $\in \{m_1, m_2, ...m_n\}$. Here n is the number of words in a sentence and

t	a target word, $t \in \mathbb{R}^{V \times 1}$
v_t	target embedding of t, $v_t \in \mathbb{R}^{d \times 1}$
x_i	a context word in a sentence, $x_i \in \mathbb{R}^{V \times 1}$
m_i, c_i	input, output context embedding of word x_i, and $m_i, c_i \in \mathbb{R}^{d \times 1}$
V	number of words in vocabulary
d	vector/embedding dimension
A	input embedding matrix $A \in \mathbb{R}^{d \times V}$
C	output embedding matrix $C \in \mathbb{R}^{d \times V}$
α	attention distribution in a sentence
α_i	attention of context word i, $\alpha_i \in (0, 1)$
o	output representation, $o \in \mathbb{R}^{d \times 1}$
K	number of sentiment classes
s	sentiment score, $s \in \mathbb{R}^{K \times 1}$
y	sentiment probability

TABLE VII: Definition of notations

i is the word position/index. Both t and x_i are one-hot vectors. For an aspect expression with multiple words, its aspect representation v_t is the averaged vector of those words (Tang et al., 2016).

Attention: Attention can be obtained based on the above input representation. Specifically, an attention weight α_i for the context word x_i is computed based on the alignment function:

$$\alpha_i = softmax(v_t^T M m_i) \tag{3.1}$$

where $M \in \mathbb{R}^{d \times d}$ is the general learning matrix suggested by (Luong et al., 2015). In this manner, attention $\alpha = \{\alpha_1, \alpha_2, ..\alpha_n\}$ is represented as a vector of probabilities, indicating the weight/importance of context words towards a given target. Note that $\alpha_i \in (0, 1)$ and $\sum_i \alpha_i = 1$.

Output Representation: Another embedding matrix C is used for generating the individual (output) continuous vector c_i ($c_i = Cx_i$) for each context word x_i. A final response/output vector o is produced by summing over these vectors weighted with the attention α, i.e., $o = \sum_i \alpha_i c_i$.

Sentiment Score (or Logit): The aspect sentiment scores (also called logits) for positive, neutral, and negative classes are then calculated, where a sentiment-specific weight matrix $W \in \mathbb{R}^{K \times d}$ is used. The sentiment scores are represented in a vector $s \in \mathbb{R}^{K \times 1}$, where K is the number of (sentiment) classes, which is 3 in ASC.

$$s = W(o + v_t) \tag{3.2}$$

The final sentiment probability y is produced with a $softmax$ operation, i.e., $y = softmax(s)$.

3.3 Problem of the above Model for Target-Sensitive Sentiment

This section analyzes the problem of target-sensitive sentiment in the above model. The analysis can be generalized to many existing MNs as long as their improvements are on attention α only. We first expand the sentiment score calculation from Equation 4.2 to its individual terms:

$$\begin{aligned} s = W(o + v_t) &= W(\sum_i \alpha_i c_i + v_t) \\ &= \alpha_1 W c_1 + \alpha_2 W c_2 + ...\alpha_n W c_n + W v_t \end{aligned} \tag{3.3}$$

where "+" denotes element-wise summation. In Equation 3.3, $\alpha_i W c_i$ can be viewed as the individual sentiment logit for a context word and $W v_t$ is the sentiment logit of an aspect. They are linearly combined to determine the final sentiment score s. This can be problematic in ASC. First, an aspect word often expresses no sentiment, for example, "screen". However, if the aspect term v_t is simply removed from Equation 3.3, it also causes the problem that the model cannot handle target-dependent sentiment. For instance, the sentences (3) and (4) in Section 3.1 will then be treated as identical if their aspect words are not considered. Second, if an aspect word is considered and it directly bears some positive or negative sentiment, then when an aspect word occurs with different context words for expressing opposite sentiments, a contradiction can be resulted from them, especially in the case that the context word is a target-sensitive sentiment word. We explain it as follows.

Let us say we have two target words *price* and *resolution* (denoted as p and r). We also have two possible context words "high" and "low" (denoted as h and l). As these two sentiment words can modify both aspects, we can construct four snippets "high price", "low price", "high resolution" and "low resolution". Their sentiments are negative, positive, positive, and negative respectively. Let us set W to $\mathbb{R}^{1\times d}$ so that s becomes a 1-dimensional sentiment score indicator. $s > 0$ indicates a positive sentiment and $s < 0$ indicates a negative sentiment. Based on the above example snippets or phrases we have four corresponding inequalities: (a) $W(\alpha_h c_h + v_p) < 0$, (b) $W(\alpha_l c_l + v_p) > 0$, (c) $W(\alpha_h c_h + v_r) > 0$ and (d) $W(\alpha_l c_l + v_r) < 0$. We can drop all α terms here as they all equal to 1, i.e., they are the only context word in the snippets to attend to (the target words are not contexts). From (a) and (b) we can infer (e)

$Wc_h < -Wv_p < Wc_l$. From (c) and (d) we can infer (f) $Wc_l < -Wv_r < Wc_h$. From (e) and (f) we have (g) $Wc_h < Wc_l < Wc_h$, which is a contradiction.

This contradiction means that MNs cannot learn a set of parameters W and C to correctly classify the above four snippets/sentences at the same time. This contradiction also generalizes to real-world sentences. That is, although real-world review sentences are usually longer and contain more words, since the attention mechanism makes MNs focus on the most important sentiment context (the context with high α_i scores), the problem is essentially the same. For example, in sentences (2) and (3) in Section 1, when *price* is targeted, the main attention will be placed on "high". For MNs, these situations are nearly the same as that for classifying the snippet "high price". We will also show real examples in the experiment section.

One may then ask whether improving attention can help address the problem, as α_i can affect the final results by adjusting the sentiment effect of the context word via $\alpha_i Wc_i$. This is unlikely, if not impossible. First, notice that α_i is a scalar ranging in (0,1), which means it essentially assigns higher or lower weight to increase or decrease the sentiment effect of a context word. It cannot change the intrinsic sentiment orientation/polarity of the context, which is determined by Wc_i. For example, if Wc_i assigns the context word "high" a positive sentiment ($Wc_i > 0$), α_i will not make it negative (i.e., $\alpha_i Wc_i < 0$ cannot be achieved by changing α_i). Second, other irrelevant/unimportant context words often carry no or little sentiment information, so increasing or decreasing their weights does not help. For example, in the sentence "the price is high", adjusting the weights of context words "the" and "is" will neither help solve the problem nor be intuitive to do so.

3.4 Proposed Approaches

This section introduces six (6) alternative target-sensitive memory networks (TMNs), which all can deal with the target-sensitive sentiment problem. Each of them has its characteristics.

Non-linear Projection (NP): This is the first approach that utilizes a non-linear projection to capture the interplay between an aspect and its context. Instead of directly following the common linear combination as shown in Equation 3.3, we use a non-linear projection ($tanh$) as the replacement to calculate the aspect-specific sentiment score.

$$s = W \cdot tanh(\sum_{i} \alpha_i c_i + v_t) \tag{3.4}$$

As shown in Equation 3.4, by applying a non-linear projection over attention-weighted c_i and v_t, the context and aspect information are coupled in a way that the final sentiment score cannot be obtained by simply summing their individual contributions (compared with Equation 3.3). This technique is also intuitive in neural networks. However, notice that by using the non-linear projection (or adding more sophisticated hidden layers) over them in this way, we sacrifice some interpretability. For example, we may have difficulty in tracking how each individual context word (c_i) affects the final sentiment score s, as all context and target representations are coupled. To avoid this, we can use the following five alternative techniques.

Contextual Non-linear Projection (CNP): Despite the fact that it also uses the non-linear projection, this approach incorporates the interplay between a context word and the

given target into its (output) context representation. We thus name it Contextual Non-linear Projection (CNP).

$$s = W \sum_i \alpha_i \cdot tanh(c_i + v_t) \tag{3.5}$$

From Equation 3.5, we can see that this approach can keep the linearity of attention-weighted context aggregation while taking into account the aspect information with non-linear projection, which works in a different way compared to NP. If we define $\tilde{c}_i = tanh(c_i + v_t)$, $\tilde{c}_i$ can be viewed as the target-aware context representation of context x_i and the final sentiment score is calculated based on the aggregation of such $\tilde{c}_i$. This could be a more reasonable way to carry the aspect information rather than simply summing the aspect representation (Equation 3.3).

However, one potential disadvantage is that this setting uses the same set of vector representations (learned by embeddings C) for multiple purposes, i.e., to learn output (context) representations and to capture the interplay between contexts and aspects. This may degenerate its model performance when the computational layers in memory networks (called "hops") are deep, because too much information is required to be encoded in such cases and a sole set of vectors may fail to capture all of it.

To overcome this, we suggest the involvement of an additional new set of embeddings/vectors, which is exclusively designed for modeling the sentiment interaction between an aspect and its context. The key idea is to decouple different functioning components with different representations, but still make them work jointly. The following four techniques are based on this idea.

Interaction Term (IT): The third approach is to formulate explicit target-context sentiment interaction terms. Different from the targeted-context detection problem which is captured by attention (discussed in Section 1), here the ***target-context sentiment (TCS) interaction*** measures the sentiment-oriented interaction effect between targets and contexts, which we refer to as *TCS interaction* (or *sentiment interaction*) for short in the rest of this paper. Such sentiment interaction is captured by a new set of vectors, and we thus also call such vectors *TCS vectors*.

$$s = \sum_i \alpha_i (W_s c_i + w_I \langle d_i, d_t \rangle) \tag{3.6}$$

In Eq. Equation 3.6, $W_s \in \mathbb{R}^{K \times d}$ and $w_I \in \mathbb{R}^{K \times 1}$ are used instead of W in Eq. Equation 3.3. W_s models the direct sentiment effect from c_i while w_I works with d_i and d_t together for learning the TCS interaction. d_i and d_t are TCS vector representations of context x_i and aspect t, produced from a new embedding matrix D, i.e., $d_i = Dx_i, d_t = Dt$ ($D \in \mathbb{R}^{d \times V}$ and $d_i, d_t \in \mathbb{R}^{d \times 1}$).

Unlike input and output embeddings A and C, D is designed to capture the sentiment interaction. The vectors from D affect the final sentiment score through $w_I \langle d_i, d_t \rangle$, where w_I is a sentiment-specific vector and $\langle d_i, d_t \rangle \in \mathbb{R}$ denotes the dot product of the two TCS vectors d_i and d_t. Compared to the basic MNs, this model can better capture target-sensitive sentiment because the interactions between a context word h and different aspect words (say, p and r) can be different, i.e., $\langle d_h, d_p \rangle \neq \langle d_h, d_r \rangle$.

The key advantage is that now the sentiment effect is explicitly dependent on its target and context. For example, $\langle d_h, d_p \rangle$ can help shift the final sentiment to negative and $\langle d_h, d_r \rangle$ can help shift it to positive. Note that α is still needed to control the importance of different

contexts. In this manner, targeted-context detection (attention) and TCS interaction are jointly modeled and work together for sentiment inference. The proposed techniques introduced below also follow this core idea but with different implementations or properties. We thus will not repeat similar discussions.

Coupled Interaction (CI): This proposed technique associates the TCS interaction with an additional set of context representation. This representation is for capturing the global correlation between context and different sentiment classes.

$$s = \sum_i \alpha_i (W_s c_i + W_I \langle d_i, d_t \rangle e_i) \tag{3.7}$$

Specifically, e_i is another output representation for x_i, which is coupled with the sentiment interaction factor $\langle d_i, d_t \rangle$. For each context word x_i, e_i is generated as $e_i = Ex_i$ where $E \in \mathbb{R}^{d \times V}$ is an embedding matrix. $\langle d_i, d_t \rangle$ and e_i function together as a target-sensitive context vector and are used to produce sentiment scores with W_I ($W_I \in \mathbb{R}^{K \times d}$).

Joint Coupled Interaction (JCI): A natural variant of the above model is to replace e_i with c_i, which means to learn a joint output representation. This can also reduce the number of learning parameters and simplify the CI model.

$$s = \sum_i \alpha_i (W_s c_i + W_I \langle d_i, d_t \rangle c_i) \tag{3.8}$$

Joint Projected Interaction (JPI): This model also employs a unified output representation like JCI, but a context output vector c_i will be projected to two different continuous spaces

before sentiment score calculation. To achieve the goal, two projection matrices W_1, W_2 and the non-linear projection function *tanh* are used. The intuition is that, when we want to reduce the (embedding) parameters and still learn a joint representation, two different sentiment effects need to be separated in different vector spaces. The two sentiment effects are modeled as two terms:

$$\begin{aligned} s = & \sum_i \alpha_i W_J \tanh(W_1 c_i) \\ & + \sum_i \alpha_i W_J \langle d_i, d_t \rangle \tanh(W_2 c_i) \end{aligned} \tag{3.9}$$

where the first term can be viewed as learning target-independent sentiment effect while the second term captures the TCS interaction. A joint sentiment-specific weight matrix $W_J (W_J \in \mathbb{R}^{K \times d})$ is used to control/balance the interplay between these two effects.

Discussions: (a) In IT, CI, JCI, and JPI, their first-order terms are still needed, because not in all cases sentiment inference needs TCS interaction. For some simple examples like "the battery is good", the context word "good" simply indicates clear sentiment, which can be captured by their first-order term. However, notice that the modeling of second-order terms offers additional help in both general and target-sensitive scenarios. (b) TCS interaction can be calculated by other modeling functions. We have tried several methods and found that using the dot product $\langle d_i, d_t \rangle$ or $d_i^T W d_t$ (with a projection matrix W) generally produces good results. (c) One may ask whether we can use fewer embeddings or just use one universal embedding to replace A, C and D (the definition of D can be found in the introduction of IT). We have investigated them as well. We found that merging A and C is basically workable. But merging

D and A/C produces poor results because they essentially function with different purposes. While A and C handle targeted-context detection (attention), D captures the TCS interaction. (d) Except NP, we do not apply non-linear projection to the sentiment score layer. Although adding non-linear transformation to it may further improve model performance, the individual sentiment effect from each context will become untraceable, i.e., losing some interpretability. In order to show the effectiveness of learning TCS interaction and for analysis purpose, we do not use it in this work. But it can be flexibly added for specific tasks/analyses that do not require strong interpretability.

Loss function: The proposed models are all trained in an end-to-end manner by minimizing the cross entropy loss. Let us denote a sentence and a target aspect as x and t respectively. They appear together in a pair format (x, t) as input and all such pairs construct the dataset H. $g_{(x,t)}$ is a one-hot vector and $g^k_{(x,t)} \in \{0, 1\}$ denotes a gold sentiment label, i.e., whether (x, t) shows sentiment k. $y_{x,t}$ is the model-predicted sentiment distribution for (x, t). $y^k_{x,t}$ denotes its probability in class k. Based on them, the training loss is constructed as:

$$loss = -\sum_{(x,t)\in H}\sum_{k\in K} g^k_{(x,t)} \log y^k_{(x,t)} \tag{3.10}$$

3.5 Experiments

We perform experiments on the datasets of SemEval Task 2014 (Pontiki et al., 2014), which contain online reviews from domain *Laptop* and *Restaurant*. In these datasets, aspect sentiment

polarities are labeled. The training and test sets have also been provided. Full statistics of the datasets are given in Table XII.

Dataset	Positive		Neutral		Negative	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
Restaurant	2164	728	637	196	807	196
Laptop	994	341	464	169	870	128

TABLE VIII: Statistics of datasets

3.5.1 Candidate Models for Comparison

MN: The classic end-to-end memory network (Sukhbaatar et al., 2015).

AMN: A state-of-the-art memory network used for ASC (Tang et al., 2016). The main difference from MN is in its attention alignment function, which concatenates the distributed representations of the context and aspect, and uses an additional weight matrix for attention calculation, following the method introduced in (Bahdanau et al., 2015).

BL-MN: Our basic memory network presented in Section 3.2, which does not use the proposed techniques for capturing target-sensitive sentiments.

AE-LSTM: RNN/LSTM is another popular attention based neural model. Here we compare with a state-of-the-art attention-based LSTM for ASC, AE-LSTM (Wang et al., 2016c).

ATAE-LSTM: Another attention-based LSTM for ASC reported in (Wang et al., 2016c).

Target-sensitive Memory Networks (TMNs): The six proposed techniques, NP, CNP, IT, CI, JCI, and JPI give six target-sensitive memory networks.

Note that other non-neural network based models like SVM and neural models without attention mechanism like traditional LSTMs have been compared and reported with inferior performance in the ASC task (Dong et al., 2014; Tang et al., 2016; Wang et al., 2016c), so they are excluded from comparisons here. Also, note that non-neural models like SVMs require feature engineering to manually encode aspect information, while this work aims to improve the aspect representation learning based approaches.

3.5.2 Evaluation Measure

Since we have a three-class classification task (positive, negative and neutral) and the classes are imbalanced as shown in Table XII, we use F1-score as our evaluation measure. We report both F1-Macro over all classes and all individual class-based F1 scores. As our problem requires fine-grained sentiment interaction, the class-based F1 provides more indicative information. In addition, we report the accuracy (same as F1-Micro), as it is used in previous studies. However, we suggest using F1-score because accuracy biases towards the majority class.

3.5.3 Training Details

We use the open-domain word embeddings[1] for the initialization of word vectors. We initialize other model parameters from a uniform distribution U(-0.05, 0.05). The dimension of the word embedding and the size of the hidden layers are 300. The learning rate is set to 0.01

[1]https://github.com/mmihaltz/word2vec-GoogleNews-vectors

and the dropout rate is set to 0.1. Stochastic gradient descent is used as our optimizer. The position encoding is also used (Tang et al., 2016). We also compare the memory networks in their multiple computational layers version (i.e., multiple hops) and the number of hops is set to 3 as used in the mentioned previous studies. We implemented all models in the TensorFlow environment using same input, embedding size, dropout rate, optimizer, etc. so as to test our hypotheses, i.e., to make sure the achieved improvements do not come from elsewhere. Meanwhile, we can also report all evaluation measures discussed above[1]. 10% of the training data is used as the development set. We report the best results for all models based on their F-1 Macro scores.

3.5.3.1 Result Analysis

The classification results are shown in Table IX. Note that the candidate models are all based on classic/standard attention mechanism, i.e., without sophisticated or multiple attentions involved. We compare the 1-hop and 3-hop memory networks as two different settings. The top three F1-Macro scores are marked in bold. Based on them, we have the following observations:

1. Comparing the 1-hop memory networks (first nine rows), we see significant performance gains achieved by CNP, CI, JCI, and JPI on both datasets, where each of them has $p < 0.01$ over the strongest baseline (BL-MN) from paired t-test using F1-Macro. IT

[1]Most related studies report accuracy only.

