Lifelong Machine Learning for Topic Modeling and Classification

by

Zhiyuan Chen
B.E., Software Engineering, Dalian University of Technology, 2011

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, 2015

Chicago, Illinois

Defense Committee:
Bing Liu, Chair and Advisor
Philip S. Yu
Tanya Berger-Wolf
Brian Ziebart
Xavier Amatriain, Quora Inc.

For Vena,

the love of my life.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deep gratitude to my Ph.D. advisor Professor Bing Liu for his tremendous advising and mentoring. It was my great honor to work with him to pursue my Ph.D. degree, through which I have learned valuable research experience and numerous skills for my future career. I sincerely appreciate his guidance, encouragement and patience. I am also grateful to his gracious support in pursuing my research interests, which is essential to developing my research skills and the success of my Ph.D. study.

I also want to thank Professor Philip S. Yu, Professor Tanya Berger-Wolf, Professor Brian Ziebart, and Dr. Xavier Amatriain for taking their precious time to serve on my dissertation committee. Their constructive advices and feedbacks make important contributions to the completion of this thesis.

I would like to thank and acknowledge my collaborators and colleagues, Arjun Mukherjee, Geli Fei, Huayi Li, Shuai Wang, Nianzu Ma, Yueshen Xu, Federico Alberto Pozzi and many others, for all the productive discussions and fruitful collaboration.

Last but not the least, I owe many thanks and appreciation to my family for their consistent support. To my wife Vena, if at all I am successful in life, it is not sheer luck or my brilliance; instead, it is all her support and endless love.

ZC

CONTRIBUTION OF AUTHORS

Chapter 3 presents published manuscripts (Chen et al., 2013b; Chen et al., 2013c; Chen et al., 2013d) for which I was the primary author. Arjun Mukherjee and my advisor Professor Bing Liu contributed to the discussion about the preliminary ideas and the assistance in revising the manuscripts. Meichun Hsu, Malu Castellanos and Riddhiman Ghosh contributed to providing funding and sharing feedback to this research project.

Chapter 4 presents a published manuscript (Chen and Liu, 2014b) for which I was the primary author. My advisor Professor Bing Liu contributed to the discussion about the preliminary ideas and the assistance in revising the manuscript.

Chapter 5 presents a published manuscript (Chen and Liu, 2014a) for which I was the primary author. My advisor Professor Bing Liu contributed to the discussion about the preliminary ideas and the assistance in revising the manuscript.

Chapter 6 presents a published manuscript (Chen et al., 2015) for which I was the primary author. Nianzu Ma assisted me in running baseline models in the experiments shown in Table VI and Table VII . My advisor Professor Bing Liu contributed to the discussion about the preliminary ideas and the assistance in revising the manuscript.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

FIGURE **PAGE**

SUMMARY

Machine Learning (ML) has been successfully used as a prevalent approach for many computational tasks and applications. However, most ML algorithms are designed to address a specific problem using a single dataset. That is, given a dataset, an ML algorithm is run on the dataset to build a model. Although this one-shot learning is very important and useful, it can never make an AI system intelligent, and its accuracy is also limited.

Lifelong Machine Learning (LML), on the other hand, aims to design and develop computational systems and algorithms that learn as humans do, i.e., retaining the results learned in the past, abstracting knowledge from them, and using the knowledge to help future learning and problem solving. The rationale is that when faced with a new situation, we humans use our previous experience and knowledge to help deal with and learn from the new situation. It is essential to incorporate such a capability into a computational system to make it more versatile, holistic, and intelligent.

This thesis presents my Ph.D. research work on designing Lifelong Machine Learning approaches for both unsupervised learning and supervised learning. For unsupervised learning, we focus on the area of topic modeling, which aims to discover coherent semantic topics from the documents. For supervised learning, we propose to improve the problem of classification with the integration of Lifelong Machine Learning.

Topic modeling has been widely used to uncover topics from document collections. Such topics are important in many text mining and machine learning tasks such as classification,

retrieval, clustering and summarization. However, classic unsupervised topic models can generate many incoherent topics. To address them, we proposed several knowledge-based topic models (Chen et al., 2013d; Chen et al., 2013b; Chen et al., 2013c) which require the knowledge to be provided by domain experts. To further ameliorate the topic quality from topic models, in (Chen and Liu, 2014b; Chen and Liu, 2014a), we proposed to automatically extract, accumulate and filter knowledge with the idea of LML, i.e., Lifelong Machine Learning. The experimental results shown in these papers demonstrate the effectiveness of the proposed LML approaches.

We also apply LML for supervised learning, specifically classification. Classification is a widely studied machine learning task. The goal is to classify certain objects into a fixed set of categories. Deviated from traditional classification problem which focuses on a single domain, we proposed our Lifelong Sentiment Classification (LSC) model (Chen et al., 2015) which automatically extracts and accumulates sentiment oriented knowledge. Such knowledge is utilized using regularization under the Naïve Bayesian optimization framework. The experimental results demonstrate that our proposed LSC model is able to accomplish better and better classification performance with knowledge accumulated from an increasing number of domains, which shows the advantages of having LML.

Based on this thesis, we believe that the Lifelong Machine Learning (LML) capability can lead to more robust computational systems to overcome the dynamics and complexity of real-world problems to produce better predictability.

CHAPTER 1

INTRODUCTION

(This chapter includes and expands on my thesis proposal previously published in *Zhiyuan Chen. Lifelong Machine Learning for Topic Modeling and Beyond. In NAACL-HLT 2015 SRW, page 133-139.*)

Machine learning serves as a prevalent approach for research in many computational tasks. However, most of the existing machine learning approaches are built using a single dataset, which is often referred to as one-shot learning. This kind of one-shot approach is useful but it does not usually generalize well to various datasets or tasks. The main shortcoming of such one-shot approaches is the lack of continuous learning ability, i.e., learning and accumulating knowledge from past tasks and leveraging the knowledge for future tasks and problem solving in a lifelong manner.

To overcome the above shortcoming, **Lifelong Machine Learning** (**LML**) has attracted researchers' attention. The term was initially introduced in 1990s (Thrun, 1995; Caruana, 1997). LML aims to design and develop computational systems and algorithms that learn as humans do, i.e., retaining the results learned in the past, abstracting knowledge from them, and using the knowledge to help future learning. The motivation is that when faced with a new situation, we humans always use our previous experience and learned knowledge to help deal with and learn from the new situation, i.e., we learn and accumulate knowledge continuously. The same rationale can be applied to computational models. When a model is built using a

single dataset for a task, its performance is limited. However, if the model sees more datasets from the same or similar tasks, it should be able to adjust its learning algorithm for better performance.

According to (Chen et al., 2015), there are several questions and challenges in designing an LML system:

1. What information should be retained from the past learning tasks?
2. What forms of knowledge will be used to help future learning?
3. How does the system obtain the knowledge?
4. How does the system use the knowledge to help future learning?

Compared to the significant progress of machine learning theory and algorithm, there is relatively little study on lifelong machine learning. One of the most notable works is the Never-Ending Language Learner (NELL) (Carlson et al., 2010; Mitchell et al., 2015) which was proposed to extract or read information from the web to expand the knowledge base in an endless manner, aiming to achieve better performance in each day than the previous day. Some other LML related works include (Silver, 2013; Raina et al., 2007; Pentina and Lampert, 2014; Kamar et al., 2013; Kapoor and Horvitz, 2009). We will present more detailed related works in Chapter 2.

This section first introduces the definition of Lifelong Machine Learning (LML) (Section 1.1). Then it describes my research work on using LML on both unsupervised learning and supervised learning. For unsupervised learning, we focus on topic modeling (Section 1.2). For supervised learning, we aim at supervised classification (Section 1.3).

1.1 Definition of Lifelong Machine Learning

(*Part of this section was previously published in (Chen et al., 2015)*)

In this section, we introduce the formal definition of Lifelong Machine Learning from (Caruana, 1997; Silver et al., 2013; Chen et al., 2015).

Definition of Lifelong Machine Learning: *A learner has performed learning on a sequence of tasks, from 1 to* $N-1$*. When faced with the* N*th task, it uses the knowledge gained in the past* $N-1$ *tasks to help learning for the* N*th task.*

To emphasize the lifelong context, we call tasks 1 to $N-1$ *past tasks/domains* and Nth task *current task/domain*. To answer the questions mentioned above, an LML system needs the following four general components (Chen et al., 2015):

1. *Past Information Store* (**PIS**): It stores the information resulted from the past learning. This may involve sub-stores for information such as (1) the original data used in each past task, (2) intermediate results from the learning of each past task, and (3) the final model or patterns learned from the past tasks, respectively.
2. *Knowledge Base* (**KB**): It stores the knowledge mined or consolidated from **PIS** (Past Information Store). This requires a knowledge representation scheme suitable for the application. The scalability of knowledge base is also essential when big data is concerned.
3. *Knowledge Miner* (**KM**): It mines knowledge from **PIS** (Past Information Store). This mining can be regarded as a meta-learning process because it learns knowledge from information resulted from learning of the past tasks. The knowledge is added into the existing **KB** (Knowledge Base).

4. *Knowledge-Based Learner* (**KBL**): Given the knowledge in **KB**, this learner is able to leverage the knowledge and/or some information in PIS for the new task.

As we will see in this thesis, all the LML systems have some or all of the above components with different variances. Chapter 2 will cover the related works.

1.2 LML on Topic Modeling

(*Part of this section was previously published in (Chen and Liu, 2014a)*)

Topic modeling, such as LDA (Blei et al., 2003) and pLSA (Hofmann, 1999), have been popularly used in many NLP tasks such as opinion mining (Chen et al., 2014), machine translation (Eidelman et al., 2012), word sense disambiguation (Boyd-Graber et al., 2007), phrase extraction (Fei et al., 2014) and information retrieval (Wei and Croft, 2006). In general, topic models assume each document is a multinomial distribution over topics while each semantic topic is a multinomial distribution over words. The two types of distributions in topic modeling are document-topic distributions and topic-word distributions respectively. The intuition is that words are more or less likely to be present given the topics of a document. For example, "sport" and "player" will appear more often in documents about sports, "rain" and "cloud" will appear more frequently in documents about weather.

However, fully unsupervised topic models tend to generate many inscrutable topics. The main reason is that the objective function of topic model is not always consistent with human judgment (Chang et al., 2009). To deal with this problem, there are three main approaches:

1. *Inventing better topic models*: This approach may be effective if a large number of documents with little noise are available. However, since topic models perform unsupervised

learning, if the data is small or noisy, the information is insufficient to provide reliable statistics to generate coherent topics. Some form of supervision or external information beyond the given documents is necessary.

2. *Asking users to provide prior domain knowledge*: An obvious form of external information is the prior knowledge of the domain from the user. For example, the user can input the knowledge in the form of must-link and cannot-link. A must-link states that two terms (or words) should belong to the same topic, e.g., *price* and *cost*. A cannot-link indicates that two terms should not be in the same topic, e.g., *price* and *picture*. Some existing *knowledge-based topic models* (e.g., (Andrzejewski et al., 2009; Andrzejewski et al., 2011; Chen et al., 2013b; Chen et al., 2013c; Hu et al., 2011; Jagarlamudi et al., 2012; Mukherjee and Liu, 2012a; Petterson et al., 2010)) can exploit such prior domain knowledge to produce better topics. However, asking the user to provide prior domain knowledge can be problematic in practice because the user may not know what knowledge to provide and wants the system to discover for him/her. It also makes the approach non-automatic.
3. *Learning like humans* (*lifelong machine learning*): We still use the *knowledge-based approach* but mine the prior knowledge automatically from the results of past learning. This approach works like human learning. We humans always retain the results learned in the past and use them to help future learning. That is why whenever we see a new situation, we may notice few things are really new because we have seen many aspects of it in the past in some other contexts. Our proposed technique takes this approach. It represents a

major step forward as it closes the learning or modeling loop in the sense that the whole process is now fully automatic and can learn or model continuously.

Chapters 3, 4, 5 will introduce our work on lifelong machine learning for topic modeling. Chapter 3 first discusses knowledge-based topic modeling. Chapter 4 extends it with the idea of LML, i.e., automatically extracting and accumulating the must-type of knowledge from the topic results of past domains and exploiting such knowledge to generate higher quality topics. Chapter 5 further introduces the cannot-type of knowledge into LML on topic modeling. As we will see in Chapter 5, the cannot-type of knowledge is shown to be very effective in reducing the noise in the resulting topics.

1.3 LML on Classification

(*Part of this section was previously published in (Chen et al., 2015)*)

Classification is a widely studied machine learning problem. The task is to classify the objects into certain categories. Binary classification is the most common situation, where there are two classes (or categories): positive class and negative class. Although we focus on binary classification problem in this thesis, our proposed techniques can be naturally adapted to multi-class classification problem.

In (Chen et al., 2013a), we studied a novel problem of classification which is also of great practical value, namely, *Intention Identification*, which aims to identify discussion posts expressing certain user intentions that can be exploited by businesses or other interested parties. For example, one user wrote, "*I am looking for a brand new car to replace my old Ford Focus.* Identifying such intentions automatically can help social media sites to decide what ads to dis-

play so that the ads are more likely to be clicked. We proposed a new approach called Co-Class (Co-Classification) which is able to transfer the information from source domain data (labeled) to target domain data (unlabeled). See (Chen et al., 2013a) for more details.

In (Chen et al., 2014), we took one much further step to demonstrate the benefits of LML in classification. We focus on the problem of sentiment classification. Sentiment classification is the task of classifying an opinion document as expressing a positive or negative sentiment. The problem has been studied by many researchers. There are both supervised and unsupervised learning techniques. The books by (Liu, 2012) and (Pang and Lee, 2008) give good surveys of the existing research on sentiment classification. In this thesis, we depart from the existing research directions of supervised and unsupervised learning, feature engineering, and transfer learning or domain adaptation. We define our problem of lifelong sentiment classification (LSC) as:

Definition of Lifelong Sentiment Classification: *A learner has performed a sequence of supervised sentiment classification tasks, from 1 to* $N-1$*, where each task consists of a set of training documents with positive and negative polarity labels. Given the* N*th task, it uses the knowledge gained in the past* $N-1$ *tasks to learn a better classifier for the* N*th task.*

It is useful to note that although many researchers have used transfer learning for supervised sentiment classification, LML is different from the classic transfer learning or domain adaptation (Pan and Yang, 2010). Transfer learning typically uses labeled training data from one (or more) source domain(s) to help learning in the target domain that has little or no labeled data (Aue and Gamon, 2005; Bollegala et al., 2011) (See Chapter 2). It does not use the results

of the past learning or knowledge mined from the results of the past learning. Further, transfer learning is usually inferior to traditional supervised learning when the target domain already has good training data. In contrast, our target (or future) domain/task has good training data and we aim to further improve the learning using both the target domain training data and the knowledge gained in past learning. To be consistent with prior research, we treat the classification of one domain as one learning task.