Restaurant						Laptop					
Model	**Macro**	**Neg.**	**Neu.**	**Pos.**	**Micro**	**Model**	**Macro**	**Neg.**	**Neu.**	**Pos.**	**Micro**
MN	58.91	57.07	36.81	82.86	71.52	MN	56.16	47.06	45.81	75.63	61.91
AMN	63.82	61.76	43.56	86.15	75.68	AMN	60.01	52.67	47.89	79.48	66.14
BL-MN	64.34	61.96	45.86	85.19	75.30	BL-MN	62.89	57.16	49.51	81.99	68.90
NP	64.62	64.89	43.21	85.78	75.93	NP	62.63	56.43	49.62	81.83	68.65
CNP	65.58	62.97	47.65	86.12	75.97	CNP	**64.38**	57.92	53.23	81.98	69.62
IT	65.37	65.22	44.44	86.46	76.98	IT	63.07	57.01	50.62	81.58	68.38
CI	**66.78**	65.49	48.32	86.51	76.96	CI	63.65	57.33	52.60	81.02	68.65
JCI	**66.21**	65.74	46.23	86.65	77.16	JCI	**64.19**	58.49	53.69	80.40	68.42
JPI	**66.58**	65.44	47.60	86.71	76.96	JPI	**64.53**	58.62	51.71	83.25	70.06
AE-LSTM	66.45	64.22	49.40	85.73	76.43	AE-LSTM	62.45	55.26	50.35	81.74	68.50
ATAE-LSTM	65.41	66.19	43.34	86.71	76.61	ATAE-LSTM	59.41	55.27	42.15	80.81	67.40
MN (hops)	62.68	60.35	44.57	83.11	72.86	MN (hops)	60.61	55.59	45.94	80.29	66.61
AMN (hops)	66.46	65.57	46.64	87.16	77.27	AMN (hops)	65.16	60.00	52.56	82.91	70.38
BL-MN (hops)	65.71	63.83	46.91	86.39	76.45	BL-MN (hops)	**67.11**	63.10	54.53	83.69	72.15
NP (hops)	65.98	64.18	47.86	85.90	75.73	NP (hops)	**67.79**	63.17	56.27	83.92	72.43
CNP (hops)	66.87	65.32	49.07	86.22	76.65	CNP (hops)	64.85	58.84	53.29	82.43	70.25
IT (hops)	**68.64**	67.11	51.47	87.33	78.55	IT (hops)	66.23	61.43	53.69	83.57	71.37
CI (hops)	**68.49**	64.83	53.03	87.60	78.69	CI (hops)	66.79	61.80	55.30	83.26	71.67
JCI (hops)	**68.84**	66.28	52.06	88.19	78.79	JCI (hops)	**67.23**	61.08	57.49	83.11	71.79
JPI (hops)	67.86	66.72	49.63	87.24	77.95	JPI (hops)	65.16	59.01	54.25	82.20	70.18

TABLE IX: Results of all models on two datasets. Top three F1-Macro scores are marked in bold. The first nine models are 1-hop memory networks. The last nine models are 3-hop memory networks.

also outperforms the other baselines while NP has similar performance to BL-MN. This indicates that TCS interaction is very useful, as BL-MN and NP do not model it.

2. In the 3-hop setting, TMNs achieve much better results on Restaurant. JCI, IT, and CI achieve the best scores, outperforming the strongest baseline AMN by 2.38%, 2.18%, and 2.03%. On Laptop, BL-MN and most TMNs (except CNP and JPI) perform similarly. However, BL-MN performs poorly on Restaurant (only better than two models) while TMNs show more stable performance.

3. Comparing all TMNs, we see that JCI works the best as it always obtains the top-three scores on two datasets and in two settings. CI and JPI also perform well in most cases. IT, NP, and CNP can achieve very good scores in some cases but are less stable. We also analyzed their potential issues in Section 3.4.
4. It is important to note that these improvements are quite large because in many cases sentiment interactions may not be necessary (like sentence (1) in Section 1). The overall good results obtained by TMNs demonstrate their capability of handling both general and target-sensitive sentiments, i.e., the proposed techniques do not bring harm while capturing additional target-sensitive signals.
5. Micro-F1/accuracy is greatly affected by the majority class, as we can see the scores from Pos. and Micro are very consistent. TMNs, in fact, effectively improve the minority classes, which are reflected in Neg. and Neu., for example, JCI improves BL-MN by 3.78% in Neg. on Restaurant. This indicates their usefulness of capturing fine-grained sentiment signals. We will give qualitative examples in next section to show their modeling superiority for identifying target-sensitive sentiments.

Restaurant						**Laptop**					
Model	**Macro**	**Neg.**	**Neu.**	**Pos.**	**Micro**	**Model**	**Macro**	**Neg.**	**Neu.**	**Pos.**	**Micro**
TRMN	69.00	68.66	50.66	87.70	78.86	TRMN	68.18	62.63	57.37	84.30	72.92
RMN	67.48	66.48	49.11	86.85	77.14	RMN	67.17	62.65	55.31	83.55	72.07

TABLE X: Results with Recurrent Attention

Record 1				**Record 2**			
Text	Price was higher when purchased on MAC..			**Text**	(MacBook) Air has higher resolution..		
Target	*Price*	**Sentiment**	*Negative*	**Target**	*Resolution*	**Sentiment**	*Positive*
Result	**Sentiment Logits on context "higher"**			**Result**	**Sentiment Logits on context "higher"**		
TMN	Negative	Neutral	Positive	**TMN**	Negative	Neutral	Positive
	0.2663 (Correct)	-0.2604	-0.0282		-0.4729	-0.3949	**0.9041 (Correct)**
MN	Negative	Neutral	Positive	**MN**	Negative	Neutral	Positive
	0.3641 (Correct)	-0.3275	-0.0750		**0.2562 (Wrong)**	-0.2305	- 0.0528

TABLE XI: Sample Records and Model Comparison between MN and TMN

Integration with Improved Attention: As discussed, the goal of this work is not for learning better attention but addressing the target-sensitive sentiment. In fact, solely improving attention does not solve our problem (see Sections 3.1 and 3.3). However, better attention can certainly help achieve an overall better performance for the ASC task, as it makes the targeted-context detection more accurate. Here we integrate our proposed technique JCI with a state-of-the-art sophisticated attention mechanism, namely, the recurrent attention framework, which involves multiple attentions learned iteratively (Kumar et al., 2016; Chen et al., 2017). We name our model with this integration as Target-sensitive Recurrent-attention Memory Network (TRMN) and the basic memory network with the recurrent attention as Recurrent-attention Memory Network (RMN). Their results are given in Table X. TRMN achieves significant performance gain with $p < 0.05$ in paired t-test.

3.5.4 Effect of TCS Interaction for Identifying Target-Sensitive Sentiment

We now give some real examples to show the effectiveness of modeling TCS interaction for identifying target-sensitive sentiments, by comparing a regular MN and a TMN. Specifically, BL-MN and JPI are used. Other MNs/TMNs have similar performances to BL-MN/JPI qual-

itatively, so we do not list all of them here. For BL-MN and JPI, their sentiment scores of a single context word i are calculated by $\alpha_i W c_i$ (from Equation 3.3) and $\alpha_i W_J tanh(W_1 c_i) + \alpha_i W_J \langle d_i, d_t \rangle tanh(W_2 c_i)$ (from Equation 3.9), each of which results in a 3-dimensional vector.

Illustrative Examples: Table XI shows two records in Laptop. In record 1, to identify the sentiment of target *price* in the presented sentence, the sentiment interaction between the context word "higher" and the target word *price* is the key. The specific sentiment scores of the word "higher" towards negative, neutral and positive classes in both models are reported. We can see both models accurately assign the highest sentiment scores to the negative class. We also observe that in MN the negative score (0.3641) in the 3-dimension vector $\{0.3641, -0.3275, -0.0750\}$ calculated by $\alpha_i W c_i$ is greater than neutral (-0.3275) and positive (-0.0750) scores. Notice that α_i is always positive (ranging in $(0, 1)$), so it can be inferred that the first value in vector $W c_i$ is greater than the other two values. Here c_i denotes the vector representation of "higher" so we use c_{higher} to highlight it and we have $\{W c_{higher}\}^{Negative} > \{W c_{higher}\}^{Neutral/Positive}$ as an inference.

In record 2, the target is *resolution* and its sentiment is positive in the presented sentence. Although we have the same context word "higher", different from record 1, it requires a positive sentiment interaction with the current target. Looking at the results, we see TMN assigns the highest sentiment score of word "higher" to positive class correctly, whereas MN assigns it to negative class. This error is expected if we consider the above inference $\{W c_{higher}\}^{Negative} > \{W c_{higher}\}^{Neutral/Positive}$ in MN. The cause of this unavoidable error is that $W c_i$ is not conditioned on the target. In contrast, $W_J \langle d_i, \cdot d_t \rangle tanh(W_2 c_i)$ can change

the sentiment polarity with the aspect vector d_t encoded. Other TMNs also achieve it (like $W_I\langle d_i, d_t\rangle c_i$ in JCI).

One may notice that the aspect information (v_t) is actually also considered in the form of $\alpha_i W c_i + W v_t$ in MNs and wonder whether $W v_t$ may help address the problem given different v_t. Let us assume it helps, which means in the above example an MN makes $W v_{resolution}$ favor the positive class and $W v_{price}$ favor the negative class. But then we will have trouble when the context word is "lower", where it requires $W v_{resolution}$ to favor the negative class and $W v_{price}$ to favor the positive class. This contradiction reflects the theoretical problem discussed in Section 3.

Other Examples: We also found other interesting target-sensitive sentiment expressions like "*large bill*" and "*large portion*", "*small tip*" and "*small portion*" from Restaurant. Notice that TMNs can also improve the neutral sentiment (see Table IX). For instance, TMN generates a sentiment score vector of the context "over" for target aspect *price*: {**0.1373**, 0.0066, -0.1433} (negative) and for target aspect *dinner*: {0.0496, **0.0591**, -0.1128} (neutral) accurately. But MN produces both negative scores {**0.0069**, 0.0025, -0.0090} (negative) and {**0.0078**, 0.0028, -0.0102} (negative) for the two different targets. The latter one in MN is incorrect.

3.6 Related Work

Aspect sentiment classification (ASC) (Hu and Liu, 2004a), which is different from document or sentence level sentiment classification (Pang et al., 2002; Kim, 2014; Yang et al., 2016), has recently been tackled by neural networks with promising results (Dong et al., 2014; Nguyen and Shirai, 2015) (also see the survey (Zhang et al., 2018)). Later on, the seminal work of using

attention mechanism for neural machine translation (Bahdanau et al., 2015) popularized the application of the attention mechanism in many NLP tasks (Hermann et al., 2015; Cho et al., 2015; Luong et al., 2015), including ASC.

Memory networks (MNs) (Weston et al., 2015; Sukhbaatar et al., 2015) are a type of neural models that involve such attention mechanisms (Bahdanau et al., 2015), and they can be applied to ASC. (Tang et al., 2016) proposed an MN variant to ASC and achieved the state-of-the-art performance. Another common neural model using attention mechanism is the RNN/LSTM (Wang et al., 2016c).

As discussed in Section 3.1, the attention mechanism is suitable for ASC because it effectively addresses the targeted-context detection problem. Along this direction, researchers have studied more sophisticated attentions to further help the ASC task (Chen et al., 2017; Ma et al., 2017; Liu and Zhang, 2017). (Chen et al., 2017) proposed to use a recurrent attention mechanism. (Ma et al., 2017) used multiple sets of attentions, one for modeling the attention of aspect words and one for modeling the attention of context words. (Liu and Zhang, 2017) also used multiple sets of attentions, one obtained from the left context and one obtained from the right context of a given target. Notice that our work does not lie in this direction. Our goal is to solve the target-sensitive sentiment and to capture the TCS interaction, which is a different problem. This direction is also finer-grained, and none of the above works addresses this problem. Certainly, both directions can improve the ASC task. We will also show in our experiments that our work can be integrated with an improved attention mechanism.

To the best of our knowledge, none of the existing studies addresses the target-sensitive sentiment problem in ASC under the purely data-driven and supervised learning setting. Other concepts like sentiment shifter (Polanyi and Zaenen, 2006) and sentiment composition (Moilanen and Pulman, 2007; Choi and Cardie, 2008; Socher et al., 2013) are also related, but they are not learned automatically and require rule/patterns or external resources (Liu, 2012). Note that our approaches do not rely on handcrafted patterns (Ding et al., 2008; Wu and Wen, 2010), manually compiled sentiment constraints and review ratings (Lu et al., 2011), or parse trees (Socher et al., 2013).

3.7 Summary

In this work, we first introduced the target-sensitive sentiment problem in ASC. After that, we discussed the basic memory network for ASC and analyzed the reason why it is incapable of capturing such sentiment from a theoretical perspective. We then presented six techniques to construct target-sensitive memory networks. Finally, we reported the experimental results quantitatively and qualitatively to show their effectiveness.

Since ASC is a fine-grained and complex task, there are many other directions that can be further explored, like handling sentiment negation, better embedding for multi-word phrase, analyzing sentiment composition, and learning better attention. We believe all these can help improve the ASC task. The work presented in this paper lies in the direction of addressing target-sensitive sentiment, and we have demonstrated the usefulness of capturing this signal. We believe that there will be more effective solutions coming in the near future.

CHAPTER 4

LIFELONG LEARNING MEMORY NETWORK FOR TARGET-BASED SENTIMENT CLASSIFICATION

(This chapter includes and expands on my paper previously published in *Shuai Wang, Guangyi Lv, Sahisnu Mazumder, Geli Fei, and Bing Liu. "Lifelong Learning Memory Networks for Aspect Sentiment Classification". In* ***BigData 2018****.*)

4.1 Introduction

In this chapter, we introduce how to use the lifelong machine learning to address the aspect sentiment classification task, so as to achieve stable and satisfactory model performance even when the labeled data in one specific domain is small. As discussed in Chapter 1, considering a large amount of big unlabeled data is available on the Web, we believe a model that can automatically learn and incorporate knowledge from other domains would be more desirable. Here we proposed a novel lifelong learning approach specifically for the aspect sentiment classification and also a new model named lifelong learning memory network.

Aspect sentiment classification (ASC), also known as aspect-based sentiment classification, is a fundamental task in sentiment analysis (Liu, 2012). Given a sentence and an aspect discussed in the sentence, it aims to identify the sentiment polarity on the aspect (i.e., aspect sentiment). More specifically, it is to determine whether the sentence conveys a positive, negative or neutral aspect sentiment. For instance, in the sentence "*clear voice but the screen is*

Clear voice but the screen is scratched.
Aspect: voice, Sentiment: Positive
Aspect: screen, Sentiment: Negative

The player has had problems since the first day.
Aspect: player, Sentiment: Negative

The battery lasts very long when playing music.
Aspect: battery, Sentiment: Positive
Aspect: music, Sentiment: Neutral

Figure 2: Sample data with aspects and aspect-sentiment labels.

scratched", the sentiment polarity on aspect *voice* is positive while the one on aspect *screen* is negative. Note that aspects are also referred to as opinion targets (or *targets*) in the literature, which are usually product features/attributes. We thus use term *aspect* and *target* interchangeably in this article. In practice, aspects are either given by the user or automatically extracted using aspect extraction techniques (Liu, 2012). In this work, we assume the aspects are given and focus only on the classification problem (Wang et al., 2016c; Tang et al., 2016; Wang et al., 2018b).

To address ASC, there are two main approaches, namely, lexicon-based and supervised learning. We will discuss related works in Section 4.6. This work lies in the supervised learning direction, which is data-driven and domain-specific. Specifically, a machine learning (ML) based classifier will be trained to capture sentiment features towards aspects, with aspect-based sentiment (or *aspect-sentiment*) labels provided. Examples are shown in Figure 2. However, unlike document-level or sentence-level classification, which is to estimate an overall sentiment

polarity for an entire document/review or a single sentence, building an ML-based classifier for aspect-level sentiment analysis is somewhat tricky, because a classifier needs to consider and encode aspect information. This requirement is very important. Recall the aforementioned review sentence, where different sentiments would be inferred towards different aspects, i.e., positive on aspect *voice* but negative on aspect *screen*. Failure to encode such aspect information will be problematic for ASC. To involve aspect information, earlier studies relied on carefully engineered features (Jiang et al., 2011; Kiritchenko et al., 2014; Wagner et al., 2014), which require pattern designs, feature templates, or external resources.

Memory network (Weston et al., 2015; Sukhbaatar et al., 2015), a neural ML model, has recently become a better alternative for the ASC task. One key reason is that it can eliminate the sophisticated feature engineering, and meanwhile, achieve state-of-the-art results (Tang et al., 2016; Zhang et al., 2018). Its key advantages to ASC are its ability to learn aspect and context representation (in an embedding manner) and its attention mechanism (Mnih et al., 2014; Zhang et al., 2018). Let us use the same example to explain. When *voice* is the target aspect and represented in the embedding space, the context word "clear" will be assigned a higher attention weight than the word "scratched", under its attention mechanism. In contrast, when *screen* is the target aspect, more attention will be put on "scratched" instead of "clear". Next, the aspect-oriented sentiment can be inferred based on the weighted sum of the sentiment effect from its context words in the sentence (or called its *contexts*).

In spite of the suitability of the memory network for ASC, we observed that in practice, two crucial issues hinder its performance. First, the attention is sometimes wrongly placed.

For example, a model fails to identify that "scratched" is an important context word for aspect *screen* and thus gives it no or small attention weight. Second, when the attention is correctly assigned (i.e., a high weight is given to a correct context word), the sentiment of that word could be learned in a misleading polarity direction. For instance, a model may learn that the context word "scratched" to aspect *screen* is important but mistakenly regards it as a positive sentiment word (while a scratched screen should be negative) so the final sentiment prediction would still be wrong. We will show more examples regarding these two issues from our experiments in Section 4.5.

These two issues are caused by the fact that ASC is a fine-grained analysis task and requires a large amount of aspect-sentiment labeled data, but such labeled data are often scarce. Its *data scarcity* problem can be found or explained from multiple perspectives: (1) In practice, aspect sentiment annotation is a labor-intensive and time-consuming task. Some example labeled data are shown in Figure 2, from which we see such annotation requires substantial human effort and is often difficult to scale up. (2) In reality, one may have limited or small training data at hand (associated with gold aspect-sentiment labels) for a particular domain, while performing the ASC task. Suppose that *smartwatch* is a newly-released product and there is almost no large-scale labeled data, but manufacturers still want to analyze public opinions in time with (available) limited customer reviews. (3) In a real-world domain corpus, we should note that many product aspects are only discussed/covered by a small portion of the entire data. That is, an aspect or its context could be mentioned only a few times in the given corpus, even if the corpus itself is relatively big and well-annotated. In this case, we still have the scarcity problem

(at the aspect level). To sum up, from the above or other more perspectives, the statistically insufficient information can lead to the failure of capturing the correct attention or sentiment polarity of a word.

Given the above problem observation and analysis, this work aims at using big unlabeled data to help memory networks for ASC. The key idea is to make memory networks learn as humans do. We humans learn knowledge from our past experience and use the learned knowledge to guide our future learning. Likewise, we hope a memory network can accumulate aspect sentiment knowledge by itself from big (past) data and then use it to better guide its new/future task learning. Below, we exploit *lifelong machine learning* (LML, or *lifelong learning*) to realize this idea and propose a novel lifelong learning approach for the ASC task.

Here we first introduce the general concept of LML and then illustrate our specific solution for ASC. LML is a machine learning paradigm that enables an ML model to retain the past results as knowledge and utilize it to help future learning (Thrun, 1998a; Chen and Liu, 2016). In other words, a learner can continuously accumulate knowledge and use it to help a new task. With regard to ASC, we treat the classification task of each particular domain/product as a single learning task (we thus will use the term *domain* and *task* interchangeably). Specifically, at any point in time a learner has worked on N domains/tasks and is going to learn to perform the $(N+1)$th task (called new domain), it uses the **knowledge** obtained from the past N domains to help get a better classification result for the $(N+1)$th one. This idea is workable for ASC because although every domain is distinct, there is a considerable amount of aspect overlapping across domains. For example, many electronic products share the aspect *voice* and

screen. If certain knowledge is properly accumulated from the past domains and incorporated into the new domain, the issues discussed above in memory networks can be alleviated. For instance, when a learner has learned from the past domains like *Cellphone* and *Camera* that "scratched" is an important context word for *screen*, it will be less likely to assign wrong attention for *screen* in a new domain like *Laptop*.

To be concrete, we propose a new three-step lifelong learning approach to ASC. First, we design an automated aspect sentiment annotation strategy so as to make use of big (unlabeled) data from multiple domains. We call them assisting or past domains. Second, we retain aspect-specific attention and sentiment information from the classification results of the assisting/past domains, which are treated as raw knowledge. Third, we carry out knowledge mining to generate reliable knowledge for the new/current domain. Two different types of knowledge are considered, namely, Aspect-Sentiment Attention (ASA) and Context-Sentiment Effect (CSE). In order to leverage the mined knowledge, we propose a novel memory network named Lifelong Learning Memory Network (L2MN).

In summary, this paper makes the following contributions:

1. It indicates and analyzes the issues caused by the data scarcity problem of ASC while using memory networks, i.e., learning incorrect attention and sentiment orientation of context words. To address them, it suggests incorporating reliable knowledge mined from big unlabeled data into the learning process of memory networks.

2. It proposes to use the lifelong learning paradigm to realize the above idea, which helps memory networks work better and more stably on the ASC task. To our knowledge, no previous attempt has been made.
3. It designs a three-step lifelong learning approach to ASC, which can automatically mine two types of knowledge from multiple past domains, namely, Aspect-Sentiment Attention (ASA) and Context-Sentiment Effect (CSE), without human involvement.
4. It develops a novel model named lifelong learning memory network (L2MN) that can leverage the learned knowledge to new domains. Experimental results show its effectiveness on multiple real-world datasets.

4.2 ASC Memory Network

In this section, we briefly describe how a basic end-to-end memory network works for the ASC task. The primary model design follows a previous study (Tang et al., 2016). Notice that this basic model does not use the lifelong learning solution, but it can be easily integrated into our proposed lifelong learning memory network (L2MN, detailed later). So it can also be viewed as a non-lifelong learning (NLL) version of L2MN.

Input Representation and Attention: Given an aspect $a \in \mathbb{R}^V$, an embedding matrix $E \in \mathbb{R}^{d \times V}$ is used to convert it to a vector representation t ($t = Ea$), where V indicates the size of vocabulary and d is the embedding dimension. Similarly, each context word (each of the other non-aspect words in the sentence) $x_i \in \{x_1, x_2, ...x_n\}$ is also projected to the continuous space and stored in memory, denoted as m_i ($m_i = Ex_i$) $\in \{m_1, m_2, ...m_n\}$. Here n is the number of

words in a sentence and i indicates the word position. Attention is acquired based on the input representations. Specifically, an attention score α_i for the context word x_i is computed as:

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{n} \exp(e_j)}, e_i = \tanh(W_{att}[m_i; t] + b_{att}) \tag{4.1}$$

where $W_{att} \in \mathbb{R}^{1\times 2d}$ is a weight matrix and $b_{att} \in \mathbb{R}^{1\times 1}$ is a bias term. In this way, attention $\alpha = \{\alpha_1, \alpha_2, ..\alpha_n\}$ is represented as a vector of probabilities, indicating the weight/importance of different context words towards an aspect.

Output Representation and Sentiment Score/Logit: Another embedding matrix C is used for each context word x_i to generate its individual continuous vector c_i ($c_i = Cx_i$) $\in \mathbb{R}^d, C \in \mathbb{R}^{d\times V}$. An output vector o is produced by summing over the transformed vectors, each of which is weighted by its attention α_i. An aspect-based sentiment score is then calculated:

$$s = W(o + t), o = \sum_i \alpha_i c_i \tag{4.2}$$

where $W \in \mathbb{R}^{K\times d}$ is a sentiment weight matrix. The sentiment score/logit is represented as a vector $s \in \mathbb{R}^K$, where K is the number of (sentiment) classes. The final sentiment probability y is produced with a softmax operation $y = softmax(s)$.

From Sentiment Logit to Context-Sentiment Logit: Note that s is the final aspect-oriented sentiment score/logit. If we drop t out from Equation 4.2, we can factorize Equation 4.2 as the weighted sentiment contribution of each context word with $\sum_i \alpha_i W c_i$, where the contribution weight is determined by the importance of a context word to the aspect, i.e., α_i. As α_i

is for assigning the attention weight of a context word, the sentiment effect of this word can be presented as Wc_i and we refer to it as *Context-Sentiment Effect (CSE)*. We also define the $\alpha_i Wc_i$ as the *Context-Sentiment Logit/Score*, or sentiment logit/score for brevity. These terms will be used in the following sections.

4.3 Lifelong Learning Algorithm

This section presents our proposed lifelong learning algorithm. Notations are first introduced as follows. We define a collection of sentences in a new domain (indexed by i) as D_i^{TL}, where T indicates *target* and L indicates *labeled*. This means the sentences in D_i^{TL} have real labels (aspects and sentiments, annotated by humans). The annotated aspects and sentiments for those sentences are denoted as A_i^{TL} and S_i^{TL}. In addition, we define a collection of sentences in a past/assisting domain (indexed by j) as D_j^{PU}, where P means *past* and U means *unlabeled*. We thus have two corpora $\boldsymbol{D^{TL}} = \cup_i D_i^{TL}$, $i \in \{1, 2, ..g\}$, and $\boldsymbol{D^{PU}} = \cup_j D_j^{PU}$, $j \in \{1, 2, ..l\}$, where g and l denote the number of domains in these two corpora. Note that l is usually much larger than g because it is much easier to collect an unlabeled dataset for one domain rather than annotating detailed aspect sentiment for one domain. Associated with $\boldsymbol{D^{TL}}$, we have a collection of aspects $\boldsymbol{A^{TL}} = \cup_i A_i^{TL}$ and sentiment labels $\boldsymbol{S^{TL}} = \cup_i S_i^{TL}$. They are the input for our lifelong learning algorithm as shown in Alg. 1. From an overview perspective, Alg. 1 works in a three-step manner: first, assigning the aspect sentiment labels automatically for the big (unlabeled) data; second, building memory network classifiers and retaining (raw) knowledge; third, knowledge mining and utilization, where a newly-designed lifelong learning

memory network will be introduced for integrating the learned knowledge into its learning process.

Algorithm 1 Overview of Lifelong Learning Algorithm

Input: $\boldsymbol{D^{TL}}, \boldsymbol{D^{PU}}, \boldsymbol{A^{TL}}, \boldsymbol{S^{TL}}$

1: $\boldsymbol{D^{PL}}, \boldsymbol{A^{PL}}, \boldsymbol{S^{PL}} \leftarrow$ AutoLabelingFull($\boldsymbol{D^{PU}}$)
2: or
3: $\boldsymbol{D^{PL}}, \boldsymbol{A^{PL}}, \boldsymbol{S^{PL}} \leftarrow$ AutoLabelingLite($\boldsymbol{D^{PU}}, \boldsymbol{A^{TL}}$)
4: $\boldsymbol{RK} \leftarrow \varnothing$
5: **for** $D_j^{PL} \in \boldsymbol{D^{PL}}$ **do**
6: $\quad RK_j^{PL} \leftarrow$ L2MN($D_j^{PL}, A_j^{PL}, S_j^{PL}, NLL\_MODE$)
7: $\quad \boldsymbol{RK} \leftarrow \boldsymbol{RK} \cup RK_j^{PL}$
8: **end for**
9: **for** $D_i^{TL} \in \boldsymbol{D^{TL}}$ **do**
10: $\quad ASA_i^T, CSE_i^T \leftarrow$KnowMining ($\boldsymbol{RK}, D_i^{TL}, A_i^{TL}$)
11: $\quad$ L2MN($D_i^{TL}, A_i^{TL}, S_i^{TL}, ASA_i^T, CSE_i^T, LL\_MODE$)
12: **end for**

Step 1: Automatic Machine Labeling (lines 1-3) Note that initially no aspect or sentiment labels are available for input $\boldsymbol{D^{PU}}$. In order to make use of these unlabeled data, we design an automatic aspect sentiment labeling strategy that does not need human intervention. We refer to it as *auto-labeling*. Its idea is quite intuitive. That is, although an online review itself does not provide explicit aspect-level labels, it often contains/shows document-level rating to indicate its overall opinion. According to the theory of sentiment consistency (Abelson, 1983)

that the mentioned aspects should have consistent or similar sentiment orientation as shown by the whole review (Liu, 2012), aspect-based sentiment can be inferred to a great extent.

Specifically, while using Amazon review data[1] whose rating scores range from 1 to 5, we regard reviews with the rating of 5 (strongly positive) as reliable positive reviews and assume that opinions about the aspects discussed in each such review are also positive. Likewise, we deem reviews with the rating of 2 or 1 (strongly negative) as reliable negative reviews and consider the opinions on the aspects mentioned in each such review as negative. Certainly, this assumption may not always hold well and the resulting aspect-based sentiment labels are likely to contain noises. However, notice that we will have the following learning steps to mine reliable knowledge, instead of directly using the raw results generated from the past/assisting domains (for helping a new domain). Also, even with these (likely) noisy labels, our lifelong learning algorithm can produce reasonably good results, which will be shown in our experiments.

There are two possible ways to generate auto-labels, which are presented in lines 1 and 3. The *AutoLabelingFull* function in line 1 is to extract all aspects mentioned in D^{PU} by using an unsupervised aspect extraction approach (Liu, 2012) while the *AutoLabelingLite* function is a relaxed version that only focuses on the target aspects in $\boldsymbol{A^{TL}}$. *AutoLabelingLite* is more efficient as we only need to match and keep the sentences containing the target aspects. We use *AutoLabelingLite* for our experiments. As a result, we have auto-labeled sentences for all

---

[1]Rating ranges can vary from different sites. In such cases, reviews with the highest and lowest scores from one site are used to obtain aspect-level labels.

past domains $\boldsymbol{D^{PL}} = \cup_j D_j^{PL}, D_j^{PL} \subseteq D_j^{PU}$ along with their corresponding aspects $\boldsymbol{A^{PL}}$ and sentiments $\boldsymbol{S^{PL}}$.

Step 2: Building Classifiers and Raw Knowledge Retention (lines 4-8) For each past/assisting domain, we build a basic memory network (AMN/NLL) and retain its raw knowledge. It is important to note that here the knowledge retention does not mean that we simply save the classification results for each domain. Instead, we design proper representation to collect structured information learned by the model, which is defined as *knowledge* in this study. Specifically, two types of information will be collected, namely, attention and sentiment.

In terms of attention, we formulate its knowledge as distribution. That is, for each aspect in a domain, an aspect-sentiment attention distribution over words will be generated and retained. It basically reflects and summarizes the importance of all possible context words for a specific aspect in one domain. More concretely, the attention score of context (word) v_i for target t under sentiment r is denoted as $\alpha_{v_i,t,r}$, which is the sum over its attention divided by its total number of occurrences. It is calculated as:

$$\alpha_{v_i,t,r} = \begin{cases} 0, \sum_{q}^{N_D} \sum_{p}^{W_q} I(w_{q,p} = v_i) I(a_q = t) I(s_q = r) = 0 \\ \frac{\sum_{q}^{N_D} \sum_{p}^{W_q} \alpha_{q,p} I(w_{q,p}=v_i) I(a_q=t) I(s_q=r)}{\sum_{q}^{N_D} \sum_{p}^{W_q} I(w_{q,p}=v_i) I(a_q=t) I(s_q=r)}, otherwise \end{cases} \tag{4.3}$$

where N_D is the number of sentences in domain D and W_q is the number of words in sentence q. $w_{q,p}$ is the word in position q, p and v_i is the word i in the vocabulary. a_q and s_q denote the aspect and aspect-specific sentiment in the sentence q. $I()$ is an indicator function. Here the intuition is: if a (context) word is more positively or negatively correlated to an aspect, it should be assigned more attention most of the time when it co-occurs with the aspect. We thus

can collect a set of aspect-specific attention distributions $\boldsymbol{\alpha}^{(j)}$ from domain j for all aspects ($\boldsymbol{A^{PL}}$), for example, $\alpha_{a,s}^{(j)}$ is the distribution of aspect a under sentiment s in domain j.

In terms of sentiment, the context-sentiment effect is the focus of accumulation. Recall that we can factorize the overall sentiment logit to the individual sentiment contribution of each context in memory networks (discussed in Section 4.2). We thus construct the knowledge as a context-sentiment matrix $M \in \mathbb{R}^{V \times K}$, which is the dot product of weight matrix W and output embeddings C, i.e., $M = WC$, in each domain. So a value in $M_{v,k}^{(j)}$ indicates the sentiment effect of a context word v for sentiment k in domain j.

For each past domain, such structured attention and sentiment information are accumulated and added to the knowledge set $\boldsymbol{RK}$. However, what we have collected thus far is treated as raw knowledge, and it is not ready for use to help a new domain. Given the noises from auto-labels and mis-classification results, raw knowledge inevitably contains errors. To ensure the knowledge quality, we need further knowledge mining.

Step 3: Knowledge Mining and Application (lines 9-12) The knowledge mining (KnowMining) step mines *reliable knowledge* from the raw knowledge. Such reliable knowledge will then be used in building the lifelong classifier (LL-mode L2MN) for new domains. The reliable knowledge contains two parts, the Aspect-Sentiment Attention (ASA) knowledge, and Context-Sentiment Effect (CSE) knowledge, corresponding to the two types of raw knowledge discussed above.

To distill reliable knowledge, we employ the theory of *Frequent Pattern Mining* (FPM) (Agrawal et al., 1994). A frequent pattern is a set of items that appear frequently in a database of trans-

actions above a minimum frequency threshold, called *minimum support*. Each transaction is a set of items. In our case, we treat words with non-zero attention values ($\alpha_{v_i,t,r} \neq 0$) as items and regard a set of words in one attention distribution as one transaction. As we have accumulated a number of attention distributions (i.e., transactions) towards aspects from the past domains, FPM can filter many errors (e.g., wrong words with high aspect-sentiment attention values) that happen only in few domains. They are infrequent patterns and will be filtered based on the minimum support. In other words, the remaining frequent patterns are regarded as reliable.

This conventional data mining technique (i.e., FPM) turns out to be highly effective, because its rationale aligns well with sentiment analysis. For example, if "scratched" is assigned with negative attention towards the aspect *screen* frequently in many domains like *Cellphone, Laptop, and Camera*, we would have more confidence that "scratched" is highly correlated to aspect *screen* (negatively).

With the infrequent items/words removed, we have *denoised* knowledge. We then calculate the distributional values from the denoised knowledge for all aspects. Specifically, we average the distribution values $\alpha^{(j)}_{v'_i,t,r}$ learned from past domains, where v'_i stands for a frequent word under aspect t and sentiment r, to obtain a final set of (denoised) aspect-sentiment attention distribution $\{\alpha^{(i)}_{t_1,r}, \alpha^{(i)}_{t_2,r}, ...\}$ for a new/target domain i.

The above process results in the ASA^T_i. We also acquire CSE^T_i from the raw sentiment effect knowledge M in a similar way using FPM, to filter the words with high context-sentiment

values but appearing infrequently across domains (i.e., likely noises). They will be stored in a knowledge base (KB) and used in a new domain as prior knowledge.

4.4 lifelong learning memory network (L2MN)

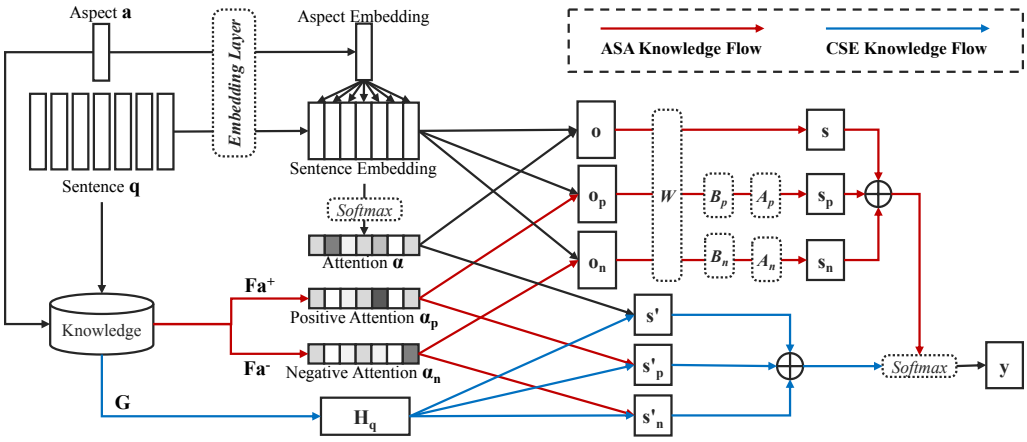


Figure 3: Lifelong Learning Memory Network (L2MN).

Here we present our proposed lifelong learning memory network (L2MN), which can leverage the learned prior knowledge to a new domain. Its model architecture is presented in Figure 3. Recall that t is the vector representation of aspect a. m_q and c_q are the input and output representation of sentence q where $m_q, c_q \in \mathbb{R}^{d \times W_q}$. W_q denotes the number of words in q. α, o, s, and y are the attention, output vector, sentiment score, and class distribution respectively as introduced in Section 4.2. Without considering other factors, they construct a basic memory

network. In other words, if we use no knowledge, L2MN can reduce to its NLL mode, i.e., the basic model AMN.

The ASA knowledge is incorporated into L2MN as two sets of knowledge-driven attention. To be concrete, given an aspect a, two types of aspect-sentiment attention distribution can be extracted from KB (with one-time creation effort) for the current domain, namely, aspect-positive attention distribution F_a^+ and aspect-negative attention distribution F_a^-. Next, the words in a sentence of current domain will be assigned with the prior aspect-positive and aspect-negative attentions. That is, additional positive attention α_p and negative attention α_n are produced for current sentence q. In this way, L2MN can utilize the accumulated attention knowledge from the past domains and provide properly estimated attention values to the context words in the sentence q of the current domain. Following the aforementioned example in Section 4.1, the negative-attention of "scratched" towards aspect *screen* can be encoded here as a type of prior information. So even if the provided data in the current domain are statistically insufficient to learn such attention ("scratch" for *screen*), this attention can still be possibly indicated by α_n from the self-accumulated ASA knowledge in L2MN.

With the involvement of α_p and α_n, L2MN moves forward to produce output representations o_p and o_n. Based on them and W, two additional sentiment scores s_p and s_n can be inferred. However, different from generating s (see Equation 4.2), two additional vectors A_p and B_p are used here for producing s_p, and two additional vectors A_n, and B_n are used for creating s_n. $A_{p/n}$ is a polarity-projection vector and $B_{p/n}$ is a polarity-selection vector. We mainly explain how A_n and B_n work below as A_p and B_p work similarly. In a binary classification case (K=2),

$y = [1, 0]$ denotes the negative class and $y = [0, 1]$ denotes the positive class. The sentiment output s_p and s_n are calculated by:

$$\begin{aligned} s_p &= A_p B_p^T W o_p, A_p = [-1, 1]^T, B_p = [0, 1]^T \\ s_n &= A_n B_n^T W o_n, A_n = [1, -1]^T, B_n = [1, 0]^T \end{aligned} \tag{4.4}$$

where $o_p = \sum_i \alpha_{p,i} c_i$ and $o_n = \sum_i \alpha_{n,i} c_i$. Here B_n pinpoints the negative-sentiment effect and A_n promotes it towards the negative class and demotes it towards the positive class. That also explains why B_n is called polarity-selection vector and A_n is called polarity-projection vector. In the three-class classification case (K=3), we will have $A_n = [1, -1, -1]$ and $B_n = [1, 0, 0]$ (for negative sentiment), where the negative, neutral, and positive classes are denoted as [1, 0 ,0], [0, 1 ,0], and [0, 0 ,1] respectively.

The CSE knowledge is incorporated as a context-sentiment matrix $G \in \mathbb{R}^{K \times V}$, where V is the vocabulary size in the current domain. Note G is derived from $M(s)$ (context-sentiment matrices from past domains) with knowledge mining and vocabulary mapping, i.e., only the reliable knowledge and the words occurring in the current domain are used. It can be directly extracted from KB as well (with one-time creation effort). In regard to a particular sentence q, a sentence-specific matrix $H_q \in \mathbb{R}^{W_q \times K}$ encodes the prior sentiment effect of the context words in sentence q. Together with the attention α, another sentiment score s' will be produced, i.e., $s' = \alpha H_q$.

Furthermore, two other sentiment scores s'_p and s'_n can be added if we consider incorporating both types of knowledge simultaneously, where $s'_p = \alpha_p H_q$ and $s'_n = \alpha_n H_q$. They are used to

encode the joint aspect-context sentiment effect learned from ASE and CSE. With them jointly considered, the final sentiment score for the aspect a in sentence q is calculated as:

$$\begin{aligned} s_{joint} &= s + s_p + s_n + s' + s'_p + s'_n \\ &= Wo + Wt + A_p B_p^T W o_p + A_n B_n^T W o_n \\ &+ \alpha H_q + \alpha_p H_q + \alpha_n H_q \end{aligned} \tag{4.5}$$

The final sentiment probability y is produced with a softmax operation $y = softmax(s_{joint})$. Note that ASA knowledge and CSE knowledge are used in both training and testing stages, which enables L2MN to consider the prior and in-domain information jointly.