Chapter 6 will cover our work on lifelong machine learning on sentiment classification. The rationale is that if a word is appearing as a positive (or negative) sentiment word over many past domains, it is likely that this word is indicating the same sentiment polarity in the current domain. On the other hand, if a word is ambiguous in distinguishing the polarity from past domains, we should rely less on them when classifying the current domain.

This thesis ends with the conclusions, the summary of its contributions, and some interesting future directions (see Chapter 7).

CHAPTER 2

RELATED WORKS

2.1 Early Works of LML

Lifelong Machine Learning (LML) is our focused problem in this thesis. It was first studied in 1990s (Thrun, 1996b; Caruana, 1997; Thrun and Mitchell, 1995) which focused on supervised learning. (Thrun, 1996b) studied concept learning, i.e., pattern classification task, in the context of LML. A concept learning task is to learn a function $f : I \rightarrow \{0, 1\}$ where $f(x) = 1$ means x belongs to a particular concept; otherwise x does not belong to it. For example, $f_{dog}(x) = 1$ means x belongs to a concept of *dog*. In order to learn from the previous data, a distance function is learned from previous data using an artificial neural network with Back-Propagation. This distance function, which serves as the knowledge, indicates the probability that two data points are members of the same concept. The lifelong mechanism will update the distance function when more data is seen. To use the knowledge (the distance function), an explanation-based neural network (EBNN) (Thrun, 1996a) is applied.

(Caruana, 1997) studied multi-task learning with neural network (called MTL-Net), in which the tasks are trained in parallel using a common hidden layer. Instead of training individual neural network for each individual task, (Caruana, 1997) proposed to train a neural network for all tasks. The neural network takes the combined inputs of all tasks, and then produces outputs for each task. Backpropagation is done in parallel on the outputs in the MTL-Net.

Same as (Thrun, 1996b), (Caruana, 1997) stores all the data for all tasks. One task can influence the other tasks by transferring the domain knowledge to help the hidden layer learn a better internal representation. The representation that multiple tasks share the same base motivated many subsequent researches (Kumar et al., 2012; Ruvolo and Eaton, 2013a).

2.2 Related Areas to LML

There are many related areas to Lifelong Machine Learning, including Transfer Learning, Multi-task Learning, Never-ending Learning, Self-taught Learning, and Online Learning. In general, we can treat them as different variants of LML while each of them focuses on specific sub-problems. We will detail each of them in the following sub-sections.

2.2.1 Transfer Learning

Transfer learning (or domain adaptation) (Pan and Yang, 2010; Jiang, 2008) has been widely researched in the recent years. Typically, transfer learning involves with two domains: a source domain and a target domain. The source domain has a good amount of labeled training data while the target domain has little or no labeled training data. The goal is to leverage the supervised information from the source domain to help the prediction in the target domain. Transfer learning is a special case of LML as it usually only retains the source domain data. Transfer learning also usually assumes that the source domain and the target domain is closely related.

As mentioned in (Pan and Yang, 2010), there are different types of knowledge in transfer learning. (Bickel et al., 2007; Sugiyama et al., 2008; Liao et al., 2005; Dai et al., 2007c; Jiang and Zhai, 2007; Dai et al., 2007b) directly treat certain parts of data instances in the source

domain as the knowledge with instance reweighing and importance sampling. Features from the source domain serve as another type of knowledge (Ando and Zhang, 2005; Dai et al., 2007a; Daume III, 2007; Blitzer et al., 2006; Blitzer et al., 2007; Wang and Mahadevan, 2008). In such cases, features are specific to particular supervised learning tasks and are used to generate new feature representation for the target domain. Other than features, parameters can also be treated as information to transfer (Lawrence and Platt, 2004; Schwaighofer et al., 2004; Gao et al., 2008; Bonilla et al., 2008), where it is assumed that the source task and the target task share some parameters or prior distributions of hyperparameters of the models where their techniques change the parameters of the target domain model by leveraging the shared parameters or prior distributions.

2.2.2 Multi-task Learning

Multi-task learning is to learn multiple related tasks simultaneously, aiming at achieving a better performance by using the relevant information shared by the tasks (Caruana, 1997; Thrun, 1998). The rationale is to introduce inductive bias in the joint hypothesis space of all tasks by exploiting the task relatedness structure. It also prevents overfitting in the individual task and thus has a better generalization. Multi-task learning usually focuses on minimizing the errors on all tasks, and thus when a new task comes, it needs to be run on all tasks including all the past tasks. LML, on the other hand, extracts and accumulates the knowledge from past tasks and runs only on the new task using the retained knowledge.

Similar to transfer learning, multi-task learning usually assumes that the tasks are related to each other. (Evgeniou and Pontil, 2004) assumed that all data for the tasks come from the

same space and all the task models are close to a global model. Under the assumption, they modeled the relation between tasks using a task-coupling parameter with regularization. (Baxter, 2000; Ben-David and Schuller, 2003) assumed that the tasks share a common underlying representation, e.g., using a common set of learned features. Some other works used the probabilistic approach assuming that the parameters share a common prior (Yu et al., 2005; Lee et al., 2007; Daumé III, 2009).

Task parameters can also lie in a low dimensional subspace in order to learn a low dimensional representation that is shared across tasks (Argyriou et al., 2008). However, the low rank assumption does not distinguish tasks. When some unrelated tasks are considered, the performance may deteriorate. To address this issue, some works assume that there are disjoint groups of tasks and apply clustering to tasks (Jacob et al., 2009; Xue et al., 2007). The tasks within a cluster are considered to be close to each other. On the other hand, (Yu et al., 2007) and (Chen et al., 2011) assume that there is a group of related tasks while the unrelated tasks are a small number of outliers. (Gong et al., 2012) assumed that the related tasks share a common set of features while the outlier tasks do not. (Kang et al., 2011) incorporated grouping structure using a regularization framework. However, each group subspace does not overlap, meaning that the possible sharing structure between tasks from different groups is ignored.

Recently, (Kumar et al., 2012) assume that the parameter vector of each task is a linear combination of a finite number of underlying bases. Instead of using the assumption of disjoint task groups (Jacob et al., 2009; Xue et al., 2007), they assume that the tasks in different groups can overlap with each other in one or more bases. They proposed a model called GO-MTL

(Grouping and Overlap in Multi-Task Learning). The goal is to discover the true structure that includes both the tight and loose connections between tasks. Later on, (Ruvolo and Eaton, 2013a) proposed the Efficient Lifelong Learning (ELLA) model that dramatically improves the efficiency of GO-MTL.

There is a recent trend of applying deep Neural Network in the multi-task setting, e.g., (Yim et al., 2015; Liu et al., 2015; Cheng et al., 2015; Huang et al., 2013; Bengio, 2011). (Bengio, 2011) discussed how unsupervised pre-training of representations or feature structures can be exploited in the scenario of transfer learning. (Liu et al., 2015) worked on multi-task deep NN for semantic classification of search queries. Multi-labeling problem is tackled using a similar setting where one classification of deep NN is generalized into multiple binary classification tasks (Huang et al., 2013).

2.2.3 Never-ending Learning

Never-ending Learning shares a similar rationale to LML in the sense that it aims to achieve better and better performance after seeing more and more data. The most well-known never-ending leaning system is proposed by (Carlson et al., 2010; Mitchell et al., 2015), which aims to obtain information from the web to generate a structured knowledge base. In each day, the learning is aimed to achieve a better performance than the previous day. The system is called Never-Ending Language Learner (NELL) (Carlson et al., 2010). There are two types of knowledge in NELL:

1. Instance of category: the semantic categories of the noun phrases. For example, “Los Angeles” is in the category “city”.

2. Relationship of a pair of noun phrases, e.g., given a name of an organization *org* and a location *loc*, check if hasOfficesIn(*org*, *loc*) is true, which indicates whether *org* has offices in *loc*.

The starting knowledge base defines a set of predicates of categories and relations, with a handful of seed examples. As the NELL system runs, it keeps crawling and reading the web and generating candidate facts and beliefs, which are filtered and integrated in the component called knowledge integrator.

To extract knowledge from the web, NELL uses several subsystem components. For example, it first identifies contextual patterns such as "X plays for Y" using a free-text extractor. Then, the co-occurrence statistics between noun phrases and contextual patterns are used to discover more categories and relations. To further classify noun phrases into categories, a set of binary L2-regularized logistic regression models are built where the training data comes from the existing knowledge base. The lists and tables on the webpages are mined to extract new instances of predicates in the knowledge base. Finally, a first-order learner is applied to learn probabilistic Horn clauses, which are used to infer new relation instances. Through these subsystem components, a set of candidate facts are generated.

For knowledge transfer, the authors designed the Knowledge Integrator (KI) component. Given the candidate facts, KI uses a threshold (i.e., 0.9) to filter those candidates with low-confidence. Furthermore, if a knowledge is validated from multiple sources, it will be promoted even if its confidence is not high. After a candidate fact is promoted as a belief, it will never be demoted.

2.2.4 Self-Taught Learning

Self-taught learning (Raina et al., 2007) is a special type of transfer learning in which the source domain is the same as the target domain. So it only focuses on one single domain. The knowledge comes from a large amount of unlabeled data (big data), which is much easier to obtain than the labeled data. The labeled data and unlabeled data are denoted by D_L and D_U respectively. There is no assumption about the relationship between D_U and D_L. D_U can have different generative distribution from D_L. D_U does not need to contain the labels of D_L.

The basic steps of self-taught learning are as follows:

1. Learn a higher level representation from D_U.
2. Regenerate new features for D_L mapping the original features into the learned representation in Step 1.
3. Build a supervised learning model (e.g., SVM) on the regenerated features from Step 2.

The rationale of learning a higher level representation from D_U is that through the large amount of unlabeled data, the algorithm may be able to learn the "basic element" that comprise an object. For example, for images, the original feature for D_L can be pixel intensity values. Through the unlabeled data learning, the algorithm may learn to represent images using the edges on the images rather than the raw pixel intensity values. By applying this learned representation to D_L, we obtain a higher level representation to D_L which is expected to be more generalizable. After the unsupervised representation is learned, each original training example is transformed to the new dimension space and a supervised learning algorithm, for example SVM, can be built using the transformed training data.

2.2.5 Online Learning

Online learning has been widely studied in the machine learning community, e.g., (Blum, 1998; Foster and Vohra, 1999; Crammer et al., 2006; Shalev-Shwartz and Srebro, 2008; Hu et al., 2009; Yang et al., 2010). The task is to learn from a constant flow of data. The setting of online learning is similar to LML in the sense that it works in the scenario of streaming data. But online learning usually assumes that the new data shares the same distribution with the existing data while LML also considers the new data may come from a new task that does not share the same distribution (or even irrelevant). LML is closer to online multi-task learning (Dekel et al., 2006; Dekel et al., 2007; Ruvolo and Eaton, 2013b; Ammar et al., 2014; Ruvolo and Eaton, 2014). Instead of training on all tasks together (batch version), it works on the online version. For example, (Ammar et al., 2014) proposed an online formulation of policy gradient reinforcement learning in the context of robotics. (Ruvolo and Eaton, 2014) developed an online learning algorithm for sparse dictionary optimization in the online multi-task setting. The tasks they are working on are very different from the tasks (i.e., topic modeling and classification) in this thesis.

LML is related to but also very different from Reinforcement Learning (Kaelbling et al., 1996) where an agent learns behavior through trial-and-error interactions with a dynamic environment. In our scenario, we do not have the concept of environment. The model needs to learn the patterns from the data rather than the interactions as in Reinforcement Learning.

2.3 Related Works on Topic Modeling

(*Part of this section was previously published in (Chen et al., 2014; Chen and Liu, 2014a)*)

Since we apply LML on topic modeling, our work is very related to Knowledge-based Topic Modeling (KBTM). Knowledge-based topic models have been proposed to incorporate prior domain knowledge from the user to improve model performance. Existing works such as (Andrzejewski et al., 2009; Chen et al., 2013d; Mukherjee and Liu, 2012a) considered only the must-link type of knowledge (e.g., *price* and *cost* should be in the same topic) while (Andrzejewski et al., 2009; Chen et al., 2013c) also used the cannot-link type of knowledge (e.g., *price* and *picture*). All of the above models assume the input knowledge to be correct and provided by the user.

Topic models have also been used to help transfer learning (Pan and Yang, 2010; Xue et al., 2008). However, transfer learning in these papers is for traditional supervised classification, which is very different from our work of topic extraction. (Kang et al., 2012) transferred labeled documents from the source domain to the target domain to produce topic models with better fitting. However, we do not use any labeled data in our topic modeling with LML. (Yang et al., 2011) modeled the language gap between topics using a user provided parameter indicating the degree of technicality of the domain. In contrast, our proposed models in (Chen and Liu, 2014b; Chen and Liu, 2014a) are fully automatic with no human intervention. Another key difference is that transfer learning typically uses the data from one source domain to help the target domain classification, while we use the knowledge obtained from a large number of past (source) domains to help the new (target) domain learning or modeling.

Since our experiments are carried out using product reviews, aspect extraction in opinion mining (Liu, 2012) is related. Aspect extraction has been studied by many researchers in

sentiment analysis (Liu, 2012; Pang and Lee, 2008), e.g., using supervised sequence labeling or classification (Choi and Cardie, 2010; Jakob and Gurevych, 2010; Kobayashi et al., 2007; Li et al., 2010; Yang and Cardie, 2013) and using word frequency and syntactic patterns (Hu and Liu, 2004; Ku et al., 2006; Liu et al., 2013; Popescu and Etzioni, 2005; Qiu et al., 2011; Somasundaran and Wiebe, 2009; Wu et al., 2009; Xu et al., 2013; Yu et al., 2011; Zhao et al., 2012; Zhou et al., 2013; Zhuang et al., 2006). However, these works only perform extraction but not aspect term grouping or resolution. Separate aspect term grouping has been done in (Carenini et al., 2005; Guo et al., 2009; Zhai et al., 2011). They assume that aspect terms have been extracted beforehand.