Learning: The L2MN model is trained in an end-to-end manner by minimizing the cross entropy loss and using stochastic gradient descent. Let us denote a sentence and a target aspect as x and t respectively. They appear together in a pair format (x, t) as input and all such pairs construct the dataset D. $g_{(x,t)}$ is a one-hot vector and $g^k_{(x,t)} \in \{0, 1\}$ denotes a gold sentiment label, i.e., whether (x, t) shows sentiment k. $y_{x,t}$ is the model-predicted sentiment distribution for input (x, t). $y^k_{x,t}$ denotes the probability of being class k. Finally, the training loss is constructed as:

$$loss = -\sum_{(x,t)\in D} \sum_{k\in K} g^k_{(x,t)} \log y^k_{(x,t)} \tag{4.6}$$

4.5 Experiments

4.5.1 Candidate Models for Comparison

The candidate models we compare can be categorized into four general groups: long short-term memory networks (LSTMs), memory networks (MNs), non-lifelong knowledge memory networks (NLKs) and lifelong learning memory networks (L2MNs). Note that while both NLKs and L2MNs are knowledge-based models, the difference is that L2MNs use the knowledge learned from our proposed lifelong learning algorithm but NLKs use other information as their (fed) knowledge. By comparing L2MNs and NLKs, we can gain an insight into the importance of knowledge mining in the lifelong learning setting.

AT-LSTM: This is a state-of-the-art LSTM/RNN based model with aspect embedding and attention modeling for ASC (Wang et al., 2016c).

ATAE-LSTM: Another LSTM based model with aspect embedding used in both the input representation and hidden layer representation (Wang et al., 2016c).

Memory Network (MN): End-to-end memory network (Sukhbaatar et al., 2015).

Memory Network Layer-wise (MNL): A multiple-hops MN (a hop means a computational layer), where the embedding matrices are typed the same across different layers.

Memory Network Adjacent (MNA): Another version of multiple-hops MN, where the output embedding of one layer is the input embedding of its next layer.

ASC Memory Network (AMN/NLL): This is a memory network particularly proposed for the ASC task following (Tang et al., 2016). It is used as our basic model without any knowledge incorporation, i.e., it can be viewed as the non-knowledge version of L2MN.

ASC Memory Network Multi-hops (AMNM): The multiple-hops version of AMN (Tang et al., 2016).

Raw-knowledge Memory Network (RKMN) : This is the first NLK model, which directly uses the raw knowledge extracted from past domains without further knowledge mining.

Lexicon-enhanced Memory Network (LexMN): A NLK model using an opinion lexicon as its knowledge. We use the opinion lexicon from (Hu and Liu, 2004b), which consists of 2007 positive and 4873 negative sentiment words. These words play the role of ASA knowledge. Since these sentiment words are not learned from the past domains, we do not know their values in the aspect-sentiment attention distribution. We thus set a constant value ranging from {0.1, 0.2, ..., 1.0} for estimation and report the best result.

Universal-knowledge Memory Network (UKMN): Instead of applying the aspect-specific sentiment knowledge, this NLK model uses a form of universal sentiment knowledge. That is, the knowledge is an aspect-independent (or universal) sentiment attention distribution, which is the average sentiment distribution (of words) over all aspects from all past domains.

Universal-domain-knowledge Memory Network (UDKMN): This model is similar to UKMN but computes universal sentiment knowledge in another way. It first collects N sets of sentiment attention distribution from N domains, each of which is the average sentiment attention distribution over all aspects in each domain. It then averages these N distributions for the final universal knowledge. Note different domains may cover different numbers of aspects. In this case, UDKMN can mitigate the impact of domain difference.

Aspect-Sentiment Attention L2MN (ASA): Our proposed lifelong learning memory net-

work using ASA knowledge.

Context-Sentiment Effect L2MN (CSE): Our proposed lifelong learning memory network using CSE knowledge.

ASA + CSE L2MN (JOINT): Our proposed lifelong learning memory network using both ASA and CSE knowledge.

4.5.2 Experimental Setup

Datasets: We use two groups of Amazon review data. The first group provides real aspect-level manual annotation of aspects and their corresponding sentiment polarities. This group of data is used for model evaluation since it contains gold labels. We also call it **Gold Data**. Specifically, four products *Camera*, *DVD Player*, *MP3*, and *Laptop* are used as four different **target domains (or target datasets)**. The first three datasets are from (Hu and Liu, 2004b), each of which is split into training and test sets by 70% and 30%. Their data sizes are also different which help to test the model generality. The fourth dataset (*Laptop*) from SemEval 2014 (Pontiki et al., 2014) is a benchmark dataset that has been used in related studies (Tang et al., 2016; Wang et al., 2016c). Its training and test sets have already been separated. Full data statistics are reported in Table XII.

The second group of data from (Chen and Liu, 2014b) consists of reviews from 50 domains (50 datasets about different electronic products), but their reviews only have document-level ratings. So we use our proposed auto-labeling strategy to create aspect-level annotations. Since they are not gold labels, the data are not used for evaluation. However, they are still split into two sets as training and validation sets, so as to track the model learning performance. As

Data		Positive		Neutral		Negative	
Dataset	**Size**	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Camera	649	164	61	250	113	36	25
DVD Player	828	135	60	273	124	173	63
MP3	2016	349	167	834	334	225	107
Laptop	2966	994	341	464	169	870	128

TABLE XII: Statistics of the datasets in Gold Data.

discussed, they are used as **assisting/past domains** to help a target domain. We also call them **AST Data**.

Settings: For LSTMs and MNs, the models using no knowledge, only Gold data are used. For NLKs and L2MNs (except LexMN), Gold data and AST data are used together: a target domain (from Gold data) is regarded as the new domain and the 50 assisting domains (from AST data) are treated as the past domains. We then conduct experiments on the four different target domains independently to form four sets of evaluation.

Note that when a knowledge-based model starts to process a target domain, only the target domain data and self-accumulated knowledge can be used. No additional data from past domains are available, i.e., previous data cannot be accessed. So there is no specific source domain, which is different from other settings like transfer learning. Our experimental setup follows prior research about lifelong learning (Chen and Liu, 2014b; Chen and Liu, 2016; Shu et al., 2016).

For all models, we use the same set of pre-trained word embeddings[1] learned from a Google News corpus for initialization. We randomize other model parameters from a uniform distribution U(-0.05, 0.05). The dimension of word embeddings and the size of hidden layers are 300 and the learning rate is 0.01. For the multiple-hops models, we set the hop number to 3 following the previous study (Sukhbaatar et al., 2015). For each model, its hyper-parameters are set by using the *Laptop* dataset, with 10% of its training data used as the validation set. All MN models use the location attention as suggested in (Tang et al., 2016). For FPM, we empirically set the minimum supports to 8 and 3 for positive and negative sentiment knowledge, as the positive reviews are usually much more than the negative reviews according to the real-world data distribution (at the document level). Notice that this is a general FPM setting we suggest as it can basically work well for most domains, but we also found that fine-tuning the minimum supports for the four different target datasets/domains individually could lead to better results.

We test all models with two aspect sentiment classification settings: (1) Binary classification: all models are trained and tested only using positive and negative samples. (2) Three-class classification: all models are trained and tested on the full data including positive, negative, and neutral samples.

Evaluation Measure: Since the class distribution is skewed in almost all settings on all target datasets (except the binary classification on the *DVD Player* dataset), F1 score is

---

[1]https://github.com/mmihaltz/word2vec-GoogleNews-vectors

Model Description			Camera			DVD Player			MP3		
Knowledge	**Group**	**Model**	**Mac.**	**Neg.**	**Pos.**	**Mac.**	**Neg.**	**Pos.**	**Mac.**	**Neg.**	**Pos.**
No Knowledge	LSTMs	AE-LSTM	72.98	57.89	88.06	78.75	80.30	77.19	82.53	78.26	86.80
		ATAE-LSTM	73.94	60.00	87.88	78.79	80.00	77.59	82.63	79.46	85.80
	MNs	MN	66.22	47.37	85.07	84.55	84.55	84.55	80.63	75.96	85.29
		MNL	56.96	32.43	81.48	83.74	83.61	83.87	81.31	77.42	85.20
		MNA	56.09	31.58	80.60	84.55	84.55	84.55	81.46	76.38	86.53
		AMN	72.98	57.89	88.06	81.22	82.44	80.00	82.39	77.83	86.96
		AMNM	73.94	60.00	87.88	83.69	84.62	82.76	82.46	78.05	86.88
With Knowledge	NLKs	RKMN	75.13	61.54	88.72	81.22	82.44	80.00	79.85	75.00	84.71
		LexMN	74.17	59.46	88.89	84.55	84.55	84.55	82.32	77.61	87.03
		UKMN	75.13	61.54	88.72	83.73	84.13	83.33	85.33	81.90	88.76
		UDKMN	77.20	65.00	89.39	82.93	82.93	82.93	85.64	82.13	89.15
	L2MNs	ASA	75.13	61.54	88.72	85.36	85.25	85.48	85.79	83.26	88.69
		CSE	79.84	69.77	89.92	87.79	87.39	88.19	**87.77**	**85.19**	**90.36**
		JOINT	**82.19**	**73.91**	**90.48**	**88.62**	**88.71**	**88.52**	87.37	84.65	90.09

TABLE XIII: Binary classification results on first three datasets of Gold Data.

primarily used as our evaluation measure. Accuracy (Acc.) is not suggested for imbalanced datasets, as an inferior model may simply classify most samples as the majority class to achieve a high score. Specifically, both the F-Macro (averaged F1-score over all classes) and all individual class-based F1 scores will be reported. We denote F-Macro as Mac. in the following tables. The positive, neutral, and negative F1 scores are denoted as Neg., Neu., and Pos. respectively. We also provide a P&N measure to show the averaged F1 score of Pos. and Neg. for the three-class classification tasks.

4.5.3 Result Analysis

We provide quantitative results with analyses in this subsection. We first present the comprehensive results for *Camera*, *DVD Player*, and *MP3*. We then analyze *Laptop*, where we also

report accuracy as it is used in previous studies.

Binary Classification Results: We report the binary classification results in Table XIII. The highest score in each measure is marked in bold. We have the following observations:

1. L2MNs consistently perform the best on Mac., Pos., and Neg. measures. Among L2MNs, the JOINT model achieves the best results on *Camera* and *DVD Player*. CSE performs slightly better than JOINT on *DVD Player* but their scores are very close. Notice that ASA, CSE, and JOINT can all improve AMN/AMNM markedly, which shows the effectiveness of both types of knowledge.
2. Comparing L2MNs with NLKs, we can see that L2MNs work better on all datasets. Although some NLKs have competitive performances with L2MNs on some datasets, they are unstable. For example, UDKMN performs better than ASA on *Camera* and achieves the highest Mac. score among NLKs, but it works poorly on *DVD Player*. Also, note while both L2MNs and NLKs utilize knowledge to help, the superior results from L2MNs indicate the necessity of knowledge mining. In other words, simply involving extra information from (past) data like what NLKs do does not guarantee performance gains.
3. Comparing NLKs with MNs, we can see the highest scores are always from the NLKs group on all datasets, which shows the involvement of proper prior knowledge can benefit this task. LSTMs have similar performances to AMN/AMNM, but are inferior to NLKs.

Three-Class Classification Results: Results are reported in Table XIV. Note for all knowledge-based models, we so far only accumulate and incorporate positive knowledge and

	Camera					DVD Player					MP3				
Model	**Mac.**	**P&N**	**Neg.**	**Neu.**	**Pos.**	**Mac.**	**P&N**	**Neg.**	**Neu.**	**Pos.**	**Mac.**	**P&N**	**Neg.**	**Neu.**	**Pos.**
AE-LSTM	59.87	52.99	47.06	73.63	58.91	63.74	56.74	56.92	77.74	56.57	57.63	47.82	33.94	77.24	61.70
ATAE-LSTM	63.66	55.91	42.86	79.17	68.97	64.97	58.99	60.34	76.92	57.63	67.44	60.62	53.76	81.09	67.47
MN	64.17	56.61	41.67	79.30	71.54	62.94	57.20	59.54	74.40	54.87	65.13	60.97	55.81	73.45	66.13
MNL	59.01	50.40	38.30	76.23	62.50	64.08	60.58	62.82	71.07	58.33	65.76	61.45	57.40	74.38	65.51
MNA	61.60	53.56	42.55	77.68	64.57	64.06	59.46	62.77	73.25	56.14	65.72	61.28	57.51	74.62	65.05
AMN	67.44	59.81	48.65	82.70	70.97	67.43	60.81	66.67	**80.68**	54.95	69.22	63.27	58.38	81.11	68.17
AMNM	67.90	60.75	48.78	82.20	**72.73**	68.48	62.80	66.67	79.85	58.93	70.08	64.71	59.70	80.81	69.72
ASA	**69.07**	62.00	51.28	**83.19**	**72.73**	68.49	63.56	65.57	78.36	61.54	69.92	63.93	60.77	**81.91**	67.08
CSE	67.54	60.83	52.63	80.97	69.03	71.44	67.85	70.31	78.63	65.38	69.61	63.74	58.70	81.34	68.79
JOINT	68.59	**62.12**	**53.66**	81.51	70.59	**73.25**	**70.33**	**70.49**	79.07	**70.18**	**71.22**	**66.14**	**61.39**	81.38	**70.89**

TABLE XIV: Three-class classification results on first three datasets of Gold Data.

negative knowledge (the mining and utilization of neutral knowledge are left to future work). We can draw additional conclusions from the results:

1. L2MNs again achieve the best performance on Mac. and also on P&N. This means, even if only positive and negative knowledge are used, L2MNs can still improve the overall classification results. They generally have lower Neu. scores than AMN and AMNM, which is expected, but can attain much higher Neg. and Pos. scores. We anticipate that with proper neutral knowledge being mined and added, better results can be produced.
2. LSTMs and MNs are inferior to L2MNs and also NLKs. NLKs, whose results are omitted here due to space limit, work better than MNs but worse than L2MNs, very similar to the observations we have from binary classification. We thus do not repeat their analyses.
3. Last but not least, L2MNs generate consistently good results on datasets of different sizes. This supports the hypothesis we discussed in Section 4.1. That is, with lifelong learning,

the self-accumulated knowledge can alleviate some shortcomings of memory networks and ensure their stabler or better performance on ASC.

Laptop Results: Table XV reports the results on the *Laptop* dataset, which has been tested in previous studies but only accuracy scores were provided. Here we want to gain further insights. For consistency, we also report accuracy but will shed more light on the performance of every individual class. Additionally, we provide a multiple-hop L2MN (JOINT-3 with 3 hops) as opposed to the regular single-hop L2MN (JOINT-1). We can observe that both JOINT-3 and JOINT-1 outperform the state-of-the-art baseline models, and JOINT-3 achieves the best scores on almost all measures. We have also tried JOINT-3 on the previous three datasets but found limited improvement, probably because a single-hop version of L2MN already works quite well on smaller datasets (with the help of knowledge accumulation), i.e., the performance gains achieved from NLL to JOINT-1 are already noticeably large. This also indicates that deeper lifelong memory networks like JOINT-3 are more suitable for bigger data.

Model	**Mac.**	**Neg.**	**Neu.**	**Pos.**	**Acc.**
AE-LSTM	62.45	55.26	50.35	81.74	68.50
ATAE-LSTM	59.41	55.27	42.15	80.81	67.40
AMN	61.77	56.78	48.78	79.76	67.08
AMNM	65.62	63.23	51.37	82.25	70.86
JOINT-1	67.02	63.43	**55.70**	81.91	71.32
JOINT-3	**67.92**	**65.57**	54.48	**83.70**	**72.73**

TABLE XV: Results on laptop.

Knowledge Examples: Table XVI shows aspect-sentiment attention knowledge examples for aspects *product* and *software*. For each aspect, the top words are presented along with their attention distribution values. For *product*, words like "love", "excellent", and "amazing" have the strongest attentional correlation with positive polarity. That means, if the sentence "*I bought a new product and love it so much*" is given, L2MN can use its prior knowledge to better place the attention on "love" and assign stronger positive sentiment. On the other hand, when the sentence "*I have returned the product*" is given, L2MN is more likely to generate negative sentiment because "returned" is a word associated with strong negative attention toward *product*. Likewise, the sentence "*the software is intuitive to use*" would be better identified by L2MN as showing positive sentiment, since "intuitive" is learned/accumulated as the knowledge of strong positive attention towards aspect *software*.

Aspect-Sentiment Attention Knowledge		
Aspect	**Senti.**	**Attention Distribution**
product	Pos.	love(0.287), excellent (0.283), amazing (0.279), happy (0.263), definitely (0.228), highly (0.216) ...
	Neg.	disappointed (0.258), defective (0.237), poor (0.178), terrible (0.122), returned (0.117), waste (0.117) ...
software	Pos.	easy (0.173), intuitive (0.129), great (0.105), nice (0.097), good (0.07), simple (0.076) ...
	Neg.	horrible (0.170), bad (0.097), problem (0.087), poor (0.074), tried (0.070), barrel (0.069) ...

TABLE XVI: Knowledge examples.

4.5.4 Case Study

We present a real case that is wrongly classified by AMN/NLL but corrected by L2MN in Figure 4. Attention is shown as a heat map horizontally. Darker color means higher attention. Sentiment logits of contexts are shown vertically. A higher value indicates a stronger sentiment score. Let us first take a look at the attention captured by the two models in the same sentence "*however it has failed to deliver on quality*", where "quality" is the given aspect. The attention is shown horizontally as a heat map. With the automatically accumulated knowledge, L2MN better identifies that "failed" is an important context for "quality" and assigns it a higher attention weight, shown in darker red color compared with AMN. The sentiment logit of context denotes the sentiment score of a word towards sentiment classes (negative, neutral, and positive), as discussed in Section 4.2. With stronger attention, we can see the sentiment logit towards the negative class becomes higher.

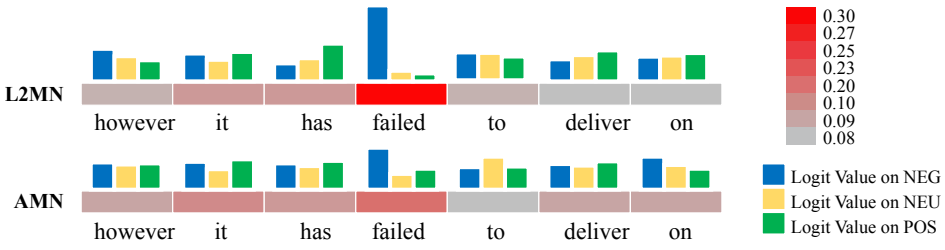


Figure 4: Attention and sentiment logit on case 1.

We present another example in Figure 5. In this example, the given aspect is "remote" in the sentence "*my other gripe is the incredibly crappy remote which is worse than other cheaper apex units*." This sentence is difficult to predict for two reasons. First, the attention becomes difficult to locate, as this sentence is relatively complicated and there are multiple sentiment-bearing context words. As we can see, AMN cannot place the attention well. In contrast, L2MN can capture attention better and find the correct sentiment context word "crappy". However, there is still another issue, even if a model detects the attention correctly. That is, a model needs to figure out the correct sentiment orientation of a context word. We found that "crappy" is a relatively infrequent word in its current domain (i.e., target dataset), which makes its sentiment polarity hard to judge. As shown in Figure 5, "crappy" has a higher positive sentiment score (shown in green color) than other two sentiments in AMN. In contrast, L2MN still works well and identifies that "crappy" is a negative sentiment word (the negative sentiment logit of "crappy" shown in blue color is greater than other two sentiments). This is attributed to the accumulated context sentiment effect (CSE) knowledge.

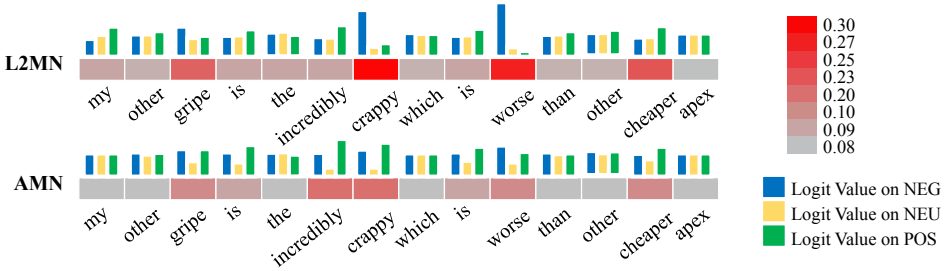


Figure 5: Attention and sentiment logit on case 2.