To extract and group aspects simultaneously, topic models have been applied by researchers (Branavan et al., 2008; Brody and Elhadad, 2010; Chen et al., 2013c; Fang and Huang, 2012; He et al., 2011; Jo and Oh, 2011; Kim et al., 2013; Lazaridou et al., 2013; Li et al., 2011; Lin and He, 2009; Lu et al., 2009; Lu et al., 2012; Lu and Zhai, 2008; Mei et al., 2007; Moghaddam and Ester, 2013; Mukherjee and Liu, 2012a; Sauper and Barzilay, 2013; Titov and McDonald, 2008a; Wang et al., 2010; Zhao et al., 2010). Besides the knowledge-based topic models discussed above, document labels are incorporated as implicit knowledge in (Blei and McAuliffe, 2010; Ramage et al., 2009). Geographical region knowledge has also been considered in topic models (Eisenstein et al., 2010). All of these models assume that the prior knowledge is correct.

2.4 Related Works on Sentiment Classification

Since we apply LML on sentiment classification, sentiment classification is clearly related. (Liu, 2012) and (Pang and Lee, 2008) provided good surveys of the existing research on sentiment

classification. Here, we only cover the works of sentiment classification in more than one domain, which is more related to our work. (Aue and Gamon, 2005) trained classifiers for the target domain using various mixes of labeled and unlabeled reviews. (Yang et al., 2006) exploited feature selection. (Tan et al., 2007) first trained a classifier using the labeled source data to label some good examples in the target domain. These examples are then used to build a target domain classifier. (Blitzer et al., 2007) proposed to first find some common or pivot features from the source and the target, and then find correlated features with the pivot features. The final classifier is built using the combined features. Other works along the line include (Jiang and Zhai, 2007; Joshi et al., 2012; Li et al., 2010; Li et al., 2013; Pan et al., 2010). (He et al., 2011; Gao and Li, 2011) used topic modeling to identify opinion topics from both domains to bridge them. (Andreevskaia and Bergler, 2008) used an ensemble method. (Li et al., 2012) extracted sentiment and topic lexicons from cross domains. (Ku et al., 2009) utilized syntactic structures and (Wu et al., 2009) used a graph propagation algorithm. (Xia and Zong, 2011) found that features of some POS tags are often domain-dependent, while of some others are domain-free.

Our work is also related to transfer learning that uses multiples source domains. (Bollegala et al., 2011) proposed a method that create a sentiment sensitive thesaurus using both labeled and unlabeled data from multiple source domains. The created thesaurus is then used to help train a target classifier. (Yoshida et al., 2011) transferred from multiple source to multiple target domains using topic modeling. However, using topic modeling for classification is usually poorer than supervised learning. (Li and Zong, 2008) also assumes labeled training data but no

unlabeled data in the target domain while working with multiple domains. They built a meta-classifier (called CLF) using the outputs of each base classifier constructed in each domain. We will compare with this method in our experiments as it is closely related to our work (Section 6.3).

2.5 Summary

This chapter discussed the related works of LML in general, as well as the related works specific to topic modeling and sentiment classification. We note that although LML is related to Transfer Learning, Multi-task Learning, Never-ending Learning, Self-taught Learning, Online Learning and Reinforcement Learning, there are no consistent terminologies that clearly define and distinguish each of them, which was also mentioned in (Silver et al., 2013). This thesis can serve as an important effort to bridge these learning paradigms.

CHAPTER 3

KNOWLEDGE BASED TOPIC MODELING

(This chapter includes and expands on my papers previously published in

- *Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Discovering Coherent Topics Using General Knowledge. In CIKM 2013, pages 209-218.*
- *Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Exploiting Domain Knowledge in Aspect Extraction. In EMNLP 2013, pages 1655-1667.*
- *Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Leveraging Multi-Domain Prior Knowledge in Topic Models. In IJCAI 2013, pages 2071-2077.*)

Unsupervised topic models, such as pLSA (Hofmann, 1999) and LDA (Blei et al., 2003), provide a powerful framework for extracting latent topics in text documents. However, researchers have found that these unsupervised models often produce topics that are not interpretable or meaningful (Mimno et al., 2011). One key reason is that the objective functions of these models do not always correlate well with human judgments (Chang et al., 2009).

3.1 Knowledge-Based Topic Models

In order to address the issue of incoherent topics in unsupervised topic model, as mentioned in Section 2.3, several knowledge-based topic models have been proposed (e.g., (Andrzejewski et al., 2009; Andrzejewski et al., 2011; Chen et al., 2013b; Chen et al., 2013d; Mukherjee and Liu, 2012a; Hu et al., 2011; Jagarlamudi et al., 2012; Lu et al., 2011; Petterson et al., 2010)). However, these existing knowledge-based models have a key weakness to be applied in real life data mining or machine learning applications, i.e., they all assume that the user knows the domain very well and can provide knowledge suitable for the domain, which is not always the case because in many real life data mining applications, the user wants to discover something new to him or her. The knowledge provided by the user usually needs to be repeatedly tuned in order to fit the domain (i.e., domain dependent knowledge) for model improvement.

This chapter introduces the *first* knowledge-based topic model that explicitly deals with wrong knowledge. The model is called GK-LDA (General Knowledge based LDA) (Chen et al., 2013b).

3.2 Leveraging General Knowledge

There is a vast amount of lexical knowledge about words and their relationships available in online dictionaries or other resources that can be exploited in a model to generate more coherent topics. Such knowledge is domain independent and can be easily extracted automatically from online dictionaries to form a general knowledge base. This knowledge base can serve as an integral part of a topic model system as it does not change from domain to domain and can be applied to any domain without any user involvement.

In particular, we utilize a specific type of lexical knowledge, i.e., lexical semantic relations, which are relations about words. Such relations include synonymy, antonym, hyponymy, taxonomy, meronymy, troponymy, adjective-attribute, etc (Barker, 2006). We focus on synonym, antonym and adjective-attribute relations to demonstrate the benefits of these relations to topic models.

1. **Synonymy**: Two expressions a and b of a language are synonyms *iff* they mean exactly or nearly the same. The notion is typically applied to lexical items, including idioms, but it can be used for larger expressions as well. In this work, we only use the word level synonyms, e.g., *expensive* and *pricey*.
2. **Antonym**: Two expressions a and b of a language are antonyms *iff* they have opposite meanings. Again, the notion can be words or larger expressions. In this work, we only use the word level antonyms, e.g., *expensive* and *cheap*.
3. **Adjective-attribute**: An adjective is a word that modifies nouns and pronouns, primarily by describing a particular quality/attribute of the word it is modifying. Although there are some general adjectives which can describe/modify anything, e.g., *good* and *bad*, most adjectives describe some specific attributes or properties of nouns. For example, *expensive* usually describes *price*, and *beautiful* often describes *appearance*.

Note that some antonyms can be useful (e.g., *expensive* and *cheap* both describe the topic "price") while some synonyms can be harmful (e.g., *picture* and *painting* may not fit coherently for the topic "image" in the domain of digital cameras). We believe that the lexical relations

are beneficial to topic models in the sense that words in such relations are likely to belong to the same topic.

However, there is a major challenge in using lexical relations, i.e., many such relations may not be appropriate for a particular application because a word can have multiple meanings/senses. Each meaning/sense can have a different synonym set, a different antonym set and a different adjective-attribute set. For a particular application, typically only one or two meanings are applicable while the other senses are inappropriate. To make matters worse, even in the same sense, some words in the synonym set (also called *synset*) may be incorrect for a particular domain. For example, the word *picture* has 10 senses as a noun in WordNet (Miller, 1995). The synset for the first sense is {*picture*, *image*, *icon*, *ikon*}. In the domain of digital cameras, *picture* and *image* should belong to the same topic, but *icon* and *ikon* should not share the same topic with *picture* and *image*. In the second sense of *picture*, the synset is {*picture*, *painting*}. These two words are not coherently related for cameras. The situation also exists in the other two lexical relations. To deal with it, we need a model that is able to automatically identify and leverage the right relations for a particular domain.

Before going further, we first describe how the lexical relations are represented in this work. We represent them as sets, e.g., synonyms: {expensive, pricey}, antonyms: {expensive, cheap}, and adjective-attributes: {expensive, price}. In our system, synonyms and antonyms are extracted from WordNet (Miller, 1995). Adjective-attribute relations are obtained from the system in (Fei et al., 2012), which identifies such relations from online dictionaries. To simplify the presentation, we call these sets *LR-sets* (for lexical relation sets). Each LR-set indicates

one sense/meaning of the words inside it. Since the WordNet (Miller, 1995) and the system in (Fei et al., 2012) can provide consistent sense IDs of each word, the synonyms, antonyms and adjective-attributes of the same sense for each word are automatically merged to form an LR-set. For example, for the word *expensive* in the above example, an LR-set {expensive, pricey, cheap, price} is automatically generated indicating one sense of this word. We will see in the Section 3.4 that LR-sets help improve resulting topics dramatically without any user involvement. We called our proposed model GK-LDA (General Knowledge based LDA). To the best of our knowledge, GK-LDA is the first knowledge-based topic model that tries to explicitly deal with the problem of wrong input knowledge for an application domain.

Note that GK-LDA shares the same plate notation with MDK-LDA (Chen et al., 2013d). However, MDK-LDA is insufficient in terms of general knowledge represented by LR-sets due to its indiscrimination on each word in the LR-set. As mentioned before, some senses of a word may not be appropriate in a particular domain, leading to completely or partially incorrect LR-sets. Before applying LR-sets in topic models, we want to estimate the correlation between the domain corpus and LR-sets to have some ideas of the quality of LR-sets. If the domain corpus can validate an LR-set, we then can have a higher confidence in the usefulness of this knowledge, and hence trust it more.

Based on the above idea, we propose a matrix called *word correlation matrix* which estimates the quality of an LR-set by validating the co-occurrences of the words in the LR-set in the corpus (represented by word probabilities under topics in LDA). In more details, the original LDA (without any knowledge) is executed on the corpus at first. The resulting word distributions

under output topics of LDA are then used to estimate the correctness of the relationships in the LR-sets in order to reduce the undesirable effects of them. The intuition is that the words in an LR-set may be less likely to be in the same topic if they have very different probability masses (too far away from each other in the order of word probabilities) under LDAs output topics. With this estimation of LR-sets, we propose the GK-LDA model by employing a new *generalized Pólya urn* (GPU) model (Mahmoud, 2008) with a new Gibbs sampler which can use the proposed word correlation matrix to discriminate the words in each LR-set. In GK-LDA, drawing word w will not only promote the LR-set as a whole, but also discriminately promote each of the correlated words according to the word correlation matrix.

3.3 GK-LDA

This section introduces our proposed model GK-LDA (Chen et al., 2013b).

3.3.1 Generative Process

Since the words in an LR-set share a similar semantic meaning, the model should redistribute the probability masses over words in the LR-set to ensure that they have similar probability under the same topic. To incorporate this idea, a new latent variable s, which denotes the LR-set assignment to each word, is added into LDA. Figure 1 shows the plate notation of the GK-LDA model. Let M be the number of documents where each document m has N_m words. The vocabulary in the corpus is denoted by $\{1, \ldots, V\}$. The number of LR-sets is S. The generative process is given as follows:

1. For each topic $t \in \{1, \ldots, T\}$
    i. Draw a per topic distribution over LR-sets, $\varphi_t \sim Dir(\beta)$

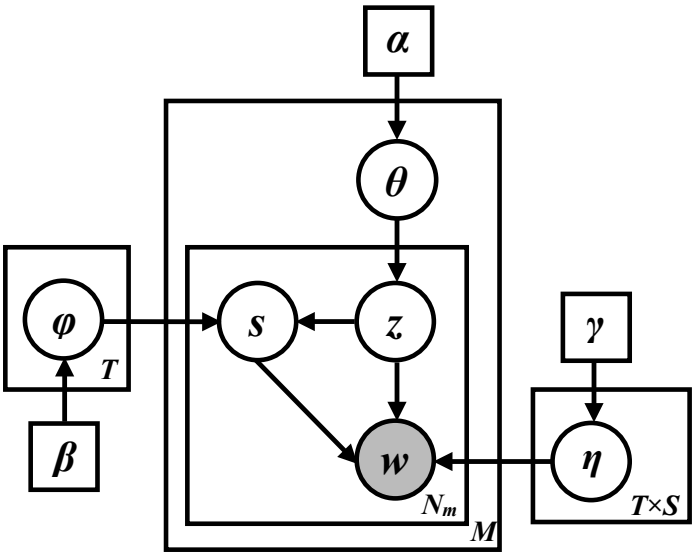

Figure 1. Plate notation of GK-LDA.

ii. For each LR-set $s \in \{1, \ldots, S\}$

a) Draw a per topic, per LR-set distribution over words, $\eta_{t,s} \sim Dir(\gamma)$

2. For each document $m \in \{1, \ldots, M\}$

i. Draw a topic distribution per document, $\theta_m \sim Dir(\alpha)$

ii. For each word position n in document m, where $n \in \{1, \ldots, N_m\}$

a) Draw a topic $z_{m,n} \sim Mult(\theta_m)$

b) Draw an LR-set $s_{m,n} \sim Mult(\varphi_{z_{m,n}})$

c) Emit word $w_{m,n} \sim Mult(\eta_{z_{m,n},s_{m,n}})$

3.3.2 Dealing with Wrong Knowledge

Since our LR-sets are general knowledge from online dictionaries, some LR-sets do not make sense in a particular domain. For example, {card, bill} is a correct LR-set in the domain "Restaurant", but unsuitable in the domain "Camera." There are two major challenges here:

1. One issue is that there may be no correct LR-sets for a word in an application domain. That is, for a word w, all LR-sets containing w do not make sense (are wrong) for the domain. In this case, the model does not have any correct LR-sets to choose from. As a result, the assigned LR-sets to w will all be incorrect, leading to promotion of LR-sets that have words that are not semantically related in a particular domain.
2. The other issue is that an LR-set may be partially correct and partially incorrect in a domain, meaning that some words in the LR-set do not share a similar semantic meaning with other words in the same LR-set for a particular domain. For example, in the domain "Camera", we have an LR-set {picture, pic, flick} where *picture* shares a similar meaning with *pic*, but both of them have different semantic meaning from *flick*. In this case, when we promote this LR-set as in the model MDK-LDA, we may promote all the relationships inside it, including the wrong ones: *picture-flick* and *pic-flick*. As a result, it may lead to merging of words with different semantic meanings, which results in multiple sub-topics inside one topic.

To address these two challenges, we first propose a word correlation matrix $\boldsymbol{C}$ to estimate the correlation of words in the LR-sets using the given corpus (Section 3.3.2.1). Using this matrix, for each word w, we relax the constraints of all wrong LR-sets (the first challenge

above) by adding a singleton LR-set $\{w\}$ (Section 3.3.2.2). To deal with the second challenge, we scale the matrix $\boldsymbol{C}$ into a matrix $\boldsymbol{C'}$, which fits in the new *generalized Pólya urn* (GPU) model in GK-LDA (Section 3.3.2.3). This new matrix $\boldsymbol{C'}$ is used to design a new Gibbs sampler for the GK-LDA model (Section 3.3.3).

3.3.2.1 Word Correlation Matrix

Given a piece of knowledge (LR-set) itself, the model may not know whether it is correct or not. However, given a corpus, it is possible to validate the LR-sets through the corpus. If an LR-set has a reasonable support in the corpus, we will have some confidence in its usefulness in the domain represented by the corpus, and consequently we give it a higher weight for promotion. Since the topics found by LDA are a reasonable summary of the corpus and the top words (with high probabilities) under each topic are more likely to share some semantic similarity, we use the topic-word distribution from LDA to estimate word correlations in each LR-set towards a domain. The idea is that if two words in an LR-set are too far from each other (i.e., have very different probabilities) under the topics of LDA, they are more likely to have different semantic meanings, i.e., less correlated.

Algorithm 1 gives a detailed algorithm for computing word correlation matrix $\boldsymbol{C}$. Intuitively, top words, with higher probabilities under a topic, are more likely to represent the semantic concept of the topic while words with low probabilities contribute much less to the semantic concept. To compute the correlation of two words, we focus on the topics where the words have high probabilities. Algorithm 1 computes for all the word pairs (w, w') in each LR-set (lines 1 and 2). The word distribution under topic t is denoted by φ_t in LDA. For those pairs not in any

Algorithm 1 Compute Word Correlation Matrix $\boldsymbol{C}$

1: **for** each LR-set $s \in \{1, \ldots, S\}$ **do**
2: **for** each pair $(w, w') \in s$ **do**
3: $P_{max}(w) = \max_{t \in \{1,\ldots,T\}} \varphi_t(w)$;
4: $P_{max}(w') = \max_{t \in \{1,\ldots,T\}} \varphi_t(w')$;
5: **if** $P_{max}(w) > P_{max}(w')$ **then**
6: **Exchange** w and w';
7: **end if**
8: $t_{max} = \operatorname{argmax}_{t \in \{1,\ldots,T\}} \varphi_t(w')$;
9: $\boldsymbol{C}_{s,w',w} = C_{s,w,w'} = \frac{\varphi_{t_{max}}(w)}{\varphi_{t_{max}}(w')}$;
10: **end for**
11: **end for**
12: Return $\boldsymbol{C}$;

LR-set, their correlation is 0 (not shown in the algorithm) and we do not need to validate them. Lines 3 and 4 find the topics that the two words w and w' have the maximum probabilities respectively. Lines 5-7 enforce that word w has a lower (or equal) maximum probability than w', restricting their ratio to be not larger than 1. Line 8 finds the topic that word w' has the maximum probability, and the ratio of probabilities of both words under this topic is estimated as the correlation (Line 9). The idea is that the ratio of word probabilities under this topic is a good indicator of the semantic correlation of the two words. Although this word correlation estimation may not be perfect due to the imperfect topic-word distributions from LDA, our experiments show that it is effective in solving the two issues discussed in Section 3.3.2.

3.3.2.2 Relaxing Wrong LR-sets

In order to deal with the first challenge mentioned in Section 3.3.2, we need to design a function to estimate the quality of an LR-set s toward a word w. Since the quality of s depends on the correlations of words inside it with w, we can estimate the quality of LR-set s towards w based on the word correlation matrix $\boldsymbol{C}$ as follows:

$$Q(s,w) = \begin{cases} \max\limits_{w' \in s, w \neq w'} \boldsymbol{C}_{s,w',w} & w \in s \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Basically, the quality function of LR-set s towards word w is the maximum correlation between any word w' ($w' \in s$ and $w' \neq w$) and w based on $\boldsymbol{C}$. This quality function can give us some hints as to which LR-sets are more likely to be correct or incorrect. We set a threshold ε such that if the quality of an LR-set s towards a word $w \ni s$ is less than ε, this LR-set s is estimated to be wrong (or low-quality) towards w in the domain. Following the first challenge, if all LR-sets of word w are estimated to be wrong, we need to add an alternative LR-set to give the model a right LR-set to choose. In this case, a singleton LR-set (i.e., $\{w\}$) is added to relax the LR-set constraint. If w has any LR-set with its quality value greater than ε, the singleton set $\{w\}$ is not added. This preprocessing ensures that the model can have at least one reasonable LR-set to assign to each word. However, note that, the estimated wrong LR-sets are not removed because the estimation above on topics generated by LDA may not be perfect.

3.3.2.3 Incorporating Correlation Matrix

Due to the power-law characteristics of natural language (Zipf, 1932), most words are rare and will not co-occur with most other words regardless of their semantic similarity. If some rare words share the same LR-set with some high-frequency words, the high-frequency words will be smoothed dramatically due to the hyperparameter γ which causes the *adverse effect issue*. For example, in the domain "Camera", the word *light* is an important word with its semantic meaning correlated with the domain. However, the words *brightness* and *luminousness* in the LR-set {light, brightness, luminousness} can harm *light* due to their infrequency. Since words in the LR-set are supposed to share some similar semantic meaning, if we see one of them, it is reasonable to expect higher probability of seeing any of the others. For the above LR-set, if *brightness* is seen in topic t, there is a higher chance of seeing both *light* and *luminousness* under topic t. To encode this characteristic, we use the *generalized Pólya urn* (GPU) model (Mahmoud, 2008), where objects of interest are represented as colored balls in an urn.

The Pólya urn model involves an urn containing balls of different colors. At discrete time intervals, balls are added or removed from the urn according to their color distributions.

In the simple Pólya urn (SPU) model, a ball is first drawn randomly from the urn and its color is recorded, then that ball is put back along with a new ball of the same color. This selection process is repeated and the contents of the urn change over time, with a self-reinforcing property sometimes expressed as "the rich get richer". SPU is actually exhibited in the Gibbs sampling for LDA.

The *generalized Pólya urn* (GPU) model differs from the SPU model in the replacement scheme during sampling. Specifically, when a ball is randomly drawn, certain numbers of additional balls of each color are returned to the urn, rather than just two balls of the same color as in SPU. In our case, the similarity of colors, which are terms, is indicated by the fact that they are from the same LR-set.

We now deal with the second challenge mentioned above, i.e., the partial incorrect LR-sets. In MDK-LDA (Chen et al., 2013d), when a word is drawn, all other words inside the LR-set will be put back equally according to a matrix, which promotes the LR-set as a whole, i.e., promoting every word inside it. Now we want to use the word correlation values of $\boldsymbol{C}$ to help determine the number of balls to put back which reduces the undesirable effects of wrong relationships in an LR-set. For this purpose, we scale $\boldsymbol{C}$ to $\boldsymbol{C}'$ as follows, which will be incorporated in the GK-LDA model.

$$\boldsymbol{C}'_{s,w',w} = \begin{cases} 1 & w \in s, w' \in s, w = w' \\ \tau \times \boldsymbol{C}_{s,w',w} & w \in s, w' \in s, w \neq w' \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

The coefficient τ governs the scale of correlation corresponding to the hyperparameters β and γ in the model. The value of τ will be discussed in Section 3.4. With the matrix $\boldsymbol{C}'$, we can design a new GPU model, i.e., drawing word w will not only increase the probability of seeing w, but also discriminatively increase the probability of seeing every correlated word with w represented by $\boldsymbol{C}'$. Following the example of LR-set $s = \{picture, pic, flick\}$ in Section 3.3.2,

since *picture* and *pic* are semantically related in the domain "Camera" (in other words, the relationship *picture-pic* is correct), they tend to have reasonable high co-occurrence in the corpus and hence LDA is likely to put them together under the same topic. On the other hand, *flick* is semantically different from both *picture* and *pic*, and thus LDA may put *flick* under a different topic. As a result, $\boldsymbol{C}'_{s,picture,pic}$ will be much larger than $\boldsymbol{C}'_{s,picture,flick}$ and $\boldsymbol{C}'_{s,pic,flick}$. In the GPU model of GK-LDA, seeing the word *picture* and *pic* will promote each other a lot, but promote the word *flick* very little, which is consistent with our aim to merge the semantically related words while separating semantically different words.

3.3.3 Inference

In this subsection, we introduce the Gibbs sampler for our GK-LDA model.

In topic models, collapsed Gibbs sampling (Griffiths and Steyvers, 2004), one of Markov Chain Monte Carlo (MCMC) methods (Robert and Casella, 2004), is a standard procedure for obtaining a Markov chain over the latent variables in the model. In GK-LDA, the latent variables (i.e., latent topic z and latent LR-set s) are jointly sampled, which gives us a blocked Gibbs sampler. An alternative way is to perform hierarchical sampling (sample z and then s). However, (Rosen-Zvi et al., 2010) argues that when the latent variables are highly related, blocked samplers improve convergence of the Markov chain and also reduce autocorrelation.

As in Section 3.3.2, the GK-LDA model employs the GPU model to promote each correlated word represented by the matrix $\boldsymbol{C}'$. However, the GPU model is nonexchangeable, meaning that the joint probability of the words in any given topic is not invariant to the permutation of those words. Inference of z and s can be computationally expensive due to the non-exchangeability

of words. We take the approach of (Mimno et al., 2011) which approximates the true Gibbs sampling distribution by treating each word as if it were the last. When sampling a word w, we first promote the LR-set of it as a whole. Then, each word in the LR-set is promoted based on its word correlation with w according to $\boldsymbol{C'}$. The idea is that if a word w' is more correlated with w, it should be promoted more when w is seen, pushing them into the same topic. Note that promotion in the GPU model is achieved by putting back balls of the corresponding colors into the urn. Denoting the random variable $\{z, s, w\}$ by singular subscripts $\{z_i, s_i, w_i\}$, where i denotes the variable corresponding to each word in each document in the corpus, the conditional probability to assign a topic t and an LR-set s (containing the word w_i) to the word w_i is given by:

$$
\begin{aligned}
P(z_i = t, s_i = s | \boldsymbol{z}^{-i}, \boldsymbol{s}^{-i}, \boldsymbol{w}, \alpha, \beta, \gamma, \boldsymbol{C'}) \propto & \frac{n^{-i}_{m,t} + \alpha}{\sum_{t'=1}^{T}(n^{-i}_{m,t'} + \alpha)} \\
\times \frac{\sum_{w'=1}^{V}\sum_{v'=1}^{V} \boldsymbol{C'}_{s,v',w'} \times n^{-i}_{t,s,v'} + \beta}{\sum_{s'=1}^{S}(\sum_{w'=1}^{V}\sum_{v'=1}^{V} \boldsymbol{C'}_{s',v',w'} \times n^{-i}_{t,s',v'} + \beta)} & \times \frac{\sum_{w'=1}^{V} \boldsymbol{C'}_{s,w',w_i} \times n^{-i}_{t,s,w'} + \gamma_s}{\sum_{v'=1}^{V}(\sum_{w'=1}^{V} \boldsymbol{C'}_{s,w',v'} \times n^{-i}_{t,s,w'} + \gamma_s)}
\end{aligned}
\tag{3.3}
$$

where n^{-i} denotes the count excluding current assignment of z_i and s_i, i.e., $\boldsymbol{z}^{-i}$ and $\boldsymbol{s}^{-i}$. $n_{m,t}$ denotes the number of occurrences that topic t was assigned to word tokens in document m. $n_{t,s}$ denotes the count that LR-set s occurs under topic t. $n_{t,s,v}$ refers to the number of times that word v is assigned with LR-set s under topic t.

3.4 Experiments

We now evaluate the proposed GK-LDA model, and compare it with baseline models: LDA (Blei et al., 2003), LDA with GPU (denoted as LDA-GPU) (Mimno et al., 2011) and DF-LDA (Andrzejewski et al., 2009), MDK-LDA(b) (Chen et al., 2013d) and MDK-LDA (Chen et al., 2013d). LDA is the basic knowledge-free unsupervised topic model. LDA-GPU applied GPU in LDA using co-document frequency. DF-LDA is perhaps the most well-known knowledge-based model which introduced must-links and cannot-links. It is also a natural fit for our proposed model as a must-link and an LR-set share the similar notion, i.e., they both aim at constraining the words in them to appear under the same topic. Note that existing models typically assume that the knowledge is correct and to our knowledge there is no prior work in topic modeling that can deal with wrong knowledge explicitly. Our proposed GK-LDA model can deal with wrong knowledge. We will see in Sections 3.4.2 and 3.4.3 that this capability of GK-LDA results in far better results than existing state-of-the-art models.

In Section 3.4.1, we describe the datasets and experimental settings. In Section 3.4.2, we evaluate our framework objectively using the Topic Coherence metric (Mimno et al., 2011) and KL-Divergence. Further, in Section 3.4.3.1, we report the human evaluation results by working with two judges who are familiar with the Amazon products and reviews. Last, we show the qualitative results with some example topics from different models in Section 3.4.3.2.