4.6 Related Work

Sentiment Analysis: Aspect sentiment classification (ASC) is a fundamental task in sentiment analysis (Liu, 2012). Different from document-level or sentence-level sentiment classification (Pang et al., 2002; Socher et al., 2011; Kim, 2014), ASC identifies sentiment polarity on a target aspect. The above studies (Pang et al., 2002; Socher et al., 2011; Kim, 2014) thus cannot be directly applied to ASC (they neither consider nor encode the aspect information). In the context of addressing ASC, there are two major types of approaches, the *lexicon-based* and the *supervised learning* approaches. Lexicon-based approaches use opinion lexicons and human-crafted rules (Hu and Liu, 2004b; Ding et al., 2008) to build a general classifier, while supervised learning approaches learn domain-specific classifiers and do not require opinion lexicons. Our work belongs to the latter. In regard to concrete supervised learning solutions (for ASC), early works mainly used pattern designs, feature templates, dependency relations, etc. (Kiritchenko et al., 2014; Wagner et al., 2014; Vo and Zhang, 2015), where manual feature engineering and external resources are required. Recently, some neural network approaches (Zhang et al., 2015; Wang et al., 2016c) have been applied to ASC to eliminate the sophisticated engineering process. Memory network (Sukhbaatar et al., 2015) is such a type of neural models which achieves state-of-the-art results. Our work is based on it to address its shortcoming and improve its performance on ASC.

Memory Network and Attention: A memory network (Weston et al., 2015; Sukhbaatar et al., 2015) includes an external memory and an attention mechanism, which can improve many application tasks like question answering and machine translation. Its key advantages are that

it can learn representations with its large external memory and the modeling of attention (Mnih et al., 2014). It has been recently applied to ASC (Tang et al., 2016) as discussed. Another popular attention-based model is the attention-based LSTM/RNN (Wang et al., 2016c). These two state-of-the-art solutions will be included in our experiments. There are other related studies (Ma et al., 2017; Chen et al., 2017; Liu and Zhang, 2017; Wang et al., 2018b) using memory networks or the attention mechanism but with different focuses. (Ma et al., 2017) considered learning an additional set of attention for aspect words, (Chen et al., 2017) suggested a recurrent attention mechanism, (Liu and Zhang, 2017) differentiated attention from left and right context, and (Wang et al., 2018b) provided solutions for target-sensitive sentiment. More details can be found in a survey article (Zhang et al., 2018). While their works are more about learning sophisticated attention or capturing additional signals from a single domain, they do not aim at solving the two fundamental issues caused by data scarcity as we do. Moreover, none of them considered the knowledge accumulation or lifelong learning for ASC.

Lifelong Machine Learning: Our work is also related to lifelong machine learning (LML) (Thrun, 1998a; Chen and Liu, 2014b; Mitchell et al., 2015). First, notice that LML distinguishes itself from other related paradigms like multitask learning and transfer learning. Multitask learning (Caruana, 1998) optimizes the learning of multiple related tasks at the same time, but not in a continual/lifelong learning setting. Transfer learning (Pan and Yang, 2010b) aims at using the information from a source domain to assist the learning of a target domain. It does not accumulate knowledge nor does it tackle multiple tasks continuously. Further discussions about their difference (also with other paradigms) can be found in a

survey book (Chen and Liu, 2016). Second, in terms of sentiment analysis, LML has been used to tackle aspect extraction (Liu et al., 2016; Shu et al., 2016), opinion mining (Wang et al., 2016b) and document-level sentiment classification (Chen et al., 2015a). However, their works are essentially different from ours as they are not concerned with sentiment classification at the aspect level (i.e., ASC). Their methods thus cannot solve our problem. In fact, ASC could be more challenging as a fine-grained analysis problem. To the best of our knowledge, we are the first to explore the lifelong learning of aspect-sentiment knowledge to help ASC.

4.7 Summary

Memory networks are state-of-the-art neural models for the ASC task, but two crucial issues caused by data scarcity can hinder their performance. To address them, we aimed to propose a general solution that can make memory networks work consistently better. To achieve this goal, we employed the idea of lifelong learning and designed a novel three-step lifelong learning approach for ASC. In addition, a new lifelong learning memory network (L2MN) model was developed, which can leverage the meta-mined ASA knowledge and CSE knowledge to help future tasks. Experimental results using real-world datasets demonstrated the effectiveness of our approach.

CHAPTER 5

LIFELONG ASPECT SENTIMENT TOPIC MODELING FOR MINING TARGET-SPECIFIC SENTIMENT

(This chapter includes and expands on my paper previously published in *Shuai Wang, Zhiyuan Chen, and Bing Liu. "Mining aspect-specific opinion using a holistic lifelong topic model". In* ***WWW 2016***.)

5.1 Introduction

In this Chapter, we use lifelong machine learning to mine aspect-specific opinions. Aspect-level sentiment analysis or opinion mining is a comprehensive task that aims to extract aspects, identify opinions, classify opinion polarity, and recognize general opinions and aspect-specific opinions. In this work, we refer to these four sub-tasks as *four dimensions* of aspect-level sentiment analysis. To give an example about these four dimensions, let us say a review about a cellphone product mentions "The screen is very clear and great."

1. For aspect extraction, "screen" should be extracted as an aspect.
2. For opinion identification, "clear" should be identified as an opinion word (or simply opinion). Likewise, "great" should also be identified.
3. For polarity classification, "clear" and "great" should be recognized as expressing positive opinions about the "screen".

4. For general and aspect-specific opinion separation, "clear" is an aspect-specific opinion as it indicates the clarity of the aspect *screen*. On the contrary, "great" is a general opinion as it can be used to modify many other aspects. In this paper, we call the characteristic of an opinion (word) expressing a general or aspect-specific opinion as *opinion generality*.

The first three dimensions are clearly useful as they are core problems of sentiment analysis (Liu, 2012). The fourth dimension is also important because it allows the system to discover opinion reasons, which are interesting to users (e.g., consumers and businesses) too as they almost always want to know what aspects are liked and disliked, and the reasons behind the sentiments/opinions. For example, the review sentence "The picture is bad" expresses a negative sentiment/opinion, but it does not say why the picture is bad, i.e., no reason is given, because the opinion word *bad* is a general opinion word. However, the sentence "The picture is blurry" clearly gives the reason of the negative sentiment because *blurry* is aspect-specific to the aspect *picture* indicating a specific (negative) property. Thus, opinion generality is important and is considered in our work.

Existing research has attempted to tackle some of the above dimensions of aspect-level sentiment analysis. Topic modeling has been popularly applied recently. For example, (Lin and He, 2009) proposed a joint sentiment/topic (JST) model to identify sentiment polarities of aspects. (Jo and Oh, 2011) extended the work and proposed an aspect and sentiment unification model (ASUM) which assumes that all the words in a single sentence are generated from one aspect. (Zhao et al., 2010) separated opinions and aspects by using a maximum entropy model. There are also some other related works, which will be discussed in Section 5.5. However, the

existing models do not have the capacity to model all four dimensions simultaneously. We believe that the unified joint modeling can benefit each dimension through their correlation. In this paper, we take a major step forward and present a holistic solution to jointly model all the four dimensions using a unified framework. Following the existing works, in our paper, an aspect corresponds to a topic in topic modeling.

We first propose a fine-grained topic model, called the JAST (Joint Aspect-based Sentiment Topic) model, to jointly model all four dimensions in a holistic manner. The strength of JAST is that all the component dimensions can help improve each other during the joint modeling process. The rationale here is that we can model each dimension as latent variables in a graphical model, which captures their relationship simultaneously. Experimental results show that JAST achieves significant improvements over the baseline models (Section 5.4). However, on analysis of the results to gain insights of the JAST model, we found that there was still some room for further improvement.

The main issue with JAST is that it sometimes identifies some general opinion words as aspect-specific, and vice versa. For example, opinion "nice" might sometimes be mistakenly assigned as an aspect-specific opinion for aspect *screen*. One cause of this issue is that fully-unsupervised topic models are not guaranteed to generate coherent topics that are consistent with human judgment (Chang et al., 2009). Due to the power law distribution of natural language words, most words do not co-occur with most other words (Zipf, 1932). That means, topic models, which are based on *higher-order word co-occurrences* (Heinrich, 2009), will suffer

from low word co-occurrences. As a result, some coherent aspect-specific opinions cannot be identified while they are mixed with other general opinions within the same topic.

For illustration, let us use an example from our experiments. The word "smooth" should be an aspect-specific opinion word for aspect *screen*. However, in some reviews of the domain/product like Laptop, the co-occurrence for "smooth" and *screen* may not be high enough since not every laptop is equipped with touch-screen ("smooth" is usually more associated with touch-screens). Thus, "smooth" cannot be discovered as an aspect-specific opinion word for aspect *screen* in the JAST model, even though they are in fact being mentioned together in some reviews. On the other hand, the word "nice" is mistakenly identified as an aspect-specific opinion since many occurrences of "nice" happen in the same sentences with *screen*. Due to this high co-occurrence of "nice" and *screen*, the JAST model made the mistake by treating "nice" as an aspect-specific opinion for aspect *screen*.

In order to solve the above problem, we propose a more advanced model called the LAST (Lifelong Aspect-based Sentiment Topic) model. The LAST model incorporates the idea of lifelong machine learning (LML) (Thrun, 1998b; Chen and Liu, 2014c), which has the advantage of extracting and cumulating knowledge from the past learning and using the knowledge for future learning. In the context of the combination of topic modeling with LML, it was first realized in (Chen and Liu, 2014c), which proposed the Lifelong Topic Model. However, the model is not for opinion mining and it did not jointly model the four dimensions as we do in our work. We believe that the idea of LML can be a promising direction for addressing the above issue, because a system (or a model) that has worked on many domains and retained

the discovered knowledge should be able to utilize them to help opinion mining. It is like we humans gain experience from the past and it can guide our future behaviors.

Specifically, LAST is a knowledge-based topic model that extracts and incorporates knowledge from multiple products or domains. In other words, the knowledge is automatically mined from the model results in other domains, including the discovered aspects, opinions, and aspect-opinion pairings (e.g., aspect *screen* and opinion "smooth"), and then assists the modeling of the target domain or a new coming domain. The knowledge transfer is feasible because there is a considerable amount of aspect and opinion overlapping or sharing across domains. Note that we do not use the past results directly but will perform an additional mining to discover more reliable and general knowledge to be used in the new task/domain. The rationale is that when some words appear in the same topic across many past domains, it indicates that these words are likely to be related. Following the previous example, there are other domains like Tablet and Cellphone that are likely to have touch-screens, and the words "smooth" and *screen* may co-occur very frequently in those domains. Based on such domains, we can extract the knowledge indicating "smooth" is likely to be an aspect-specific opinion to *screen*. Back in the domain Laptop, such knowledge can be leveraged to guide the model to discover the similar relationship.

In term of the mined knowledge, there are 3 types that we consider in this paper. We use another aspect *shipping* as an example for explanation (this example will be further discussed in Section 5.4):

1. Aspect-opinion pair, e.g., {shipping, quick}.

2. Aspect-aspect pair, e.g., {shipping, delivery}.

3. Opinion-opinion pair, e.g, {quick, fast}.

Each type of knowledge comes from aspects, opinions, and aspect-opinion pairings respectively (see Section 5.3.1). To leverage the extracted knowledge, we use the *generalized Pólya urn* (GPU) model, which will be illustrated in Section 5.3.2. Briefly, the key advantage of LAST is that it is able to mine more aspect-specific opinions that are coherent with the corresponding aspect as well as higher quality aspects, by extracting and leveraging prior knowledge automatically without any human invention.

In summary, this paper makes three main contributions:

1. It proposes a novel fine-grained holistic topic model, called JAST, to deal with four dimensions in aspect-level sentiment analysis, i.e., to identify aspects, opinions, opinion polarity and opinion generality simultaneously.
2. It proposes a more advanced model called LAST that can extract and leverage aspect, opinion, and their correspondence knowledge from multiple domains to further generate better aspect-specific opinions and more coherent aspects. To our knowledge, this is the first work that learns aspect, opinion, and their correspondence knowledge from the results of many domains with lifelong machine learning.
3. It conducts experiments using reviews of 50 different types of products. The experimental results show significant improvements of the proposed models over state-of-the-art baselines.

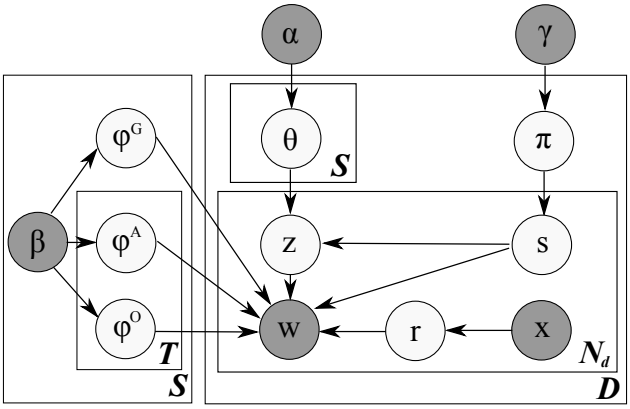


Figure 6: The graphical model of JAST

5.2 JAST Model

We now present the proposed JAST model, which jointly models aspect, opinion, polarity, and generality. The graphical model is given in Figure 6 and the notations are explained in Table XVII.

The generative process is shown as follows:

1. For each document d, we draw a sentiment distribution $\pi_d \sim Dir(\gamma)$;
2. For each sentiment s under document d, we draw a topic distribution $\theta_{d,\,s} \sim Dir(\alpha)$;
3. For each sentiment s, we draw three types of word distributions:
   (a) A general opinion word distribution under sentiment s, denoted as ${\varphi_s}^G \sim Dir(\beta_s)$;
   (b) An aspect distribution under sentiment s and topic k, which is $\varphi^A_{s,\,k} \sim Dir(\beta_s)$;
   (c) An aspect-specific opinion distribution under sentiment s and topic k, $\varphi^O_{s,\,k} \sim Dir(\beta_s)$;

S	the number of sentiment polarities
D	the number of documents
T	the number of aspect topics
V	the number of words or terms in vocabulary
N_d	the number of words in document d
s, d, z	sentiment polarity, document, topic
w, x, r	word, lexicon indicator, word type
π	multinomial distribution over sentiments
θ	multinomial distribution over topics or aspects
φ^G	multinomial distribution over general opinion words
φ^A	multinomial distribution over aspect words
φ^O	multinomial distribution over aspect-specific opinion words
α, β, γ	Dirichlet prior for θ, φ, π
w_i, z_i, s_i	word in position i (word i), topic of word i, sentiment polarity of word i
$\boldsymbol{w}$, $\boldsymbol{z}$, $\boldsymbol{s}$	all the words or terms in all documents, all the assigned topics, sentiment polarity
$\boldsymbol{z}^{-i}$, $\boldsymbol{s}^{-i}$	all the assigned topics, sentiment polarity excluding the one assigned to word i
$n_{d,\,l}^{-i}$	the number of words in document d and sentiment l except word i
$n_{d,\,k,\,l}^{-i}$	the number of words under document d and sentiment l and topic k except word i
$n_{k,\,l,\,v}^{-i}$	the number of vocabulary terms v under sentiment l and topic k except word i
$n_{l,\,v,\,c}^{-i}$	the number of words of vocabulary term v under sentiment l and word type c expect word i
$n_{k,\,l,\,v,\,c}^{-i}$	the number of words of vocabulary term v under topic k, sentiment l, word type c expect word i

TABLE XVII: Definition of notations

4. For each word w_i in document d:

   (a) choose a sentiment $s_i \sim Multi(\pi_d)$;

   (b) choose a topic $z_i \sim Multi(\theta_{d,\,s})$;

   (c) choose a word type r_i based on indicator x_i;

   (d) emit a word $w_i \sim Multi(\varphi^{r_i}_{s_i,\,z_i})$ or $w_i \sim Multi(\varphi^{r_i}_{s_i})$.

The model has three types of word distributions: φ^G_s, $\varphi^A_{s,\,k}$ and $\varphi^O_{s,\,k}$. φ^G_s indicates a general opinion word distribution under sentiment s; $\varphi^A_{s,\,k}$ and $\varphi^O_{s,\,k}$ are respectively the aspect and the aspect-specific opinion word distributions under sentiment s and topic k. Here we use the opinion sentence "The screen is very clear and great" given in Section 5.1 again to illustrate. The term "screen" is drawn from $\varphi^A_{s,\,k}$, while the term "clear" and the term "great" are selected from $\varphi^O_{s,\,k}$ and φ^G_s, and they are all under the positive sentiment s.

To model the separation of aspect and opinion, the word type r_i and indicator x_i are introduced. There are several possible approaches to construct these factors. Here we utilize the opinion lexicon, because on the one hand, the opinion lexicon can provide reliable information for polarity and also the identification of aspect and opinion terms, and on the other hand, no manual labeling is needed. So in the JAST model, the observed factor x and the hidden factor r serve for aspect and opinion identification. $x \in \{0, 1\}$ denotes whether a word exists in the opinion lexicon. If w_i appears in lexicon, then $x_i = 1$; otherwise $x_i = 0$. $r \in \{0, 1, 2\}$ indicates the word type of w_i, being an aspect, an aspect-specific opinion, or a general opinion respectively.

JAST assumes that the lexicon words are more likely to be opinion words than non-lexicon words. However, this is a soft constraint. Thus, two supporting elements λ^O and λ^A (see Equation 5.3) are designed. They are viewed as the prior information for the determination of whether a word is an aspect or opinion. Specifically λ^O controls how much we rely on the lexicon for identifying opinion words (i.e., $x = 1, r = 1$ *or* 2), while λ^A controls how much we believe a non-lexicon word is an aspect word (i.e., $x = 0, r = 0$). Although treating the words that are not in the lexicon as likely aspect terms may not always be correct, our experiments show that the model still generates rational and good results (see Section 5.4). Based on our observation, simply relying on the lexicon does not cause much problem in a fine-grained model, since the irrelevant words (or background works) are often ranked low in topics due to the naturally pairing of opinion and aspects in the opinion text.

Note that there could be other alternatives to model the identification process of aspect and opinion in JAST. Instead of fully relying on the lexicon, we can estimate the prior information λ^O and λ^A in the JAST model using supervised learning. In other words, those priors can be learned in a supervised manner without the direct auxiliary of the opinion lexicon. Following the works in (Zhao et al., 2010; Mukherjee and Liu, 2012), we also proposed a semi-supervised model which uses a Maximum Entropy classifier as the supervised component. In particular, for each word, we use the surrounding three words as the window. Inside the window, we use the parts-of-speech as features for learning. The labeled data is obtained by checking the words in each sentence with the auxiliary of the opinion lexicon, i.e., if the word appears in the lexicon, it is labeled as an opinion; otherwise, an aspect. This approach saves us from obtaining

expensive human labeled data. The advantage of this method over the simply relying on the lexicon is that it can provide more information in terms of identifying other opinion or aspect terms not appearing in the lexicon. We refer to this JAST model variant that integrates with a supervised component as JAST-S. We will see its performance in Section 5.4.

Inference: We use Gibbs Sampling (Griffiths and Steyvers, 2004), which is a standard inference technique for topic modeling. The conditional distributions are shown in Equation 5.1, Equation 5.2 and Equation 5.3 (see Table XVII for notations). In our Gibbs sampler, for each word position i in each document d, a topic k and a sentiment l are sampled first and a word type c is sampled after that.

$$
\begin{aligned}
&P(z_i = k, s_i = l | \boldsymbol{z}^{-i}, \boldsymbol{s}^{-i}, \boldsymbol{w}, \alpha, \beta, \gamma) \\
&\propto \frac{n^{-i}_{d,\,l} + \gamma_l}{\sum_{l'}^{S} (n^{-i}_{d,\,l'} + \gamma_{l'})} \times \frac{n^{-i}_{d,\,k,\,l} + \alpha_k}{\sum_{k'}^{T} (n^{-i}_{d,\,k',\,l} + \alpha_{k'})} \\
&\times \frac{n^{-i}_{k,\,l,\,w_i} + \beta_{w_i,l}}{\sum_{v}^{V} (n^{-i}_{k,\,l,\,v} + \beta_{v,\,l})}
\end{aligned}
\tag{5.1}
$$

$$
\begin{aligned}
&P(r_i = c | x_{w_i}, \boldsymbol{z}, \boldsymbol{s}, \boldsymbol{w}, \alpha, \beta, \gamma) \\
&\begin{cases}
g(c, x_{w_i}) \times \frac{n^{-i}_{l,\,w_i,\,c} + \beta_{w_i,\,l}}{\sum_{v}^{V} (n^{-i}_{l,\,v,\,c} + \beta_{v,\,l})} & c = 2 \\
g(c, x_{w_i}) \times \frac{n^{-i}_{k,\,l,\,w_i,\,c} + \beta_{w_i,\,l}}{\sum_{v}^{V} (n^{-i}_{k,\,l,\,v,\,c} + \beta_{v,\,l})} & \text{otherwise}
\end{cases}
\end{aligned}
\tag{5.2}
$$

$$g(r,x) = \begin{cases} \lambda^A & x = 0, r = 0 \\ \lambda^O & x = 1, r = 1 \; or \; 2 \\ 1 - \lambda^A & x = 0, r = 1 \; or \; 2 \\ 1 - \lambda^O & x = 1, r = 0 \end{cases} \tag{5.3}$$

5.3 LAST Model

This section introduces the LAST Model. It incorporates aspect and opinion knowledge learned/mined from multiple past domains in our proposed Gibbs sampler using the *generalized* pólya urn model.