3.4.1 Datasets and Settings

Datasets: Since LR-sets and the proposed framework are domain independent mechanisms for finding topics from text collections, we use multiple datasets from different domains of online

reviews for our evaluation. We collected reviews from four domains from Amazon.com. Each domain collection (or corpus) contains 500 reviews. The four domains are "Camera & Photo", "Cell Phones & Accessories", "Gourmet Food & Grocery", and "Computers & Accessories". For easy presentation, we simply use "Camera", "CellPhone", "Food", and "Computer" to denote the four domain corpora respectively.

Pre-processing: We ran the Stanford Parser [1] to perform sentence detection, lemmatization, and POS tagging. Then, punctuations, stop words [2], numbers and words appearing less than 5 times in each corpus were removed. For each domain, the domain name was also removed as it appears very frequently and co-occurs with most words in the corpus, leading to high similarity among topics. Our LR-sets depend on POS tags of words. In this work, we only use nouns and adjectives to produce LR-sets since they are the main parts of the topics. Verbs have a high level of noise. Note that duplicate LR-sets have been removed.

Sentences as documents: As pointed out in (Titov and McDonald, 2008b), when standard topic models are applied to reviews, they tend to produce topics that correspond to global properties of product, which lead to large overlappings among topics. Since applying topic models to reviews mainly aims to find different aspects or features (as topics) of products (Jo and Oh, 2011; Mukherjee and Liu, 2012a; Titov and McDonald, 2008b; Zhao et al., 2010), using individual reviews for modeling is not very effective (Titov and McDonald, 2008b). Al-

---

[1] `http://www-nlp.stanford.edu/software/corenlp.shtml`

[2] `http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop`

though there are models dealing with sentences in complex ways (Jo and Oh, 2011; Titov and McDonald, 2008b), we take the approach in (Brody and Elhadad, 2010), dividing each review into sentences and treating each sentence as a document. Sentences can be used by all three baselines without any change to their models. Although the relationship between sentences of a review is lost, the data is fair to all models.

Parameter settings: For all models, posterior inference was drawn after 1000 Gibbs iterations with an initial burn-in of 100 iterations. For all models, we set $\alpha = 1$ and $\beta = 0.1$. We found that small changes of α and β did not affect the results much, which was also reported in (Jo and Oh, 2011) who also used online reviews. For the number of topics, we tried different values (see Section 3.4.2.1). Note that it is difficult to know the exact number of topics. While non-parametric Bayesian approaches (Teh et al., 2006) aim to estimate the number of topics from the corpus, they are often sensitive to hyperparameters (Heinrich, 2009). In this work, the heuristic values obtained from our experiments produced good results.

For DF-LDA, we followed the definition of must-link to generate must-links from LR-sets. LR-sets don't contain cannot-link knowledge. Note that the generated must-links contain wrong knowledge due to the issue of multiple senses, which degrades the performance of DF-LDA as we will see in Sections 3.4.2 and 3.4.3. We then ran DF-LDA (implementation downloaded from its authors website) while keeping the parameters as proposed in (Andrzejewski et al., 2011) (we also experimented with different parameter settings but they did not produce better results). For our framework, we empirically set $\lambda = 2000$. For the threshold of ε, in GK-LDA, we estimated it using some labeled LR-sets in a development corpus, "Watch", which was not

used in the evaluation (different from the four domains used). Based on the labeled data, we empirically chose the threshold $\varepsilon = 0.07$, meaning that if the quality of LR-set is lower than this value, we will add a singleton set as described in Section 3.3.2.2. We then averaged the word correlation values (Algorithm 1) of word pairs in the estimated correct LR-sets to set $\tau = 2$ in Equation 3.2. Although these three parameters come from the domain "'Watch", we use them for all four other domains.

3.4.2 Objective Evaluation

In this section, we evaluate our framework objectively. Topic models are often evaluated using perplexity on held-out test data. However, the perplexity measure does not reflect the semantic coherence of individual topics learned by a topic model (Newman et al., 2010b). Recent research has shown potential issues with perplexity as a measure: (Chang et al., 2009) suggested that the perplexity measure can sometimes be contrary to human judgments. Also, perplexity does not really reflect our goal of finding coherent topics with accurate semantic clustering. It only provides a measure of how well the model fits the data. Thus, we choose two evaluation metrics, Topic Coherence and KL-Divergence, which directly evaluate our framework on topic interpretability and topic distinctiveness (Kawamae, 2010; Mukherjee and Liu, 2012b). We also report statistical significance of improvements of our framework calculated based on paired t-test.

Topic Coherence: The *Topic Coherence* metric (Mimno et al., 2011) (also called UMass measure (Stevens and Buttler, 2012) was proposed for assessing topic quality. The metric relies upon word co-occurrence statistics within the documents, and does not depend on external

resources or human labeling. (Mimno et al., 2011) shows that topic coherence is highly consistent with human labeling. Higher Topic Coherence score indicates higher quality of topic, i.e. better topic interpretability. The definition of Topic Coherence is stated as below:

$$\text{Topic Coherence}(t; V^{(t)}) = \sum_{i=2}^{N} \sum_{j=1}^{i-1} log \frac{\#D(v_i^{(t)}, v_j^{(t)}) + 1}{\#D(v_j^{(t)})} \tag{3.4}$$

where $V^{(t)} = (v_1^{(t)}, v_2^{(t)}, \ldots, v_N^{(t)})$ is a list of N most probable words in topic t. $\#D(v)$ is the document frequency of word v, i.e., the number of documents containing at least one occurrence of word v. $\#D(v, v')$ is the co-document frequency of word v and v', (i.e., the number of documents containing both v and v').

KL-Divergence: Another important metric for topic models is topic distinctiveness (Kawamae, 2010; Mukherjee and Liu, 2012b). We want to evaluate how distinctive the discovered topics are. To measure the distinctiveness, we use KL-Divergence as in (Kawamae, 2010; Mukherjee and Liu, 2012b). Since KL-Divergence is asymmetric, we compute its values between all pairs of topics and average them to get the average KL-Divergence. Clearly, for more distinctive topic discovery and better topic quality, it is desirable to have larger average KL-Divergence. The KL-Divergence of word distributions P_t and $P_{t'}$ under two topics t and t' is defined as:

$$KL(P_t, P_{t'}) = \sum_{w} ln \left(\frac{P_t(w)}{P_{t'}(w)} \right) P_t(w) \tag{3.5}$$

where $P_t(w)$ is the probability of word w under topic t.

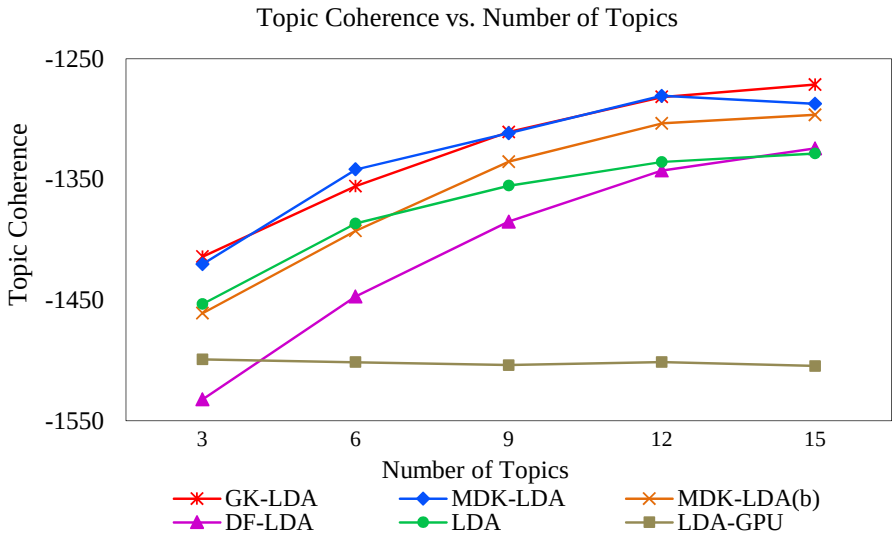

Figure 2. Average Topic Coherence score of each model given different number of topics.

3.4.2.1 Effects of Number of Topics

Since the models in our experiments are all parametric topic models, we first compare the performance of each model given different number of topics. Figure 2 and Figure 3 show the average Topic Coherence score and KL-Divergence (over all domains) of each model given different number of topics. We can make the following observations:

1. From the Topic Coherence results, given different number of topics, our framework consistently achieve higher Topic Coherence scores than the baseline models. Among them, GK-LDA performs best with the highest Topic Coherence score. GK-LDA and MDK-LDA improves significantly ($p < 0.001$) over the three baseline models and MDK-LDA(b).

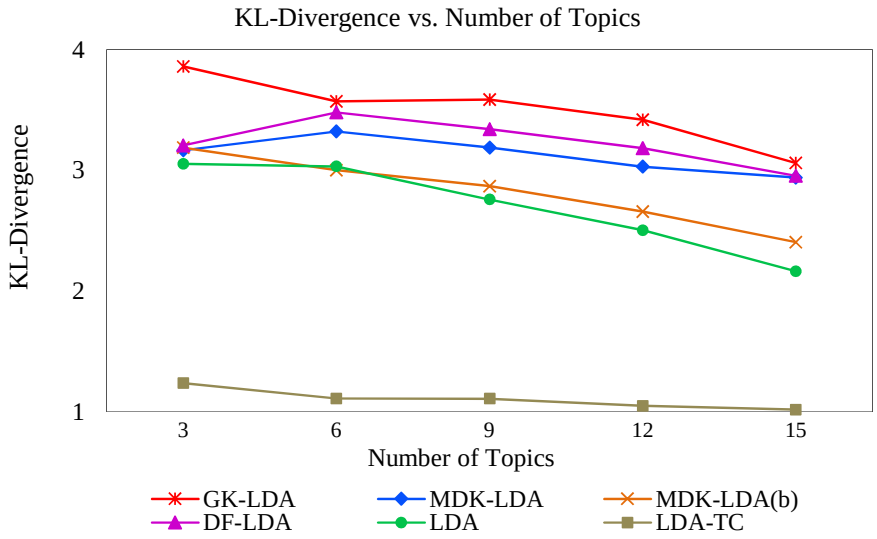

Figure 3. Average KL-Divergence of each model given different number of topics.

2. From the KL-Divergence results, given different number of topics, GK-LDA produces the most distinctive topics with the largest KL-Divergence. GK-LDA improves significantly over DF-LDA ($p < 0.03$) and all other models ($p < 0.001$). Note that GK-LDA performs much better than MDK-LDA in terms of KL-Divergence.
3. Although DF-LDA has larger KL-Divergence than MDK-LDA and LDA, its Topic Coherence score is not as high as MDK-LDA and LDA. The wrong knowledge does degrade the performance of DF-LDA due to its incapability of handling wrong knowledge. In Section 3.4.2.2, we further analyze the effects of knowledge on DF-LDA in more details.
4. LDA-GPU does not produce as good topics as other models with the lowest Topic Coherence score and smallest KL-Divergence. As frequent words usually have high co-document

frequency with many other words, the frequent words are ranked top in many topics. This shows that general lexical knowledge is more effective than co-document frequency without knowledge as was proposed in (Mimno et al., 2011).

5. In general, with more topics, the Topic Coherence score increases while KL-Divergence decreases, which is in accordance with results in (Kawamae, 2010; Mukherjee and Liu, 2012b). We found that when T is larger than 15, topics became more and more similar with each other (average KL-Divergence < 2.5). In addition, 15 topics of each model for four domains are a reasonable amount of work for human evaluation. Thus, we fix $T = 15$ to compare the detailed Topic Coherence results (in Section 3.4.2.2) and human evaluation results (in Section 3.4.3).

3.4.2.2 Effects of Knowledge

In order to see the effects and sensitivity to general lexical knowledge, we show the detailed Topic Coherence score of $T = 15$ in Figure 4. Again, we can find that the GK-LDA model has the highest scores across four domains, meaning that it produces the most coherent topics. We can also see that DF-LDA performs better than LDA in the domain "Food" and "Computer" but worse in the domain "Camera" and "Cellphone". In order to fully understand it, we investigated the knowledge in each domain. We found that the knowledge is very different in the four domains:

1. In the domains "Food" and "Computer", the knowledge is simpler with one word usually expressing one meaning/sense. Also, most of the wrong pieces of knowledge in these two domains only contain infrequent words. For example, {menu, bill} is not a suitable

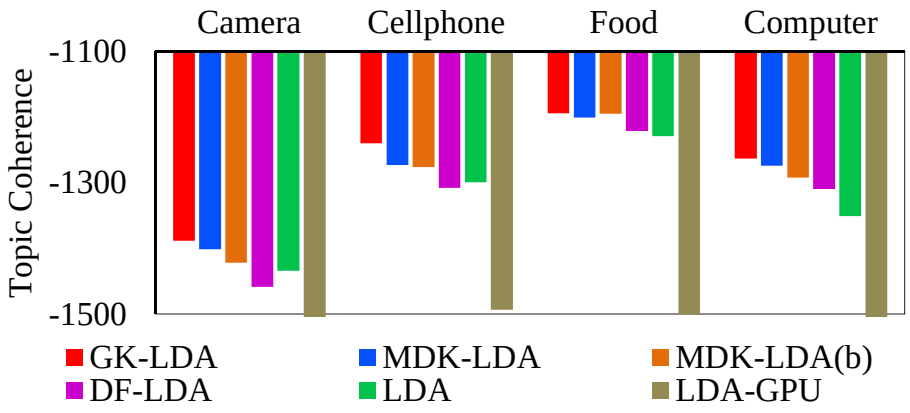

Figure 4. Detailed Average Topic Coherence score given 15 topics.

knowledge for the domain "Computer". But since both *menu* and *bill* are very rare words in this domain, their probabilities under each topic are already very low (< 0.0001). Even if DF-LDA makes their probability closer, the redistribution of probability mass does not influence the probability of frequent (or important) words. Thus, the benefits of correct knowledge outweigh the costs of wrong knowledge and hence DF-LDA performs better than LDA in the domains "Food" and "Computer". However, since there is still a small amount of knowledge involving multiple senses (and may be wrong) which is harmful to DF-LDA, it is not performing as well as our framework.

2. On the other hand, in the domains "Camera" and "Cellphone", the knowledge is more complicated with the words having multiple senses (resulting in wrong knowledge) and

mixing of frequent and infrequent words. In this case, DF-LDA performs worse than LDA due to its inability to deal with them.

In summary, we can conclude that our proposed framework is highly effective in producing distinctive topics where each topic is highly coherent compared to the baseline models.

3.4.3 Human Evaluation

3.4.3.1 Quantitative Results

Since our aim is to make topics more interpretable and conform to human understanding, we worked with two judges who are familiar with Amazon products and reviews to evaluate the models subjectively. Since topics from topic models are rankings based on word probability and we do not know the number of correct topical words, a natural way to evaluate these rankings is to use $Precision@n$ (or $p@n$) which was also used in (Mukherjee and Liu, 2012a; Zhao et al., 2010), where n is the rank position. We give $p@n$ for $n = 5, 10, 15$ and 20. There are two steps in human evaluation: Topic labeling and Word labeling.

Topic Labeling: We followed the instructions in (Mimno et al., 2011) and asked the judges to label each topic as *good* or *bad*. Each topic was presented as a list of 20 most probable words in descending order of their probabilities under that topic. The models which generated the topics for labeling were oblivious to the judges. In general, each topic was annotated as *good* if it had a few words coherently related to each other representing a semantic concept together; otherwise *bad*.

Word Labeling: After topic labeling, we chose the topics, which were labeled as good by both judges, as good topics. Then, we asked the two judges to label each word of the top 20

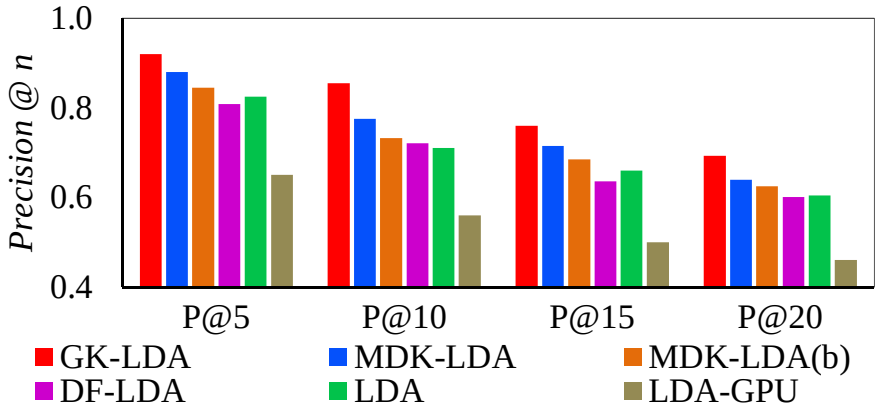

Figure 5. Average *Precision @ n* (*p @ n*) of good topics over all four domains.

words in these good topics. Each word was annotated as *correct* if it was coherently related to the concept represented by the topic; otherwise *incorrect*. Since judges already had the conception of each topic in mind when they were labeling topics, labeling each word was not very difficult.

Precision @ n: Figure 5 gives the average *precision@n* of all good topics over all four domains. We can make the following observations:

1. GK-LDA performs the best, improving LDA by more than 11% on average. The model successfully identifies and leverages the correct knowledge and also addresses the wrong knowledge in the form of LR-sets for each domain.

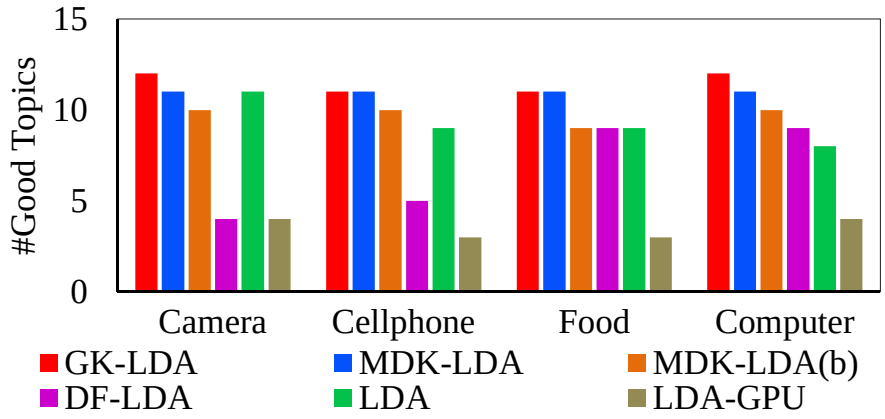

Figure 6. Number of good topics generated by each model.

2. MDK-LDA improves precision of MDK-LDA(b) by about 3% and LDA by about 5%. We can see that the promotion of LR-sets through GPU model is effective. But it still suffers from the wrong knowledge. MDK-LDA(b) improves the precision of LDA by only 2%.
3. DF-LDA performs slightly worse than LDA with less than 1%. This shows that not dealing with wrong knowledge can dramatically reduce the model's effectiveness, which is understandable because the wrong knowledge may guide the model mistakenly.
4. LDA-GPU does not perform well in our data due to its use of co-document frequency. GK-LDA performs much better. Again, this shows that LR-set knowledge is more effective than co-document frequency.

We can see that the human evaluation results are highly consistent with Topic Coherence and KL-Divergence results in Section 3.4.2. Upon significance testing of improvement of GK-LDA

and MDK-LDA over other models in Figure 5, GK-LDA and MDK-LDA improves significantly over all other models ($p < 0.005$).

Number of Good topics: Figure 6 shows the number of good topics discovered by each model in each domain. In general, GK-LDA can generate about 2 more good topics than LDA and more than 5 additional topics compared to DF-LDA and LDA-GPU. These are very important in practice. For DF-LDA, it discovered fewer good topics than LDA in the domains "Camera" and "Cellphone" but more good topics than LDA in the domain "Computer", which is consistent with the analysis in Section 3.4.2.2. We also found that all topics discovered by LDA, LDA-GPU and DF-LDA can be uncovered by MDK-LDA(b), MDK-LDA and GK-LDA. Thus, our framework not only produces additional good topics but also covers the good topics of the baseline models.

3.4.3.2 Qualitative Results

This section shows some qualitative results to give us an intuitive feeling of the results from different models. There are a large number of topics that GK-LDA makes major improvements. Due to space limitations, we can only show some examples. To further focus, we will just show some results of LDA and GK-LDA. The results from LDA-GPU and DF-LDA were inferior and even hard to match with topics found by the other models.

Table I shows six example topics and top ranked topical words from LDA and GK-LDA. Wrong topical words are in italic and marked red (we tried to find the best possible match for the models). We can see that GK-LDA produces much better topics. Since the labeling of topics and topical words are somewhat subjective, we do not expect everyone to agree with

Camera		Cellphone		Food		Computer	
Photographer		Price		Taste		Image Quality	
LDA	GK-LDA	LDA	GK-LDA	LDA	GK-LDA	LDA	GK-LDA
dslr	dslr	*product*	price	taste	salt	quality	resolution
point	*year*	price	cheap	salt	taste	picture	pixel
year	professional	*review*	money	*almond*	flavor	*easy*	quality
canon	amateur	*time*	expensive	*fresh*	tasty	high	image
photography	pro	*item*	cost	*pack*	delicious	*money*	picture
nikon	photography	*device*	cheaper	tasty	sweet	*inch*	*dead*
photographer	experience	money	inexpensive	*oil*	salty	movie	high
shoot	*month*	*star*	*shipping*	*roasted*	tasting	*price*	low
price	photographer	cheap	worth	pepper	spice	*problem*	higher
digital	*model*	*shipping*	dollar	*easy*	yummy	*size*	lower

TABLE I. Six example topics in four domains. The domain names are on the first row and the topic names (manually assigned) are on the second row. Errors are marked in red/italic).

the labeling, but we tried our best to have the consensus with two human judges. Clearly, the results in Table I do not tell all the story. We also want to highlight several important points below.

1. One of the most common and important topics in online reviews is the *price* of products. However, out of the four domains, only in the domain "Food", LDA was able to find the topic *price* with a reasonable precision. In other domains, the *price* related topical words were mixed with all kinds of other topics by LDA. We show one example in Table I (in the column "Cellphone"), where the best *price* related topic of LDA is still poor. We believe that LDAs inability is mainly due to the fact that in English, sentences like "The price of this phone is expensive." are relatively rare. Thus, there is probably no co-occurrence of *price* and *expensive* (or other adjectives related to *price*, e.g., *cheap*) within a sentence.

Our adjective-attribute knowledge is very effective in this case, discovering the good *price* topic (see "GK-LDA" in the column "Cellphone").

2. LDA tends to split one topic into multiple topics, i.e., the topical words for a semantic topic appear at the top ranked positions of several topics. GK-LDA is much better in this regard. The results in Table I also show that.
3. There are also many other examples we could not list here due to space limitations. For example, in the food domain, GK-LDA discovered the topic *Healthy Eating: protein, fat, fiber, healthy, nutrient, nutrition, vitamin, magnesium.* These words are all highly coherent. The best one that LDA could find were: *time, snack, food, point, healthy, calorie, weight, year.* In the computer domain, GK-LDA was able to find the topic of *Program Execution: game, slow, word, fast, web, star, speed, slower.* Most of the words here are highly relevant except star. LDA was unable to find any topic related.

In summary, we can say that GK-LDA produces much better results, both in terms of precision and the number of good topics, which indicates that our proposed framework of exploiting general lexical knowledge is highly promising.

3.5 Summary

We proposed a novel approach of utilizing the general knowledge of lexical semantic relations in topic models in order to produce more coherent topics. In any language, there is a vast amount of such knowledge stored in dictionaries. Since such knowledge is domain independent, it should be applicable to any application domain. However, due to multiple meanings or senses of a word, some knowledge may not be suitable for a particular application domain.

We proposed the GK-LDA model as a comprehensive framework to effectively leverage general knowledge in topic models and to also deal with the wrong knowledge. To our knowledge, this is the first work that proposes a principled model to systematically incorporate the general knowledge to produce more coherent topics. What is even more important is that our proposed framework can automatically deal with wrong knowledge without any user input.

CHAPTER 4

LIFELONG TOPIC MODELING

(This chapter includes and expands on my paper previously published in *Zhiyuan Chen and Bing Liu. Topic Modeling using Topics from Many Domains, Lifelong Learning and Big Data. In ICML 2014, pages 703-711.*)

Chapter 3 introduced the knowledge-based topic models (KBTM) which are able to leverage the knowledge where the knowledge comes from domain experts such as online dictionaries. However, there are 3 major shortcomings in that approach:

1. It is usually hard and tedious to obtain knowledge from domain experts. We may not be able to find experts for every domain we are interested in.
2. The existing works typically assume that the knowledge has little or no noise. This is impractical as the domain experts may make mistakes. Sometimes, the knowledge they provide could be biased as well.
3. One of the major purposes of topic model is to help user or expert to quickly understand a large number of document collections. That means the expert may not have any idea about what is mentioned in the document corpus. However, the existing knowledge-based topic models ask the experts to provide knowledge in order to generate interpretable topics while the experts want to run the model to study the topics first.

To address these shortcomings, in this chapter, we show that much of the prior knowledge from the user can actually be mined automatically (without user input) from a large amount

of data in many domains. In most cases, such data is readily available on the Web. This is possible because although every domain is different, there is a fair amount of concept or topic overlapping across domains. For example, every product review domain probably has the topic *price*, reviews of most electronic products share the topic of *battery* and reviews of some products share the topic of *screen*. Topics produced from a single domain can be erroneous (i.e., a topic may contain some irrelevant words in its top ranked positions), but if we can find a set of shared words among some topics generated from multiple domains, these shared words are more likely to be coherent for a particular topic. They can serve as a piece of prior knowledge to help topic modeling in each of these domains (i.e., *past domains* in LML) or in a new domain (i.e., *current domain* in LML).

For example, we have product reviews from three domains. We run LDA to generate a set of topics from each domain. Every domain has a topic about *price*, which is listed below with its top four words (words are ranked based on their probabilities under each topic):

Domain 1: *price*, *color*, *cost*, *life*

Domain 2: *cost*, *picture*, *price*, *expensive*

Domain 3: *price*, *money*, *customer*, *expensive*

These topics are not perfect due to the incoherent words: *color*, *life*, *picture*, and *customer*. However, if we focus on those topical words that appear together in the same topic across at least two domains, we find the following two sets:

{*price*, *cost*} and {*price*, *expensive*}.

We can see that the words in such a set are likely to belong to the same topic. Such, {*price*, *cost*} and {*price*, *expensive*}, can serve as *prior knowledge*, which we call *prior knowledge sets* (or *pk-sets* for short), in a KBTM to improve the output topics for each of the three domains or a new domain. Note that pk-sets have the same structure as LR-sets in Chapter 3. We use the name pk-sets to emphasize that it is automatically extracted from prior or past domains. For example, after running a KBTM on the reviews of Domain 1, we may find the new topic: *price*, *cost*, *expensive*, *color*, which has three coherent words in the top four positions rather than only two words as in the original topic. This represents a good topic improvement.

The above discussion suggests a three-step approach to our task. Given a set of document corpora $\boldsymbol{D} = \{D_1, \ldots, D_n\}$ from n domains, step 1 runs a topic model (e.g., LDA (Blei et al., 2003)) on each domain $D_i \in \boldsymbol{D}$ to produce a set of topics S_i. We call these topics the *prior topics* (or *p-topics* for short). Step 2 mines a set of pk-sets (prior knowledge sets) K from all the p-topics $S = \cup_i S_i$. We call S *Topic Base*. Step 3 uses the pk-sets K in a KBTM to generate topics for a test document collection D^t (D^t may or may not be from $\boldsymbol{D}$).

To further improve, our proposed method embeds step 2 in step 3 so that the mining of prior knowledge is targeted and thus more accurate. Specifically, we first run a KBTM on the test document collection D^t without any knowledge (which is equivalent to LDA) until its topics (denoted by A^t) stabilize. To distinguish these topics from p-topics, we call these topics the *current topics* (or *c-topics* for short). For each c-topic $a_j \in A^t$, we then find a set of matching or similar p-topics M_j^t in S (the set of all p-topics). The intuition here is that these matching p-topics M_j^t are targeted with respect to a_j and should provide high quality knowledge for a_j.