5.3.1 LAST Learning Algorithm

As introduced in Section 5.1, we apply multi-domain knowledge to improve JAST. The overall learning algorithm is shown in Algorithm 2. LAST has the same graphical model as JAST, but the model inference is very different.

The target domain or a new coming domain, is denoted by index i, while other domains or the existing domains (past data prepared for lifelong learning) are indicated by $-i$. The review corpus $D_{(i)}$ and the corpora $\boldsymbol{D}_{(-i)}$ are the inputs.

Step 1: Aspect Matching (lines 1 - 12). This step detects similar aspects generated from existing domains to each aspect in the target domain. With the aspects identified from our proposed fine-grained model, similar aspect matching become easier to realize. Specifically, by running the JAST model, the aspects $\boldsymbol{A}$, aspect-specific opinions $\boldsymbol{O}$ and general opinions

Algorithm 2 LAST Learning Algorithm

Input: Target domain corpus $D_{(i)}$ and other domain corpora $\boldsymbol{D}_{(-i)}$

1: $\boldsymbol{A}_{(i)}, \boldsymbol{O}_{(i)}, \boldsymbol{G}_{(i)} \leftarrow$ JAST($D_{(i)}$)
2: /* Step 1. Aspect Matching */
3: **for** each domain $D_{(j)} \in \boldsymbol{D}_{(-i)}$ **do**
4: $\quad \boldsymbol{A}_{(j)}, \boldsymbol{O}_{(j)}, \boldsymbol{G}_{(j)} \leftarrow$ JAST($D_{(j)}$)
5: $\quad$ **for** each sentiment s and topic $t_{(j)}$ **do**
6: $\quad\quad t^*_{(i)} = \min_{t_{(i)}} \text{SKL}(A_{(i)s,\, t_{(i)}},\, A_{(j)s,\, t_{(j)}})$;
7: $\quad\quad$ **if** $\text{SKL}(A_{(i)s,\, t^*_{(i)}},\, A_{(j)s,\, t_{(j)}}) < \pi$ **then**
8: $\quad\quad\quad S_{s,\, t^*_{(i)}} \leftarrow S_{s,\, t^*_{(i)}} \cup \{(A_{(j)s,\, t_{(j)}}, O_{(j)s,\, t_{(j)}})\}$;
9: $\quad\quad\quad \boldsymbol{S} \leftarrow \boldsymbol{S} \cup S_{s,\, t^*_{(i)}}$;
10: $\quad\quad$ **end if**
11: $\quad$ **end for**
12: **end for**
13: /* Step 2. Knowledge Mining */
14: **for** each $S_{s,\, t_{(i)}} \in \boldsymbol{S}$ **do**
15: $\quad K_{s,\, t_{(i)}} \leftarrow$FIM($S_{s,\, t_{(i)}}$);
16: $\quad \boldsymbol{K} \leftarrow \boldsymbol{K} \cup K_{s,\, t_{(i)}}$;
17: **end for**
18: /* Step 3. Knowledge Utilization*/
19: $\boldsymbol{K}' \leftarrow$KnowledgeFiltering($\boldsymbol{K}$, $\boldsymbol{G}_{(i)}$)
20: $\boldsymbol{A}'_{(i)}, \boldsymbol{O}'_{(i)}, \boldsymbol{G}'_{(i)} \leftarrow$LAST($\boldsymbol{K}'$, $D_{(i)}$);

$\boldsymbol{G}$ for each domain are extracted. They are represented by their top words ranked by probability. Then, we measure the aspect difference using the Symmetrised KL Divergence (short in SKL) (Kawamae, 2010). Given two aspects A_x and A_y, the aspect difference is calculated with Equation 5.4 and we filter the unlikely aspects with a threshold π (line 7). That is, the aspect from other domains j that has no matched aspect in target domain i will not be used.

After all aspects generated from $\boldsymbol{D}_{(-i)}$ are processed, we obtain an aspect-opinion set $\boldsymbol{S}$. Each $S_{s,\, t_{(i)}} \in \boldsymbol{S}$ contains a set of matched aspects and their corresponding opinions.

$$SKL(A_x, A_y) = (KL(A_x, A_y) + KL(A_y, A_x))/2 \tag{5.4}$$

Step 2: Knowledge Mining (lines 13 - 17). This step mines the knowledge from each $S_{s,\, t_{(i)}}$. We apply Frequent Itemset Mining (FIM) (Agrawal et al., 1994) to find those frequently co-occurring words or terms. The reason for using FIM is that a piece of knowledge that appears only in one domain might not be reliable or transferable to other domains. Those pieces of knowledge occurring in multiple domains are more likely to be correct and useful to other domains.

With matched aspects and corresponding aspect-specific opinions, three types of aspect-opinion knowledge are mined from $S_{s,\, t_{(i)}}$: (1) aspect-opinion pair, e.g., {shipping, quick}; (2) aspect-aspect pair, e.g., {shipping, delivery}; (3) opinion-opinion pair, e.g, {quick, fast}. Each piece of knowledge basically says that the two words should belong to the same target topic under sentiment s and topic $t_{(i)}$, or its corresponding aspect and aspect-specific opinion topics. As aspect and opinion are jointly modeled in our framework, they can mutually improve the quality of each other in modeling. Consequently, all three types of knowledge lead to better topic quality. In this paper, we use frequent itemsets of length two, which give us the knowledge as word pairs. After mining, a knowledge set $\boldsymbol{K}$ (line 16) is generated.

Since all the knowledge is generated automatically from the results of unsupervised models, inevitably there are errors, e.g., {shipping, nice} and {nice, quick}. Clearly, "nice" is not specific

to *shipping*. As discussed in Section 5.1, general opinion words like "nice" may be identified as aspect-specific opinions in fully-unsupervised topic modeling. So the knowledge mining process based on the results of JAST may also suffer from it. At this stage, we keep all the knowledge $\boldsymbol{K}$ (including errors). We will deal with them in the next step.

Step 3: Knowledge Utilization (lines 18 - 20). This step uses $\boldsymbol{K}$ to improve modeling for the target domain. We first address the knowledge with errors, e.g., {shipping, nice} and {nice, quick}. Since general opinion words are also modeled in our fine-grained model, they can be used for identifying knowledge errors. Concretely, if the knowledge contains an opinion word found in $\boldsymbol{G}_{(i)}$, that knowledge will not be utilized for the target domain. For example, since "nice" is detected in our generated positive general opinion topic $G_{(i)\,positive}$, the knowledge containing "nice" will be discarded. In other words, we can handle the error by using $\boldsymbol{G}_{(i)}$ to acquire a filtered knowledge set $\boldsymbol{K'}$. The final task is to incorporate the clean knowledge $\boldsymbol{K'}$ into the LAST modeling process. We will illustrate how it works with our proposed sampler in the following sub-section.

5.3.2 Proposed Gibbs Sampler for LAST

This subsection shows the proposed Gibbs Sampler in LAST, which is different from that in JAST. To leverage the extracted opinion knowledge, we apply the *generalized* Pólya Urn model.

5.3.2.1 Pólya Urn Model

Pólya urn model (Mahmoud, 2008) is a type of statistical model with self-reinforcing property, sometimes referred as "the rich get richer". It involves with an urn, in which there are balls

of different colors. In the formulation of topic model, each color c represents each term/word $v \in V$.

In simple Pólya urn model, at each time, a ball is drawn from the urn. The color of this ball (say c) is recorded and then two balls of the color c are put back into the urn. As a result, the proportion of balls of the color c in the urn increases. The modeling of traditional topic models, such as LDA, is equivalent to the simple Pólya urn model (Mimno et al., 2011). The limitation of simple Pólya urn model is that it only involves the operation of the ball of one color at each time, i.e., only one word's proportion gets increased.

To overcome the above limitation, the *generalized Pólya urn* (GPU) model allows the procedure of putting back balls of multiple colors. In the GPU model, when a ball of a color is randomly drawn, balls of different colors can be returned to the urn according to the color matrix δ (which is usually specified by the user or by estimation). As a result, these additional balls of different colors added to the urn increase their proportions in the urn. The GPU model was first introduced in topic modeling in (Mimno et al., 2011). However, they did not use any knowledge. Later, the GPU model was utilized to incorporate knowledge in (Chen and Liu, 2014c; Chen and Liu, 2014a). In the LAST model, knowing the correlations of two words (say w_a and w_b) from the knowledge, we want to put back some balls of the color representing w_a when drawing w_b, and vice verse.

5.3.2.2 Promotion Matrix Estimation

To use the GPU model, one challenge is how to estimate the matrix δ. In LAST, the problem is how to incorporate the learned prior knowledge into the target domain with proper values,

which we call *promotion matrix estimation*. Pointwise Mutual Information (PMI), known as an useful approach to measuring word association in documents (Newman et al., 2010a), is suitable for our task. Here we only use the positive PMI values, as the mined knowledge from multi-domains implies positive semantic correlation. It is finally used to guide the knowledge utilization in LAST with a constraint factor μ (μ>0) which controls how much we believe its indicated values. Now we can compute the promotion rate $PR(w_a, w_b)$ for words w_a and w_b, with the definition given in Equation 5.5.

$$PR(w_a, w_b) = \mu \times log \frac{P(w_a, w_b)}{P(w_a)P(w_b)} \tag{5.5}$$

$$P(w) = \frac{\#D(w)}{\#D} \tag{5.6}$$

$$P(w_a, w_b) = \frac{\#D(w_a, w_b)}{\#D} \tag{5.7}$$

$P(w)$ indicates the probability of word w occurring in a random document of the target corpus, while $P(w_a, w_b)$ is the probability of co-occurrence of words w_a and w_b in a random document of the target corpus. They are estimated using Equation 5.6 and Equation 5.7 where $\#D(w)$ is the number of documents in the target corpus that contain the word w and $\#D(w_a, w_b)$ is the number of documents that contain both words w_a and w_b. $\#D$ is the total number of documents in the target corpus. We can then estimate and leverage the learned

knowledge with the promotion matrix for the target domain (Equation 5.8). Here s denotes the sentiment polarity. t is the topic while i is the domain index.

$$\delta_{s,t_{(i)},w_a,w_b} = \begin{cases} 1 & w_a = w_b \\ PR(w_a, w_b) & (w_a, w_b) \in K'_{s,\, t_{(i)}} \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

5.3.2.3 Inference

The conditional distributions for the new Gibbs sampler are given in Equation 5.9 and Equation 5.10. The $g(c, x_{w_i})$ in Equation 5.10 is from Equation 5.3. The notations are shown in Table XVII except $\delta_{l,\, k\, ,w_j,\, w_i}$, which is the matrix defined in Equation 5.8.

$$\begin{aligned} &P(z_i = k, s_i = l | \boldsymbol{z}^{-i}, \boldsymbol{s}^{-i}, \boldsymbol{w}, \alpha, \beta, \gamma) \\ &\propto \frac{n^{-i}_{d,\, l} + \gamma_l}{\sum_{l'}^{S} (n^{-i}_{d,\, l'} + \gamma_{l'})} \times \frac{n^{-i}_{d,\, k,\, l} + \alpha_k}{\sum_{k'}^{T} (n^{-i}_{d,\, k',\, l} + \alpha_{k'})} \\ &\times \frac{\sum_{w_j}^{V} \delta_{l,\, k\, ,w_j,\, w_i} \times n^{-i}_{k,\, l,\, w_i} + \beta_{w_i,\, l}}{\sum_{v}^{V} (\sum_{w_j}^{V} \delta_{l,\, k\, ,w_j,\, v} \times n^{-i}_{k,\, l,\, v} + \beta_{v,\, l})} \end{aligned} \tag{5.9}$$

$$\begin{aligned} &P(r_i = c | x_{w_i}, \boldsymbol{z}, \boldsymbol{s}, \boldsymbol{w}, \alpha, \beta, \gamma) \\ &\begin{cases} g(c, x_{w_i}) \times \frac{n^{-i}_{l,\, w_i,\, c} + \beta_{w_i,\, l}}{\sum_{v}^{V} (n^{-i}_{l,\, v,\, c} + \beta_{v,\, l})} & c = 2 \\ g(c, x_{w_i}) \times \frac{n^{-i}_{k,\, l,\, w_i,\, c} + \beta_{w_i,\, l}}{\sum_{v}^{V} (n^{-i}_{k,\, l,\, v,\, c} + \beta_{v,\, l})} & \text{otherwise} \end{cases} \end{aligned} \tag{5.10}$$

5.3.3 Discussion

One may argue that we actually do not need to go through the proposed process to mine and use prior knowledge. Instead, a simple PMI of all pairs of words over all the domains could be used, i.e., for each pair of words, if its PMI value over all the domains is higher than a certain threshold, it is treated as a piece of prior knowledge. This is a valid approach. However, this approach is inferior due to two main reasons. First, as mentioned above, most words do not co-occur with most other words due to the power law distribution in the natural language text. For example, the PMI value of words *price* and *expensive* is small as their co-occurrence is very small. As a result, we will not be able to discover their semantic correlation as a piece of knowledge. However, topic models are able to discover the pair via higher-level co-occurrences. For example, word *buy* may co-occur frequently with *price* in some documents while in some other documents, *buy* may have a high co-occurrence with *expensive*. In such cases, the transitive higher-level co-occurrences can be captured by topic models to produce topics with *price* and *expensive* together under the same topic.

Second, even if we find a pair with a high PMI value, we do not know whether co-occurrences are from a single domain or multiple domains. Frequent co-occurrences in one domain may just indicate this pair of words is specific to that domain and may not be generally applicable. It could also be due to some idiosyncrasy of the data in that domain which causes the high and possibly spurious co-occurrences.

5.4 Evaluation

5.4.1 Candidate Models for Comparison

This section evaluates the following models:

LDA (Blei et al., 2003): The classic unsupervised topic model.

ASUM (Jo and Oh, 2011): The aspect and sentiment unifications model. Since it is reported as achieving improvement over JST (Lin and He, 2009) and is the known closest work to us, it is regarded as our most important baseline. We downloaded the system from the authors' homepage.

ASUM-L: A variation of the ASUM model by applying the opinion lexicon that we use instead of the original seed words in ASUM.

JAST: Our proposed joint aspect-specific sentiment topic model, which models the identification of aspects, opinions, opinion polarity, and opinion generality simultaneously.

JAST-S: A semi-supervised variant of JAST using a Maximum Entropy classifier as the supervised component (see Section 5.2).

LAST: Our proposed lifelong aspect-based sentiment topic model. It automatically mines and leverages aspect, opinion, and their correspondence knowledge from multiple domains.

5.4.2 Experiment Setup

Datasets. We use the 50-domains online review corpus created by the authors of (Chen and Liu, 2014c). Each domain is a type of products and has 1,000 reviews. We follow their data pre-processing procedure with the standard lemmatization and stop word removal. However,

we keep all general opinion words, e.g., *good, nice, great* (while they treated them as stop words and removed them), because general opinion topics are also one of our modeling components.

Lexicon. We use the opinion lexicon[1] of (Hu and Liu, 2004).

Parameter Setting. All the models are trained using 1000 iterations with 200 burn-in periods. The sentiment number is set as $S = 2$ for extracting positive and negative opinions. The common parameters are set as $\alpha = 0.1$, $\beta = 0.01$, $\gamma = 1$ and $T = 15$ for our proposed models based on our pilot experiments. For all baseline models, we try both our proposed parameters and the ones in their original papers, and select the better result for comparison. In LAST, for simplicity, we set λ^A and λ^O to 1, which already generates good results. For learning in LAST, we empirically set π to 7.0, μ to 0.3 and set minimum support for frequent itemset mining to $max(4, 0.7\times|\boldsymbol{D_s}|)$ for aspect-opinion, $max(4, 0.3\times|\boldsymbol{D_s}|)$ for aspect-aspect and $max(4, 0.2 \times |\boldsymbol{D_s}|)$ for opinion-opinion pairs, where $|\boldsymbol{D}_s|$ is the number of domains containing matched aspects for a target aspect. The top 15 aspect words and top 15 aspect-specific opinion words are selected to represent aspects $\boldsymbol{A}$ and opinions $\boldsymbol{O}$, which is intuitive as they are the top words for the representation of their topics. These words are used for aspect matching and knowledge mining. For general opinion $\boldsymbol{G}$, the top 25 words are used for representation, which should have more words than aspect-specific opinions by nature. It is also the similar size as the general sentiment seed words used in ASUM. Note that for LAST, each domain works as

---

[1]http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html

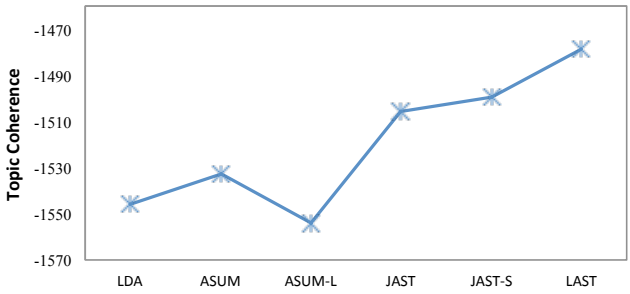


Figure 7: Average topic coherence of each model over 50 domains.

the target domain while the rest 49 domains serve as the past/existing domains used in mining prior knowledge.

5.4.3 Topic Coherence

This sub-section reports an objective evaluation based on Topic Coherence proposed in (Mimno et al., 2011). Topic models are conventionally evaluated using perplexity on held-out test data. However, as shown in (Newman et al., 2010b), perplexity is unable to reflect the real semantic coherence for individual topics. The research in (Chang et al., 2009) showed that it sometimes even contradicts human judgment. Topic Coherence is now commonly used as a better alternative for assessing topic quality, as it evaluates the coherence and interpretability of topics, which is suitable for our task, as our goal is to make the opinions, along with aspects, more coherent in individual topics.

Figure 7 shows the comparison results. A higher Topic Coherence score indicates a higher topic quality, i.e., better topic interpretation. From Figure 7, we can make the following observations.

1. Our proposed second model LAST achieves the highest topic coherence score. The knowledge from the lifelong learning mechanism greatly benefits the model in discovering higher quality opinions and aspects. Since the aspect and aspect-specific opinion become more coherent, the topic quality is naturally improved. It also shows that the proposed approach in LAST is able to deal with wrong knowledge automatically.
2. Our proposed first model JAST and its variant JAST-S are inferior to LAST, but still outperform all the baseline models, i.e., LDA, ASUM and ASUM-L. As expected, with a supervised component in the JAST-S model, it is able to better identify other potential aspects or opinion words. Note that JAST is also comparable with JAST-S, which demonstrates the reliability of the lexicon as we discussed in Section 5.2. Comparing with the baseline models (LDA, ASUM, and ASUM-L), we can clearly see that the proposed fine-grained model JAST is able to better group aspects and opinions into interpretable topics, which shows that dealing with four dimensions simultaneously benefits the modeling.
3. ASUM-L has the lowest topic coherence score. This result indicates that it is not guaranteed to achieve improvements by simply using a bigger opinion lexicon. One main reason may be that a sentence could have two words with different opinion polarities or multiple aspects/opinions, which violates the assumption made by ASUM (i.e., one sentence has only one aspect). The results show that the assumption is not suitable for more fine-grained or

aspect-specific opinion mining. ASUM and LDA do not perform as well as our proposed models. This again shows the effectiveness of our proposed models.