We then mine M_j^t to generate pk-sets K_j^t for c-topic a_j. After that, we continue the execution of the KBTM on D^t, which is now guided by the new pk-sets K^t (which is the union of all K_j^t), in order to generate better c-topics (We will present our formal algorithm in Section 4.1).

Regarding knowledge-based topic models, we could not use the existing ones because they typically assume the given prior knowledge to be correct (See Section 2.3). There is clearly no guarantee that the automatically mined pk-sets are all correct for a domain. First, due to wrong topics in S (Topic Base) or mining errors, the words in a pk-set may not belong to the same topic in general. Second, the words in a pk-set may belong to the same topic in some domains, but not in others due to the domain diversity. Thus, to apply such knowledge in modeling, the model must deal with possible errors in pk-sets. We propose a new fault-tolerant knowledge-based model to deal with the problem. It can exploit the automatically mined prior knowledge and deal with incorrect knowledge to produce superior topics.

Due to this ability of using topics (or knowledge) generated from other domains to help modeling in the current domain, this work offers two novel capabilities: (1) lifelong machine learning (LML) and (2) modeling with big data. We call the proposed model *Lifelong Topic Model* (*LTM*) (Chen and Liu, 2014b).

4.1 Overall Algorithm

This section first introduces the proposed overall algorithm to leverage lifelong machine learning for topic modeling. It then introduces a lifelong machine learning approach for topic modeling. The algorithm consists of two general steps:

Algorithm 2 PriorTopicsGeneration($\boldsymbol{D}$)

```
1: for r = 0 to R do
2:    for each domain corpus D_i ∈ D do
3:       if r = 0 then
4:          S_i ← LDA(D_i);
5:       else
6:          S_i ← LTM(D_i, S);
7:       end if
8:    end for
9:    S ← ∪_i S_i;
10: end for
```

Step 1 (**prior topic generation**): Given a set of document collections $\boldsymbol{D} = \{D_1, \ldots, D_n\}$ from n domains, Algorithm 2 **PriorTopicsGeneration** runs LDA on each domain $D_i \in \boldsymbol{D}$ to produce a set of topics S_i (lines 2 and 4). The resulting topics from all n domains are unionized together to produce the Topic Base S (line 9) which is the set of all prior topics from $\boldsymbol{D}$. We call S the *prior topic* (or *p-topic*) set. The *p-topics* in S are used in the proposed model LTM to generate the prior knowledge (Line 6).

Iterative improvement: The above process can actually be run iteratively to improve the p-topics in S. That is, S from the previous iteration can help generate better topics from D using the proposed LTM model for the next iteration. This process is reflected in lines 1, 5-7 and 10. We will examine the performance of different iterations in Section 4.3.2. Note that from the second iteration ($r \geq 1$), LTM is used (line 6).

Step 2 (**testing**): Given a test document collection D^t and a prior Topic Base S, this step employs the proposed topic model LTM (Algorithm 3) to generate topics from D^t. To

distinguish these topics from p-topics, we call them the *current topics* (or *c-topics* for short). LTM is given in Algorithm 3, which we will detail in the next section. Note that D^t can be a document collection from D or a new domain. This can be seen as two ways of using the proposed algorithm: (1) the topics from D^t can be part of p-topics in S used in knowledge mining in LTM, and (2) not part of p-topics in S. We will experiment with these two settings in Section 4.3.

Lifelong Machine Learning: The above approach naturally enables lifelong machine learning. We denote the knowledge base as KB which is Topic Base (i.e., the p-topic set) generated by a system (or even specified by the user), and LTM is the learning algorithm. Given a new learning task G (e.g., topic modeling in our case) with its data (e.g., D^t), lifelong machine learning works in two main phases.

Phase 1: **Learning with prior knowledge**: This is essentially Step 2 above using LTM, which solves two sub-problems. Step 1 is the initialization.

a) *Identify shared knowledge for task G.* Identify the part of the knowledge in KB that can be used for G. In our case, the shared knowledge is K^t in Algorithm 3, which is mined from Topic Base S.

b) *Knowledge-based learning.* Learn for task G with the help of K^t using a learning algorithm. In our case, it is the GibbsSampling function in line 4 of LTM (Algorithm 3).

Phase 2: **Knowledge retention and consolidation**. In our case, we simply add the topics from G to S (Topic Base) if G is a new task. If G is an old task, we replace its topics in S. This is not included in Algorithms 2 or 3, but can be added easily.

Algorithm 3 LTM(D^t, S)

1: $A^t \leftarrow$ GibbsSampling(D^t, $\emptyset$, N); // Run N Gibbs iterations with no knowledge (equivalent to LDA).
2: **for** $i = 1$ **to** N **do**
3: $K^t \leftarrow$ KnowledgeMining(A^t, S);
4: $A^t \leftarrow$ GibbsSampling(D^t, K^t, 1); // Run with knowledge K^t.
5: **end for**

Algorithm 4 KnowledgeMining(A^t, S)

1: **for** each p-topic $s_k \in S$ **do**
2: $j^* = \min_j$ KL-Divergence(a_j, s_k) for $a_j \in A^t$;
3: **if** KL-Divergence(a_{j^*}, s_k) $\leq \pi$ **then**
4: $M^t_{j^*} \leftarrow M^t_{j^*} \cup \{s_k\}$;
5: **end if**
6: **end for**
7: $K^t \leftarrow \cup_{j^*}$ FIM($M^t_{j^*}$); // Frequent Itemset Mining.

4.2 LTM Model

Like many topic models, LTM uses Gibbs sampling for inference (Griffiths and Steyvers, 2004). Its graphical model is the same as LDA, but LTM has a very different sampler which can incorporate prior knowledge and also handle errors in the knowledge.

LTM works as follows: It first runs the Gibbs sampler of LTM for N iterations (or sweeps) to find a set of initial topics A^t from D^t with no knowledge (line 1, Algorithm 3). Since there is no knowledge, the sampler is equivalent to that of LDA. It then makes another N Gibbs sampling sweeps (lines 2-5). But in each of these new sweeps, it first mines pk-sets K^t for all topics in A^t using the function **KnowledgeMining** (Algorithm 4, detailed in Section 4.2.1) and then uses

K^t to generate a new set of topics A^t from D^t. Note that to make the algorithm more efficient, we do not need to mine knowledge for every sweep (see Section 4.3.5). Below, we focus on the knowledge mining function of LTM. The Gibbs sampler will be given in Section 4.2.2.

4.2.1 Knowledge Mining

The knowledge-mining function is given in Algorithm 4. For each p-topic $s_k \in S$, it finds the best matching (or the most similar) c-topic a_{j^*} in the c-topic set A^t (line 2). $M^t_{j^*}$ is used to find matching pk-sets for c-topic a_{j^*} (line 7). We find the matching p-topics for each individual c-topic a_{j^*} because we want a_{j^*} specific p-topics for more accurate knowledge set mining. Below, we present the algorithms for topic match and knowledge set mining.

Topic matching (lines 2-5, Algorithm 4): To find the best matching for s_k with a c-topic a_{j^*} in A^t, we use KL Divergence to compute the difference of the two distributions (lines 2 and 3). In this paper, we use Symmetrised KL (SKL) Divergence for all divergence computing, i.e., given two distributions P and Q, the divergence is calculated as:

$$SKL(P,Q) = \frac{KL(P,Q) + KL(Q,P)}{2} \tag{4.1}$$

$$KL(P,Q) = \sum_i ln\left(\frac{P(i)}{Q(i)}\right) P(i) \tag{4.2}$$

We denote the c-topic with the minimum SKL Divergence with s_k as a_{j^*}. π is used to ensure the p-topics in $M^t_{j^*}$ are reasonably correlated with a_{j^*}.

Mine knowledge sets using frequent itemset mining (FIM): Given the p-topics in each matching set M_{j*}^t, this step finds sets of words that appear together multiple times in these p-topics. The shared words among matching p-topics across multiple domains are likely to belong to the same topic. To find such shared words in the matching set of p-topics M_{j*}^t, we use frequent itemset mining (FIM) (Agrawal and Srikant, 1994).

FIM is stated as follows: Given a set of transactions X, where each transaction $x_i \in X$ is a set of items. In our context, x_i is a set of top words of a p-topic (no probability attached). X is actually M_{j*}^t without lowly ranked words in each p-topic as only the top words are usually representative of a topic. The goal of FIM is to find every itemset (a set of items) that satisfies some user-specified frequency threshold (also called *minimum support*), which is the minimum number of times that an itemset should appear in X. Such itemsets are called *frequent itemsets*. In our context, a frequent itemset is a set of words that have appeared together multiple times in the p-topics of M_{j*}^t. Such itemsets are our prior knowledge pk-sets.

In this work, we use only frequent itemsets of length two, i.e., each pk-set has only two words. For example, {battery, life}, {battery, power}, {battery, charge}. Using two words in a pk-set is sufficient to cover the semantic relationship of words belonging to the same topic. Longer sets tend to contain more errors since some words in a set may not belong to the same topic as others. Such errors can hurt the downstream modeling.

4.2.2 Gibbs Sampler

This sub-section gives the Gibbs sampler of the LTM model, which differs from LDA as LTM needs additional mechanisms to leverage the prior knowledge and to also deal with wrong

knowledge during sampling. Below, we first discuss the techniques used for these two capabilities, and then present the final Gibbs sampler.

4.2.2.1 Incorporating Prior Knowledge and Dealing with Wrong Knowledge

As each pk-set reflects a possible semantic similarity relation between a pair of words, we again use the *generalized Pólya urn* (GPU) model (Mahmoud, 2008) to leverage this knowledge in Gibbs sampling to encourage the pair of words to appear in the same topic.

Under the GPU model, when a word w is assigned to a topic t, each word w' that shares a pk-set of topic t with w is also assigned to the topic t by a certain amount, which is decided by the matrix $A'_{t,w',w}$. Note that we use different notation for the promotion matrix from Chapter 3 to highlight the more advanced way to compute it in LTM. w' is thus promoted by w, meaning that the probability of w' under topic t is also increased. Here, a pk-set of a topic t means this pk-set is extracted from the p-topics matching with topic t.

The problem is how to set proper values for matrix $A'_{t,w',w}$. To answer this question, let us also consider the problem of wrong knowledge. Since the pk-sets are mined from p-topics in multiple previous domains automatically, the semantic relationship of words in a pk-set may not be correct for the current domain. It is a challenge to determine which pk-set is not appropriate. One way to deal with both problems is to assess how the words in a pk-set correlated with each other in the current domain. If they are more correlated, they are more likely to be correct for a topic in the domain and thus should be promoted more. If they are less correlated, they are more likely to be wrong and should be promoted less (or even not promoted).

To measure the correlation of two words in a pk-set in the current domain, we use Pointwise Mutual Information (PMI), which is a popular measure of words association in text. It has also been used to evaluate topic models (Newman et al., 2010a). PMI is the logarithmic ratio of the actual joint probability of two events to the expected joint probability if the two events were independent (Church and Hanks, 1990). In our case, it measures the extent to which two words tend to co-occur, which corresponds to the higher-order co-occurrence on which topic models are based (Heinrich, 2009). The PMI of two words is defined as follows:

$$PMI(w_1, w_2) = log \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \tag{4.3}$$

where $P(w)$ denotes the probability of seeing word w in a random document, and $P(w_1, w_2)$ denotes the probability of seeing both words co-occurring in a random document. These probabilities are empirically estimated using the current domain collection D^t:

$$P(w) = \frac{\#D^t(w)}{\#D^t} \tag{4.4}$$

$$P(w_1, w_2) = \frac{\#D^t(w_1, w_2)}{\#D^t} \tag{4.5}$$

where $\#D^t(w)$ is the number of documents in D^t that contain the word w and $\#D^t(w_1, w_2)$ is the number of documents that contain both words w_1 and w_2. $\#D^t$ is the total number of documents in D^t. A positive PMI value implies a true semantic correlation of words, while a non-positive PMI value indicates little or no semantic correlation. Thus, we only consider

pk-sets with positive PMI values. We also add a parameter factor μ to control how much the GPU model should trust the word relationships indicated by PMI (see the setting of μ in Section 4.3.1). Finally, the amount of promotion for word w' when seen w is defined as follows:

$$A'_{t,w,w'} = \begin{cases} 1 & w = w' \\ \mu \times PMI(w, w') & (w, w') \text{ is a pk-set of } t \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

4.2.2.2 Conditional Distribution of Gibbs Sampler

Similar with (Chen et al., 2013b), we take the approach of (Mimno et al., 2011) which approximates the true Gibbs sampling distribution by treating each word as if it were the last. The approximate Gibbs sampler has the following conditional distribution:

$$\begin{aligned} &P(z_i = t|\boldsymbol{z}^{-i}, \boldsymbol{w}, \alpha, \beta, A') \propto \\ &\frac{n_{d,t}^{-i} + \alpha}{\sum_{t'=1}^{T}(n_{d,t'}^{-i} + \alpha)} \times \frac{\sum_{w'=1}^{V} A'_{t,w',w_i} \times n_{t,w'}^{-i} + \beta}{\sum_{v=1}^{V}(\sum_{w'=1}^{V} A'_{t,w',v} \times n_{t,w'}^{-i} + \beta)} \end{aligned} \tag{4.7}$$

where n^{-i} is the count excluding the current assignment of z_i, i.e., $\boldsymbol{z}^{-i}$, $\boldsymbol{w}$ refers to all the words in all documents in the document collection D^t and w_i is the current word to be sampled with a topic denoted by z_i. $n_{d,t}$ denotes the number of times that topic t was assigned to words in document d, where d is the document index of word w_i. $n_{t,v}$ refers to the number of times that word v appears under topic t. α and β are predefined Dirichlet hyperparameters.

T is the number of topics, and V is the vocabulary size. A' is the promotion matrix defined in Equation 4.6.

4.3 Evaluation of LTM

This section evaluates the proposed LTM model and compares it with four state-of-the-art baselines:

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

DF-LDA (Andrzejewski et al., 2009): A knowledge-based topic model that can use the user-provided knowledge.

GK-LDA (Chen et al., 2013b): A knowledge-based topic model that uses the ratio of word probabilities under each topic to reduce the effect of wrong knowledge (Chapter 3).