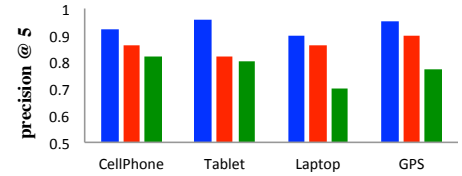
Statistical tests show that both the improvements for JAST and LAST are significant ($p < 0.001$) against the baselines using paired t-test.

5.4.4 Topic Quality Evaluation

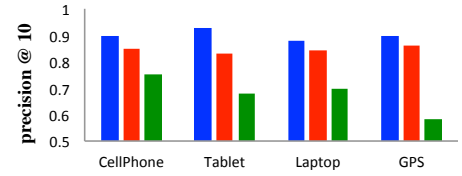
Here we analyze the results using human judgment. Two human labelers who are familiar with Amazon product reviews are asked to label the results. In the above five models, LDA does not detect opinions, and its resulting topics are also not as coherent as those of the ASUM model. ASUM-L is worse in the objective evaluation than ASUM. The result of JAST-S is similar to JAST. Thus, we primarily compare the JAST and LAST models with ASUM. Note that since ASUM does not separate aspect and sentiment words in a topic, we manually identify and extract the top opinion words appearing in its generated topics. Results from four domains (types of products) are selected for manual evaluation based on the familiarity of the annotators towards the domains.

5.4.4.1 Opinion Precision

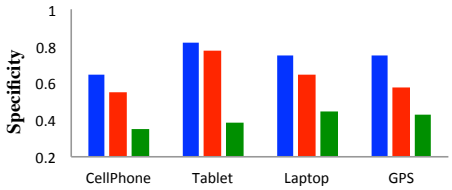
We first evaluate the precision of aspect-specific opinions. We define aspect-specific opinion precision based on the following: a correct opinion word should (a) have the correct polarity and (b) reasonably express opinion about the aspect. For example, for a negative opinion topic for aspect screen, both "fuzzy" and "bad" are correct for aspect *screen*, but "good" and "noisy" are incorrect. Note that here specific and general opinions are not distinguished (we will further evaluate them in the next sub-section).



(a) Precision@5



(b) Precision@10



(c) Specificity

Figure 8: Opinion evaluation - models in each figure from left to right are LAST, JAST and ASUM

Evaluation Measure. Since the aspect-based opinion words are generated by the topic model with ranking, we do not know the exact number of correct opinion words, a natural and commonly used metric for evaluation is *precision*@n (p@n for short), where n is a rank position. We give p@n for $n = 5$ and 10.

Topic Matching. As different models give different topic (aspect) distributions, we manually match ten best aspect topics for each domain, five positive and five negative respectively, and then compute the average opinion precision for each model.

Battery (Negative)			Shipping&Order (Positive)		
LAST	**JAST**	**ASUM**	**LAST**	**JAST**	**ASUM**
die	old	*problem*	new	free	*great*
dead	die	hot	free	*happy*	*good*
short	fail	*bad*	fast	fast	quickly
drain	*suck*	die	quick	*pleased*	*well*
fail	useless	*original*	refund	refund	*love*
old	hassle	old	promptly	*recommend*	*perfect*
hassle	*bad*	*new*	original	new	*nice*
wrong	*concern*	*long*	correct	works	*perfectly*
useless	bother	break	works	quick	new
complain	*nervous*	*hate*	accurate	promptly	fast

TABLE XVIII: Opinion words for Battery and Shipping&Order aspects. Incorrect opinion words are italicized and marked in red. Non-specific opinion words are italicized and marked in blue.

Result Analysis. 8a and 8b give the average p@5 and p@10 for each labeled domain. LAST achieves the highest precision for all domains. JAST is also better than ASUM but not as good as LAST. On average, LAST improves ASUM by 15.8% in p@5, 22.5% in p@10. JAST also improves ASUM by 8.6% and 16.8% respectively. Cohen's Kappa agreement scores for p@5 and p@10 are 0.848 and 0.804.

5.4.4.2 Opinion Specificity

We now evaluate whether the identified aspect-specific opinion words are indeed specific. After the previous sub-section, we filter out those incorrect opinion words for further evaluation. There are still two types of opinions, general and aspect-specific opinions. Two example opinion topics are shown for two aspects in Table XVIII. For example, "great" for aspect *shipping* is

not really specific but "quick" is oppositely informative. The opinion words marked in blue are general opinion words, e.g., *problem, bad*. Here we evaluate whether an opinion word is specific enough to give meaningful description about the aspect. We call it *opinion specificity*. Besides the opinion words, the top 20 aspect words of each topic are additionally provided to the annotators (for reference), so that they can better understand what the corresponding aspect should be and then identify correct aspect-specific opinion words.

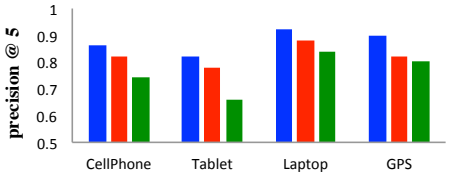
Evaluation Measure. We calculate the opinion specificity using Equation 5.11.

$$Specificity = \frac{n(specific@10)}{n(correct@10)} \tag{5.11}$$

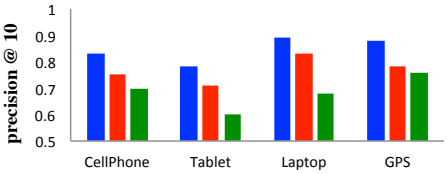
The annotators evaluate the top 10 correct opinion words (denoted as n($correct$)@10) in every topic. The count of valid aspect-specific opinion words is n($specific$)@10. If n($correct$@10) is less than 5, we do not evaluate that topic, as a very small denominator may lead to a false high value.

Results: 8c gives the results. We can see that LAST and JAST improve 34.1% and 23.8% over ASUM respectively. A lot of general opinion words with high probabilities are found in ASUM, e.g., *problem, great, good*, while the opinion words in JAST and LAST are more specific to the aspect. Cohen's Kappa agreement is 0.823.

Example Opinion Topics: Table XVIII gives the aspect-specific opinion words of two example aspects. Incorrect opinion words are italicized and marked in red. Non-specific opinion words are italicized and marked in blue. For instance, for aspect *Battery*, *new, original*, and *long* are incorrect as they are not negative aspect-specific opinion words. The words in blue



(a) Precision@5



(b) Precision@10

Figure 9: Aspect precision - models in each figure from left to right are LAST, JAST and ASUM

color like *problem, bad*, and *suck* are not aspect-specific, though correct in polarity. We can see that LAST discovers many aspect-specific and coherent opinion words in both example topics.

General Opinions. We also compute the average precision of the positive and negative general opinion words to see whether they are indeed general. The results are: $p@10 = 83.8\%$, $p@20 = 79.4\%$ for JAST and $p@10 = 85.0\%$, $p@20 = 80.0\%$ for LAST. We use more words here because the number of general opinion words is large. The polarities of top words (no filtering) are all correct. ASUM does not model general opinions.

5.4.4.3 Aspect Precision

For aspect topics, we also report *precision*@5 and *precision*@10 for the four domains. 9a and 9b give their corresponding results averaged over topics of each domain. We observe that LAST achieves dramatic improvements over ASUM. The margins of improvement of JAST over

Battery			Shipping&Order		
LAST	**JAST**	**ASUM**	**LAST**	**JAST**	**ASUM**
battery	battery	charge	order	arrive	*screen*
charge	charge	battery	receive	receive	receive
hour	life	recharge	arrive	order	arrive
life	hour	*iphone*	shipping	purchase	order
power	*device*	*sd*	ship	*expect*	*privacy*
charger	cable	*card*	today	send	cost
recharge	*phone*	receive	delivery	ship	money
night	*ipad*	replacement	usual	shipping	*monitor*
outlet	power	*purcharse*	*expect*	back	purchase
aaa	plug	*star*	*manner*	*seller*	*seller*

TABLE XIX: Example aspect words for Battery and Shipping&Order. Errors are marked in red.

ASUM are also large. LAST is the best, which demonstrates that making use of knowledge learned from past domains is very helpful. Table XIX shows the aspect words of two example topics. We can see the superior performance of LAST. Cohen's Kappa agreement is 0.811. Note that since the objective of our models is essentially for opinion mining in a holistic manner, we do not target at outperforming the existing models that are specialized in the aspect extraction task. Here the results are for showing that, while mining more coherent opinions the joint modeling process can in fact improve the aspect quality as well.

5.5 Related Work

Aspect-based opinion mining has been an important research direction (Hu and Liu, 2004b). In recent years, various researches have been conducted to perform different sub-tasks. Since

our work focuses on topic modeling, we will mainly discuss the existing related works using topic modeling.

The most related works are the joint models that model aspects and opinions. (Lin and He, 2009) proposed a joint sentiment-topic model (JST). Rather than modeling topics (or aspects) only as in LDA, JST models both opinion/sentiment and aspect as random variables. However, JST does not separate aspects and opinions, and does not tackle the opinion generality problem. Later on, (Jo and Oh, 2011) proposed a model called ASUM assuming that *one sentence is generated by one topic or aspect*, i.e., ASUM assigns all words in a sentence to the same topic. It was shown in (Jo and Oh, 2011) that the ASUM model outperformed JST. Similar to JST, ASUM does not separate opinions and aspects nor does it address the opinion generality issue. (Mukherjee and Liu, 2012) utilized some topical word seed sets as the knowledge to improve the modeling of aspects and opinions. Each seed set consists of a set of seed words for a particular topic. However, their seed sets are manually provided while our proposed method is fully automatic.

(Zhao et al., 2010) provided an approach to separating aspects and opinion words by integrating supervised learning into topic modeling. They also distinguished general and specific opinions, but they do not identify opinion polarity and their supervised component needs manually labeled data. Their supervised learning model also classifies a word as a background word or not. Some aspect and opinion terms may be lost if they are predicted as background words so we do not adopt it in our model. To address those problems, we utilize an opinion lexicon. On one hand, the opinion lexicon provides information to help identify opinion polarity. On

the other, it helps separate aspect and opinion words with reliable prior information. We do not model the background topic explicitly as we observed that in our finer-grained models, background words usually do not have high probabilities in aspect and opinion topics. Thus, they do not cause much problem. Meanwhile, aspect and opinion information will not be lost by misclassifying words to background words in this way. Recently, (Wang et al., 2015) proposed a novel unsupervised approach for aspect (words) and opinion (words) extraction based on Restricted Boltzmann Machine (Salakhutdinov and Hinton, 2009). However, apart from the opinion lexicon, it also relies on Parts-Of-Speech (POS) tagging and external Google n-gram corpus for prior information estimation, which we do not use. It also requires manual aspect-topic assignment, which we do not adopt.

Though not aimed at opinion mining, a related fine-grained model is reported in (Diao et al., 2014) for movie recommendation. It combines collaborative filtering and topic modeling. The model covers user, movie, review content, and review rating in a comprehensive manner. Since our work focuses on opinion mining, it does not include user, movie/product, or review rating information, nor is it concerned with recommendation. Thus our model is quite different.

There are also many topic models that have been used for the task of aspect extraction and categorization in, for example, (Branavan et al., 2009; Fang and Huang, 2012; Li et al., 2011; Lu et al., 2009; Yang and Cardie, 2013; Lu and Zhai, 2008; Moghaddam and Ester, 2013; Mukherjee and Liu, 2012; Sauper and Barzilay, 2013; Diao et al., 2014; Brody and Elhadad, 2010; He et al., 2011; Kim et al., 2013; Lazaridou et al., 2013; Mei et al., 2007; Titov and McDonald, 2008b; Wang et al., 2010). Although related, their focuses are very different from ours because

they do not target at full aspect-based opinion mining. Here we discuss some of the papers to indicate the type of differences. (Fang and Huang, 2012) aims to find informative sentences that are related to certain aspects. (Yang and Cardie, 2013) proposed a joint optimization framework to identify the relationship between opinion, opinion holder and opinion target. (Titov and McDonald, 2008b) uses rating in their topic model, which is not used in our case. In the context of aspect-level sentiment analysis, (Lazaridou et al., 2013) uses discourse structure to improve the performance. However, we do not consider discourse here. To address the sparsity issue of the cold start items, i.e., items that have less than 10 reviews, (Moghaddam and Ester, 2013) proposed the Factorized LDA (FLDA) model with the consideration of ratings. Again, we do not consider rating in our work.

Additionally, there are some other existing generative approaches that model cross-collection and multi-faceted (or multi-dimensional) information or topics. (Zhai et al., 2004) proposed a topic model for comparative text mining. It discovers common topics across multiple collections, and distinguishes the general cross-collection and collection-specific information under a discovered common topic. (Paul and Girju, 2009) extended the work and proposed a new cross-collection mixture model to identify cross-culture differences in blogs and forums. Despite the usage of multiple corpora or domains, these works are not for opinion mining and their models also function quite differently as our goal is not to find cross-collection (or cross-domain) commonalities and differences. On multi-faceted or multi-dimensional analysis, a two-dimensional model is reported in (Paul and Girju, 2010) that discovers different facets under one topic, e.g., extracting two different perspectives under a specific issue (topic), e.g., Israeli-Palestinian

Conflict. A k-dimensional model called factorial-LDA was proposed in (Paul and Dredze, 2012) and it improved the performance of (Paul and Girju, 2010). (Paul and Dredze, 2013) further enhanced the factorial-LDA model and adapt it to the task of summarization of drug experiences. Although, to some extent, these multiple-dimensional models are related to aspect extraction or opinion mining (if aspect and aspect-specific opinion are treated as two dimensions), they are not specialized models nor fine-grained models for opinion mining as (Lin and He, 2009; Jo and Oh, 2011; Zhao et al., 2010). (Griffiths et al., 2004) introduced a model that can model both the semantic and syntactic information based on a traditional topic model and a hidden Markov model (HMM). However, all these models are quite different from ours in both the goal and model composition. Additionally, none of these existing models is able to automatically learn prior knowledge and use it to improve its model inference and its modeling results.

Since our LAST model can exploit prior aspect and opinion knowledge, it is thus related to knowledge-based topic models such as (Andrzejewski et al., 2009; Jagarlamudi et al., 2012; Petterson et al., 2010; Mukherjee and Liu, 2012; Hu et al., 2014). However, the knowledge used in these systems are all provided by the user. Our work is also related to transfer learning and lifelong machine learning. Topic models have been used to help transfer learning in (Xue et al., 2008; Pan and Yang, 2010a; Kang et al., 2012; Yang et al., 2011). However, transfer learning in these works is for supervised classification and requires human labeling. Our work is more related to (Chen and Liu, 2014c; Chen and Liu, 2014a) which combines topic modeling with lifelong machine learning. (Chen and Liu, 2014c) considered the positive word correlation as the knowledge while (Chen and Liu, 2014a) further utilized the negative word correlation.

However, they did not separate aspects and opinions and nor did they consider polarity or generality as we do in our fine-grained modeling. Moreover, they did not consider aspect-opinion knowledge and they cannot identify aspect-specific opinions. Also in the context of sentiment analysis, (Chen et al., 2015b) proposed the LSC (Lifelong Sentiment Classification) model that tackles the supervised polarity classification problem. However, it did not use topic modeling and the classification was at the document level, while our models are unsupervised and for aspect-level opinion mining.

5.6 Summary

This work proposed to jointly model aspect, opinion, polarity and generality. The goal is to provide a holistic solution for the four dimensions and make the extracted aspect-specific opinions more coherent to aspects. For that we first presented a new joint model called JAST that can simultaneously model all the four dimensions, and then introduced a more advanced model called LAST, which can extract and leverage the prior knowledge from multiple domains to improve the performance of JAST, incorporating the idea of lifelong machine learning. Experimental results using reviews from 50 product types show significant improvements over state-of-the-art baseline models.

CHAPTER 6

CONCLUSIONS

In this thesis, we have studied target-oriented content and sentiment analysis. First, we introduced a target-oriented content analysis task named targeted topic modeling and proposed a new topic model to address it. Next, we introduced a target-oriented sentiment analysis task named target-sensitive sentiment classification and proposed several alternative approaches to solving it. After discussing them, we indicated one of their shortcomings is that they only consider the data in one domain independently, which limits the available information and their model capabilities. To address this issue and to improve their model performance in a fundamental way, we introduced the concept of lifelong machine learning (LML).

LML enables a system to learn as humans, using the knowledge accumulated from the past to help future learning. Two further studies were then presented to show the effectiveness of using LML for target-oriented content and sentiment analysis. One is to address the aspect sentiment classification problem, by learning and incorporating target-specific attention and sentiment knowledge. Another one is to holistically identify aspect terms and aspect-specific opinion terms, with the knowledge involved in the topic modeling process.

In a nutshell, the contributions of our works are summarized as follows:

1. We thoroughly introduced the concept of target-oriented content and analysis. We also discussed its research and application importance in this big data era, where a huge

amount of user-generated content is available on the Web, especially the social media platforms.

2. We discussed two specific target-oriented content and analysis tasks, namely, targeted topic modeling for focused analysis, and target-sensitive aspect sentiment classification. We also discussed their technical contribution in details and reported their experimental results to demonstrate their effectiveness.
3. We also proposed to use the lifelong machine learning idea for the target-oriented content and analysis tasks. Two research works showed its usefulness in both target-level supervised and unsupervised learning. Also, both of the two studies have successfully employed the big (unlabeled) data from multiple domains in a lifelong learning setting.

We expect that there will be more or more diverse target-oriented analysis and research in a short future. There are many possible directions for future works, for example, target-oriented dialog generation, target-oriented graph construction, and target-oriented emotion detection. We may further consider target-oriented image analysis and target-oriented multi-modal learning to affect broader fields. In addition, we also believe that lifelong machine learning will further benefit target-oriented analysis. On one hand, one can apply lifelong machine learning to other analysis tasks; on the other hand, one can propose more suitable knowledge or more sophisticated algorithms to make systems learn more like humans.

APPENDICES

Appendix A : Copyrights

Title: Lifelong Learning Memory Networks for Aspect Sentiment Classification

Conference Proceedings: 2018 IEEE International Conference on Big Data (Big Data)

Author: Shuai Wang

Publisher: IEEE

Date: Dec. 2018

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

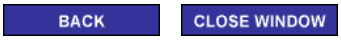
Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.



Association for Computing Machinery (ACM) Author Rights[1]

REUSE

Authors can reuse any portion of their own work in a new work of *their own* (and no fee is expected) as long as a citation and DOI pointer to the Version of Record in the ACM Digital Library are included.

- Contributing complete papers to any edited collection of reprints for which the author is *not* the editor, requires permission and usually a republication fee.

Authors can include partial or complete papers of their own (and no fee is expected) in a dissertation as long as citations and DOI pointers to the Versions of Record in the ACM Digital Library are included. Authors can use any portion of their own work in presentations and in the classroom (and no fee is expected).

- Commercially produced course-packs that are *sold* to students require permission and possibly a fee.

[1]https://authors.acm.org/main.html

Association for Computational Linguistics (ACL) Copyright Policy[1]

Each of the authors and the employers for whom the work was performed reserve all other rights, specifically including the following:

(1) All proprietary rights other than copyright and publication rights transferred to ACL;

(2) The right to publish in a journal or collection or to be used in future works of the author's own (such as articles or books) all or part of this work, provided that acknowledgment is given to the ACL and a full citation to its publication in the particular proceedings is included;

(3) The right to make oral presentation of the material in any forum;

(4) The right to make copies of the work for internal distribution within the author's organization and for external distribution as a preprint, reprint, technical report, or related class of document.

[1]http://www.cs.columbia.edu/~mcollins/ACL2012.copyright.pdf

CITED LITERATURE

[Abelson, 1983] Robert P Abelson. 1983. Whatever became of consistency theory? *Personality and Social Psychology Bulletin*.

[Agrawal et al., 1994] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *VLDB*.

[Andrzejewski et al., 2009] David Andrzejewski, Xiaojin Zhu, and Mark Craven. 2009. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *ICML*, pages 25–32. ACM.

[Archambeau et al., 2015] Cedric Archambeau, Balaji Lakshminarayanan, and Guillaume Bouchard. 2015. Latent ibp compound dirichlet allocation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):321–333.

[Bahdanau et al., 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

[Bengio et al., 2011] Yoshua Bengio, Aaron C Courville, and James S Bergstra. 2011. Unsupervised models of images by spike-and-slab rbms. In *ICML*, pages 1145–1152.

[Blei et al., 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

[Branavan et al., 2009] SRK Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. 2009. Learning document-level semantic properties from free-text annotations. *Journal of Artificial Intelligence Research*, pages 569–603.

[Brody and Elhadad, 2010] Samuel Brody and Noemie Elhadad. 2010. An unsupervised aspect-sentiment model for online reviews. In *NAACL*, pages 804–812. Association for Computational Linguistics.

[Caruana, 1998] Rich Caruana. 1998. Multitask learning. In *Learning to learn*. Springer.

[Chang et al., 2009] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber, and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *NIPS*, pages 288–296.

[Chen and Liu, 2014a] Zhiyuan Chen and Bing Liu. 2014a. Mining topics in documents: standing on the shoulders of big data. In *KDD*, pages 1116–1125. ACM.

[Chen and Liu, 2014b] Zhiyuan Chen and Bing Liu. 2014b. Topic modeling using topics from many domains, lifelong learning and big data. In *ICML*.

[Chen and Liu, 2014c] Zhiyuan Chen and Bing Liu. 2014c. Topic Modeling using Topics from Many Domains, Lifelong Learning and Big Data. In *ICML*, pages 703–711.

[Chen and Liu, 2016] Zhiyuan Chen and Bing Liu. 2016. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*.

[Chen et al., 2011] Xi Chen, Yanjun Qi, Bing Bai, Qihang Lin, and Jaime G Carbonell. 2011. Sparse latent semantic analysis. In *SDM*, pages 474–485. SIAM.

[Chen et al., 2012] Xu Chen, Mingyuan Zhou, and Lawrence Carin. 2012. The contextual focused topic model. In *KDD*, pages 96–104. ACM.

[Chen et al., 2015a] Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2015a. Lifelong learning for sentiment classification. In *ACL*.

[Chen et al., 2015b] Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2015b. Lifelong Learning for Sentiment Classification. In *ACL*, pages 750–756.

[Chen et al., 2017] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent attention network on memory for aspect sentiment analysis. In *Empirical Methods in Natural Language Processing*, pages 452–461.

[Cho et al., 2015] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. 2015. Describing multimedia content using attention-based encoder-decoder networks. In *IEEE Transactions on Multimedia*, pages 1875–1886. IEEE.

[Choi and Cardie, 2008] Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Empirical Methods in Natural Language Processing*, pages 793–801.

[Deng et al., 2014] Li Deng, Dong Yu, et al. 2014. Deep learning: methods and applications. In *Foundations and Trends® in Signal Processing*, pages 197–387. Now Publishers, Inc.

[Diao et al., 2014] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 193–202. ACM.

[Ding et al., 2008] Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *ACM International Conference on Web Search and Data Mining*, pages 231–240.

[Dong et al., 2014] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Annual Meeting of the Association for Computational Linguistics*, pages 49–54.

[Eisenstein et al., 2011] Jacob Eisenstein, Amr Ahmed, and Eric P Xing. 2011. Sparse additive generative models of text.

[Fang and Huang, 2012] Lei Fang and Minlie Huang. 2012. Fine granular aspect analysis using latent structural models. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 333–337. Association for Computational Linguistics.

[Griffiths and Steyvers, 2004] Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *PNAS*, 101(suppl 1):5228–5235.

[Griffiths and Tenenbaum, 2004] DMBTL Griffiths and MIJJB Tenenbaum. 2004. Hierarchical topic models and the nested chinese restaurant process. *NIPS*, 16:17.

[Griffiths et al., 2004] Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. 2004. Integrating topics and syntax. In *Advances in neural information processing systems*, pages 537–544.

[He et al., 2011] Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Hu-*

man Language Technologies-Volume 1, pages 123–131. Association for Computational Linguistics.

[Heinrich, 2009] Gregor Heinrich. 2009. A generic approach to topic models. In *ECML-PKDD*, pages 517–532. Springer.

[Hermann et al., 2015] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

[Hofmann, 1999] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *SIGIR*, pages 50–57. ACM.

[Hong et al., 2011] Liangjie Hong, Dawei Yin, Jian Guo, and Brian D Davison. 2011. Tracking trends: incorporating term volume into temporal topic models. In *KDD*, pages 484–492. ACM.

[Hu and Liu, 2004a] Minqing Hu and Bing Liu. 2004a. Mining and summarizing customer reviews. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

[Hu and Liu, 2004b] Minqing Hu and Bing Liu. 2004b. Mining and summarizing customer reviews. In *SIGKDD*.

[Hu et al., 2014] Yuening Hu, Jordan Boyd-Graber, Brianna Satinoff, and Alison Smith. 2014. Interactive topic modeling. *Machine learning*, 95(3):423–469.

[Ishwaran and Rao, 2005] Hemant Ishwaran and J Sunil Rao. 2005. Spike and slab variable selection: frequentist and bayesian strategies. *Annals of Statistics*, pages 730–773.

[Jagarlamudi et al., 2012] Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. 2012. Incorporating lexical priors into topic models. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 204–213. Association for Computational Linguistics.

[Jiang et al., 2011] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *Annual Meeting of the Association for Computational Linguistics*, pages 151–160.

[Jo and Oh, 2011] Yohan Jo and Alice H Oh. 2011. Aspect and sentiment unification model for online review analysis. In *WSDM*, pages 815–824. ACM.

[Kang et al., 2012] Jeon-Hyung Kang, Jun Ma, and Yang Liu. 2012. Transfer topic modeling with ease and scalability. In *SDM*, pages 564–575. SIAM.

[Kawamae, 2010] Noriaki Kawamae. 2010. Latent interest-topic model: finding the causal relationships behind dyadic data. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 649–658. ACM.

[Kim et al., 2013] Suin Kim, Jianwen Zhang, Zheng Chen, Alice H Oh, and Shixia Liu. 2013. A hierarchical aspect-sentiment model for online reviews. In *AAAI*.

[Kim, 2014] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Empirical Methods in Natural Language Processing*, pages 1746–1751.

[Kiritchenko et al., 2014] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *SemEval*.

[Kumar et al., 2016] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387.

[Lazaridou et al., 2013] Angeliki Lazaridou, Ivan Titov, and Caroline Sporleder. 2013. A bayesian model for joint unsupervised induction of sentiment, aspect and discourse representations. In *ACL (1)*, pages 1630–1639.

[Li et al., 2011] Peng Li, Yinglin Wang, Wei Gao, and Jing Jiang. 2011. Generating aspect-oriented multi-document summarization with event-aspect model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1137–1146. Association for Computational Linguistics.

[Lin and He, 2009] Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM.

[Lin et al., 2014] Tianyi Lin, Wentao Tian, Qiaozhu Mei, and Hong Cheng. 2014. The dual-sparse topic model: mining focused topics and focused terms in short text. In *WWW*, pages 539–550. ACM.

[Liu and Zhang, 2017] Jiangming Liu and Yue Zhang. 2017. Attention modeling for targeted sentiment. In *European Chapter of the Association for Computational Linguistics*, pages 572–577.

[Liu et al., 2016] Qian Liu, Bing Liu, Yuanlin Zhang, Doo Soon Kim, and Zhiqiang Gao. 2016. Improving opinion aspect extraction using semantic similarity and aspect associations. In *AAAI*.

[Liu, 2012] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*.

[Lu and Zhai, 2008] Yue Lu and Chengxiang Zhai. 2008. Opinion integration through semi-supervised topic modeling. In *Proceedings of the 17th international conference on World Wide Web*, pages 121–130. ACM.

[Lu et al., 2009] Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *Proceedings of the 18th international conference on World wide web*, pages 131–140. ACM.

[Lu et al., 2011] Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic construction of a context-aware sentiment lexicon: an optimization approach. In *ACM International Conference on World Wide Web*, pages 347–356.

[Luong et al., 2015] Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Empirical Methods in Natural Language Processing*, pages 1412–1421.

[Ma et al., 2017] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *International Joint Conference on Artificial Intelligence*, pages 4068–4074.

[Mahmoud, 2008] Hosam Mahmoud. 2008. *Polya Urn Models*. Chapman & Hall/CRC Texts in Statistical Science.

[Mei and Zhai, 2005] Qiaozhu Mei and ChengXiang Zhai. 2005. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD*, pages 198–207. ACM.

[Mei et al., 2007] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM.

[Mimno et al., 2011] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics.

[Min et al., 2010] Kerui Min, Zhengdong Zhang, John Wright, and Yi Ma. 2010. Decomposing background topics from keywords by principal component pursuit. In *CIKM*, pages 269–278. ACM.

[Mitchell and Beauchamp, 1988] Toby J Mitchell and John J Beauchamp. 1988. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032.

[Mitchell et al., 2015] Tom M Mitchell, William W Cohen, Estevam R Hruschka Jr, Partha Pratim Talukdar, Justin Betteridge, Andrew Carlson, Bhavana Dalvi Mishra, Matthew Gardner, Bryan Kisiel, Jayant Krishnamurthy, et al. 2015. Never ending learning. In *AAAI*.

[Mnih et al., 2014] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212.

[Moghaddam and Ester, 2013] Samaneh Moghaddam and Martin Ester. 2013. The flda model for aspect-based opinion mining: Addressing the cold start problem. In *WWW*.

[Moilanen and Pulman, 2007] Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *RANLP*, pages 378–382.

[Mukherjee and Liu, 2012] Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *ACL*, pages 339–348. Association for Computational Linguistics.

[Newman et al., 2010a] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010a. Automatic evaluation of topic coherence. In *HLT-NAACL*, pages 100–108.

[Newman et al., 2010b] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010b. Evaluating topic models for digital libraries. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 215–224. ACM.

[Nguyen and Shirai, 2015] Thien Hai Nguyen and Kiyoaki Shirai. 2015. Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis. In *Empirical Methods in Natural Language Processing*, pages 2509–2514.

[Pan and Yang, 2010a] Sinno Jialin Pan and Qiang Yang. 2010a. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359.

[Pan and Yang, 2010b] Sinno Jialin Pan and Qiang Yang. 2010b. A survey on transfer learning. *TKDE*.

[Pang et al., 2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Empirical Methods in Natural Language Processing*, pages 79–86.

[Paul and Dredze, 2012] Michael Paul and Mark Dredze. 2012. Factorial lda: Sparse multi-dimensional text models. In *Advances in Neural Information Processing Systems*, pages 2582–2590.

[Paul and Dredze, 2013] Michael J Paul and Mark Dredze. 2013. Drug extraction from the web: Summarizing drug experiences with multi-dimensional topic models. In *HLT-NAACL*, pages 168–178.

[Paul and Girju, 2009] Michael Paul and Roxana Girju. 2009. Cross-cultural analysis of blogs and forums with mixed-collection topic models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1408–1417. Association for Computational Linguistics.

[Paul and Girju, 2010] Michael Paul and Roxana Girju. 2010. A two-dimensional topic-aspect model for discovering multi-faceted topics. *Urbana*, 51:61801.

[Petterson et al., 2010] James Petterson, Wray Buntine, Shravan M Narayanamurthy, Tibério S Caetano, and Alex J Smola. 2010. Word features for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1921–1929.

[Polanyi and Zaenen, 2006] Livia Polanyi and Annie Zaenen. 2006. Contextual valence shifters. In *Computing attitude and affect in text: Theory and applications*, pages 1–10. Springer.

[Pontiki et al., 2014] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task4: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2014)*. Association for Computational Linguistics.

[Ramage et al., 2009] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. 2009. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256. Association for Computational Linguistics.

[Rosen-Zvi et al., 2004] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *UAI*, pages 487–494. AUAI Press.

[Salakhutdinov and Hinton, 2009] Ruslan Salakhutdinov and Geoffrey E Hinton. 2009. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 448–455.

[Sauper and Barzilay, 2013] Christina Sauper and Regina Barzilay. 2013. Automatic aggregation by joint modeling of aspects and values. *Journal of Artificial Intelligence Research*.

[Shu et al., 2016] Lei Shu, Bing Liu, Hu Xu, and Annice Kim. 2016. Lifelong-rl: Lifelong relaxation labeling for separating entities and aspects in opinion targets. In *EMNLP*.

[Socher et al., 2011] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Empirical Methods in Natural Language Processing*, pages 151–161.

[Socher et al., 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Empirical Methods in Natural Language Processing*, pages 1631–1642.

[Sukhbaatar et al., 2015] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

[Tang et al., 2016] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. In *Empirical Methods in Natural Language Processing*, pages 214–224.

[Thrun, 1998a] Sebastian Thrun. 1998a. Lifelong learning algorithms. *Learning to learn*.

[Thrun, 1998b] Sebastian Thrun. 1998b. Lifelong Learning Algorithms. In S Thrun and L Pratt, editors, *Learning To Learn*. Kluwer Academic Publishers.

[Titov and McDonald, 2008a] Ivan Titov and Ryan McDonald. 2008a. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120. ACM.

[Titov and McDonald, 2008b] Ivan Titov and Ryan T McDonald. 2008b. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, volume 8, pages 308–316. Citeseer.

[Vo and Zhang, 2015] Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *International Joint Conference on Artificial Intelligence*, pages 1347–1353.

[Wagner et al., 2014] Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. Dcu: Aspect-based polarity classification for semeval task 4. In *SemEval*.

[Wallach et al., 2009] Hanna M Wallach, David M Mimno, and Andrew McCallum. 2009. Rethinking lda: Why priors matter. In *NIPS*, pages 1973–1981.

[Wallach, 2006] Hanna M Wallach. 2006. Topic modeling: beyond bag-of-words. In *ICML*, pages 977–984. ACM.

[Wang and Blei, 2009] Chong Wang and David M Blei. 2009. Decoupling sparsity and smoothness in the discrete hierarchical dirichlet process. In *NIPS*, pages 1982–1989.

[Wang et al., 2010] Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *KDD*, pages 783–792. ACM.

[Wang et al., 2011] Quan Wang, Jun Xu, Hang Li, and Nick Craswell. 2011. Regularized latent semantic indexing. In *SIGIR*, pages 685–694. ACM.

[Wang et al., 2015] Linlin Wang, Kang Liu, Zhu Cao, Jun Zhao, and Gerard de Melo. 2015. Sentiment-aspect extraction based on restricted boltzmann machines.

[Wang et al., 2016a] Shuai Wang, Zhiyuan Chen, Geli Fei, Bing Liu, and Sherry Emery. 2016a. Targeted topic modeling for focused analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1235–1244. ACM.

[Wang et al., 2016b] Shuai Wang, Zhiyuan Chen, and Bing Liu. 2016b. Mining aspect-specific opinion using a holistic lifelong topic model. In *WWW*, pages 167–176. WWW.

[Wang et al., 2016c] Yequan Wang, Minlie Huang, Li Zhao, et al. 2016c. Attention-based lstm for aspect-level sentiment classification. In *Empirical Methods in Natural Language Processing*, pages 606–615.

[Wang et al., 2018a] Shuai Wang, Guangyi Lv, Sahisnu Mazumder, Geli Fei, and Bing Liu. 2018a. Lifelong learning memory networks for aspect sentiment classification. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 861–870. IEEE.

[Wang et al., 2018b] Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018b. Target-sensitive memory networks for aspect sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 957–967.

[Weston et al., 2015] Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *International Conference on Learning Representations*.

[Williamson et al., 2010] Sinead Williamson, Chong Wang, Katherine A Heller, and David M Blei. 2010. The ibp compound dirichlet process and its application to focused topic modeling. In *ICML*, pages 1151–1158.

[Wu and Wen, 2010] Yunfang Wu and Miaomiao Wen. 2010. Disambiguating dynamic sentiment ambiguous adjectives. In *International Conference on Computational Linguistics*, pages 1191–1199.

[Xue et al., 2008] GR Xue, Wenyuan Dai, Q Yang, and Y Yu. 2008. Topic-bridged PLSA for cross-domain text classification. In *SIGIR*, pages 627–634.

[Yang and Cardie, 2013] Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *ACL (1)*, pages 1640–1649.

[Yang et al., 2011] Shuang-hong Yang, Steven P Crain, and Hongyuan Zha. 2011. Bridging the language gap: topic adaptation for documents with different technicality. In *International Conference on Artificial Intelligence and Statistics*, pages 823–831.

[Yang et al., 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

[Zhai et al., 2004] ChengXiang Zhai, Atulya Velivelli, and Bei Yu. 2004. A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 743–748. ACM.

[Zhang et al., 2015] Meishan Zhang, Yue Zhang, and Duy Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *Empirical Methods in Natural Language Processing*, pages 612–621.

[Zhang et al., 2018] Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. In *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1253. Wiley Online Library.

[Zhao et al., 2010] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. 2010. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *EMNLP*, pages 56–65. Association for Computational Linguistics.

[Zhu and Xing, 2012] Jun Zhu and Eric P Xing. 2012. Sparse topical coding. *arXiv preprint arXiv:1202.3778*.

[Zipf, 1932] G Zipf. 1932. Selective studies and the principle of relative frequencies in language.

VITA

Shuai Wang

Education

Ph.D., University of Illinois at Chicago, Chicago, Illinois, 2019.

M.S., Chinese University of Hong Kong, Hong Kong, 2013.

B.E., Guangdong University of Foreign Studies, Guangzhou, China, 2012.

Publications

- Guangyi Lv, Shuai Wang, Bing Liu, Enhong Chen, and Kun Zhang. Sentiment Classification by Leveraging the Shared Knowledge from a Sequence of Domains. In Proceedings of the 24th International Conference on Database Systems for Advanced Applications (DASFAA 2019).
- Shuai Wang, Guangyi Lv, Sahisnu Mazumder, Geli Fei, and Bing Liu. Lifelong Learning Memory Networks for Aspect Sentiment Classification. In Proceedings of the 2018 IEEE International Conference on Big Data (IEEE BigData 2018).
- Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. Target-sensitive Memory Networks for Aspect Sentiment Classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018).
- Lei Zhang, Shuai Wang, and Bing Liu. Deep Learning for Sentiment Analysis: A Survey. Transactions on WIREs Data Mining and Knowledge Discovery, 2018.

- Daniel K. Cortese, Glen Szczypka, Sherry Emery, Shuai Wang, Elizabeth Hair, and Donna Vallone. Smoking Selfies: Using Instagram to Explore Young Women's Smoking Behaviors. In Social Media and Society: SAGE Journals, 2018.
- Huayi Li, Geli Fei, Shuai Wang, Bing Liu, Weixiang Shao, Arjun Mukherjee, and Jidong Shao. Bimodal Distribution and Co-Bursting in Review Spam Detection. In Proceedings of the 26th International Conference on World Wide Web (WWW 2017).
- Shuai Wang, Zhiyuan Chen, Geli Fei, Bing Liu, and Sherry Emery. Targeted Topic Modeling for Focused Analysis. In Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2016).
- Geli Fei, Shuai Wang, Bing Liu. Learning Cumulatively to Become More Knowledgeable. In Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2016).
- Shuai Wang, Zhiyuan Chen, and Bing Liu. Mining Aspect-specific Opinion using a Holistic Lifelong Topic Model. In Proceedings of the 25th International World Wide Web Conference (WWW 2016)
- Shuai Wang, Zhiyuan Chen, Bing Liu and Sherry Emery. Identifying Search Keywords for Finding Relevant Social Media Posts. In Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016).

Working Experience

Research Intern at Rakuten Institute of Technology, Boston, MA, USA, May 2018 - August 2018

Intern Scientist at Yahoo! Research, Sunnyvale, CA, USA, May 2016 - August 2016

Research Assistant at Health Media Collaboratory, Institute for Health Research and Policy, Chicago, IL, May 2015 - August 2015

Research Staff at Chinese University of Hong Kong, Hong Kong, June 2013 - May 2014

Honors

ACL Travel Award, 2018

IEEE BigData Travel Award, 2018

UIC Graduate Student Presenter Award, 2018

Fifty for The Future, awarded by Illinois Technology Foundation, 2017

AAAI Student Scholarship Award, 2016

WWW Student Grant Award, 2016

KDD ACL Travel Award, 2016

UIC Graduate Student Presenter Award, 2016

UIC Graduate Student Presenter Award, 2015