AKL (Chen et al., 2014): A knowledge-based topic model that applies clustering to learn the knowledge and utilizes the knowledge in the form of knowledge clusters.

Note that although both DF-LDA and GK-LDA can take prior knowledge from the user, they cannot mine any prior knowledge, which make them not directly comparable with LTM. Thus, we have to feed them the knowledge produced using our proposed knowledge mining algorithm (Algorithm 4). This allows us to assess the knowledge handling capability of each model. AKL uses its own way to generate and incorporate knowledge.

4.3.1 Experimental Settings

Dataset. We have created a large dataset containing 50 review collections from 50 product domains crawled from Amazon.com (see Table II). Each domain has 1,000 (1K) reviews. We followed (Chen et al., 2013b) to pre-process the dataset. To test the behaviors of LTM for large

Alarm Clock	Computer	Kindle	Network Adapter	Telephone
Amplifier	DVD Player	Lamp	Printer	TV
Battery	Fan	Laptop	Projector	Vacuum
Blu-Ray Player	GPS	Media Player	Radar Detector	Video Player
Cable Modem	Graphics Card	Memory Card	Remote Control	Video Recorder
Camcorder	Hard Drive	Microphone	Rice Cooker	Voice Recorder
Camera	Headphone	Microwave	Scanner	Watch
Car Stereo	Home Theater System	Monitor	Speaker	Webcam
CD Player	Iron	Mouse	Subwoofer	Wireless Router
Cell Phone	Keyboard	MP3Player	Tablet	Xbox

TABLE II

Domain names of 50 different products from Amazon.

datasets (see Sections 4.3.4 and 4.3.5), we created four large review collections with 10,000 (10K) reviews in each. Note that most product domains in our collections do not have such a large number of reviews.

Parameter Setting. For all models, posterior estimates of latent variables were taken with a sampling lag of 20 iterations in the post burn-in phase (first 200 iterations for burn-in) with 2,000 iterations in total. The parameters of all topic models are set as $\alpha = 1$, $\beta = 0.1$, $T = 15$. The other parameters for baselines were set as suggested in their original papers. For parameters of LTM, the top 15 words of each topic were used to represent the topic in the topic matching process and also frequent itemset mining (Algorithm 4). This is intuitive as the top words in each topic are more likely to be semantically coherent while words at lower positions are much less related. The minimum support threshold is empirically set to $min(5, 0.4 \times \#Trans)$ where $\#Trans$ is the size of each M_{j*}^{t} (Section 4.2.1). This is also intuitive as appearances in a reasonable number of domains show likely word semantic correlations. The parameter π in

Algorithm 4 is empirically set to 7.0. The parameter μ in Equation 4.6 is set to 0.3, which determines the extent of promotion of words in a pk-set using the GPU model. Intuitively, a too small value of μ will lead to an inferior performance as it basically ignores the knowledge, while a too large value can damage the model too due to the errors in the knowledge.

Test Settings: We use two test settings to evaluate LTM, which represent two ways of using LTM in Section 4.1:

1. Mine prior knowledge pk-sets from topics of all domains including the test domain.
2. Mine prior knowledge pk-sets from topics of all domains excluding the test domain.

Setting 1 has a slight advantage as in mining knowledge for a test domain collection, its own initial topics are used, which can help find more targeted knowledge. We report the results for Setting 1 in Sections 4.3.2 and 4.3.3, and the results for Setting 2 in Section 4.3.4.

4.3.2 Topic Coherence of Test Setting 1

Following Section 3.4.2, we use Topic Coherence as the first measurement (Mimno et al., 2011). A higher Topic Coherence value indicates a higher quality of topics.

Our proposed algorithm (Algorithm 2) is designed for iterative improvements, i.e., a higher quality of topics can generate better knowledge, which in turn helps discover more coherent topics. This framework is also suitable for DF-LDA, GK-LDA, and AKL, i.e., the topics learned from a model at iteration r is used to generate knowledge for that model at iteration $r + 1$. Iteration 0 is equivalent to LDA (without any knowledge). We call each of these iterations a learning iteration. Since DF-LDA and GK-LDA cannot mine any prior knowledge, they use our proposed knowledge mining method. Our knowledge in the form of pairs (sets of two words)

has the same meaning as the knowledge used in DF-LDA (must-link) and GK-LDA (LR-set). In this work, we do not use cannot-links.

Figure 7 shows the average Topic Coherence value of each model at each learning iteration. Each value is the average Topic Coherence score over all 50 domains. Note that since LDA cannot use any prior knowledge, its results remain the same. From Figure 7, we can see that LTM performs the best and has the highest Topic Coherence values in general. These show that LTM finds higher quality topics than the baselines. Both AKL and GK-LDA perform better than LDA but worse than LTM, showing their ability of dealing with wrong knowledge to some extent. DF-LDA does not perform well. Without an automated way to deal with each piece of (correct or incorrect) knowledge specifically for each individual domain, its performance is actually worse than LDA.

In summary, we can say that the proposed LTM model can generate better quality topics than all baseline models. Even though DF-LDA and GK-LDA use our method for knowledge mining, without an effective wrong knowledge handling method, they are not sufficient. The improvements of LTM are all significant ($p < 0.01$ over AKL and $p < 0.0001$ over the other baselines) based on paired t-test.

4.3.3 Human Evaluation

Here we want to evaluate the topics based on human judgment. The results are still from test Setting 1. Two human judges who are familiar with Amazon products and reviews were asked to label the generated topics. Since we have a large number of domains, we selected 10 domains for labeling. The selection was based on the knowledge of the products of the two

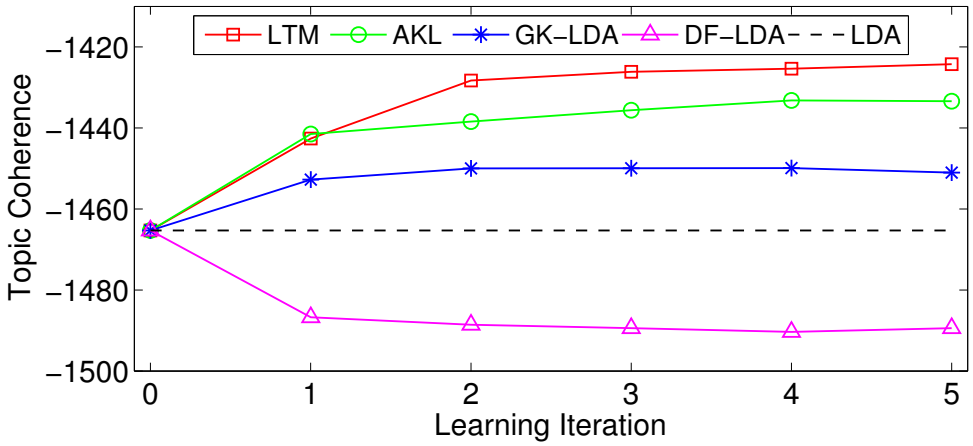

Figure 7. Average Topic Coherence values of each model at different learning iterations for Setting 1 (Iteration 0 = LDA).

human judges. Without enough knowledge, the labeling will not be reliable. We labeled the topics generated by LTM, LDA and DF-LDA at learning iteration 1. Similar as Section 3.4.3, we conducted topic labeling and word labeling. The Cohen's Kappa agreement scores for topic labeling and word labeling are 0.862 and 0.857 respectively.

Evaluation measures. Since topics are rankings of words based on their probabilities, without knowing the exact number of correct topical words, a natural way to evaluate these rankings is to use $Precision@n$ (or $p@n$) which was also used by other researchers, e.g., (Zhao et al., 2010), where n is a rank position. Apart from $p@n$, we also report the number of coherent topics found by each model.

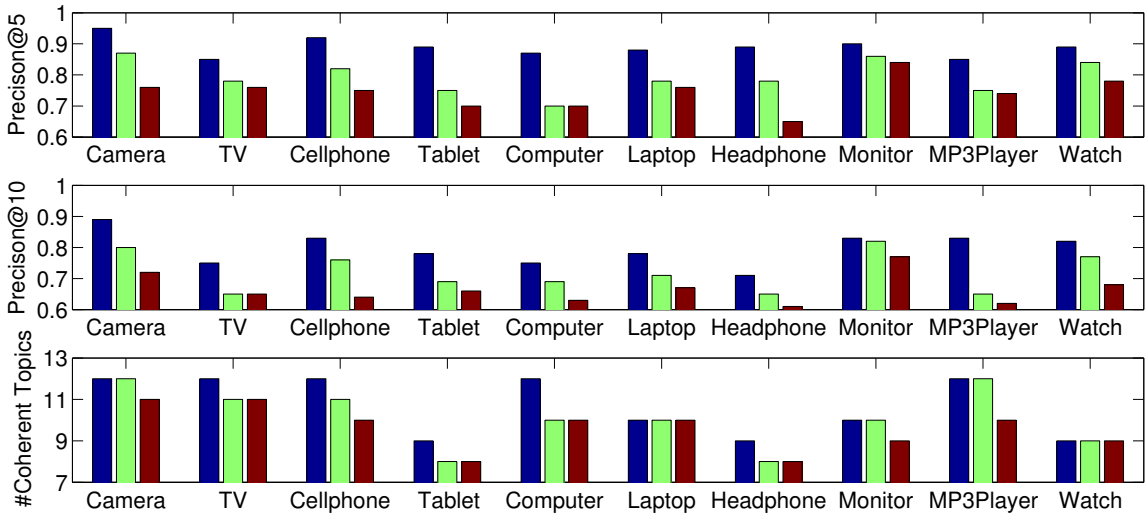

Figure 8. Top & Middle: Topical words *Precision*@5 & *Precision*@10 of coherent topics of each model respectively; Bottom: number of coherent (#Coherent) topics discovered by each model. The bars from left to right in each group are for LTM, LDA, and DF-LDA. On average, for *Precision*@5 and *Precision*@10, LTM improves LDA by 10% and 8%, and DF-LDA by 15% and 14% respectively. On average, LTM also discovers 0.6 more coherent topics than LDA and 1.1 more coherent topics than DF-LDA over the 10 domains.

Figure 8 gives the average topical words *Precision*@5 (top chart) and *Precision*@10 (middle chart) of only good topics (those bad topics are not considered) for each model in each domain. It is clear that LTM achieves the highest p@5 and p@10 values in all 10 domains. LDA is slightly better than DF-LDA in general, but clearly inferior to LTM. This is consistent with the Topic Coherence results in Section 4.3.2. The improvements of LTM vary in domains. For some domains, e.g., Camera, Tablet and Headphone, LTM achieves marked improvements. We found that these domains tend to have a lot of topic overlapping with many other domains. On the

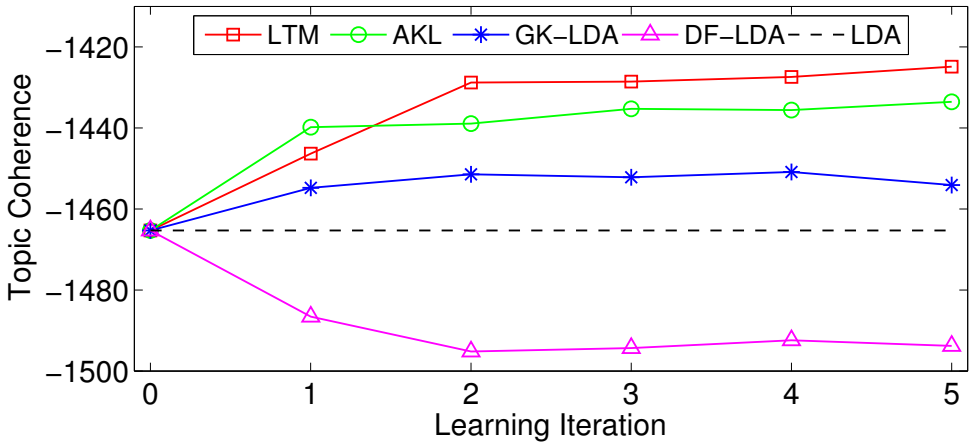

Figure 9. Average Topic Coherence values of each model at different learning iterations in Setting 2. The results are slightly worse than those of Setting 1 (Figure 7).

other hand, the improvements in the Monitor domain are less because of less topic overlapping with other domains. Significance testing using paired t-test shows that the improvements of LTM over the baselines on p@5 and p@10 are both significant ($p < 0.0001$). The bottom chart of Figure 8 shows that LTM also discovers more coherent topics than LDA and DF-LDA.

We can then conclude that LTM is superior to the baselines based on both Topic Coherence and human judgment.

4.3.4 Topic Coherence of Test Setting 2

We now evaluate LTM in Test Setting 2. That is, in mining pk-sets, we do not use the topics from the current domain but only p-topics from the other domains. We set the minimum support threshold for knowledge mining to be one less than that for Setting 1 as the current

topics are not used for knowledge mining. Here we also experiment the iterative process. We use each of the 50 domains as the current domain and the rest 49 domains as the prior domains. Figure 9 shows the average Topic Coherence values for this set of experiments. We can see that LTM again achieves higher Topic Coherence values in general, which is consistent with the results in previous sections. The results of LTM (and other knowledge-based models) are slightly worse than those of Setting 1 (Figure 7). This is expected as it does not use its own topics in knowledge mining, which can help mine more suitable knowledge for the domain.

Applying knowledge to 10K reviews. Figure 7 and Figure 9 showed that LTM improves topics for 1,000 (1K) reviews. An interesting question is whether LTM can also improve on 10K reviews given that LDA should perform better with 10,000 (10K) reviews as more data give more reliable statistics. We then apply the knowledge learned from test setting 2 at each learning iteration on each of four domains with 10K reviews. Figure 10 gives the average Topic Coherence values over these four domains. We can see that with larger datasets, LTM still gets significant improvements over LDA ($p < 0.0001$ based on paired t-test).

4.3.5 Improving topic modeling for Big Data

This sub-section shows that our approach can also be exploited to make topic modeling on a single big data more effective, slight improvements in topic quality and major improvements in efficiency.

Following our approach of learning from multiple domains, we randomly divide a big dataset into a number of small datasets and pretend that they are from multiple domains. With multiple small datasets, we can run our experiments just like that in Section 4.3.2. Here we use each of

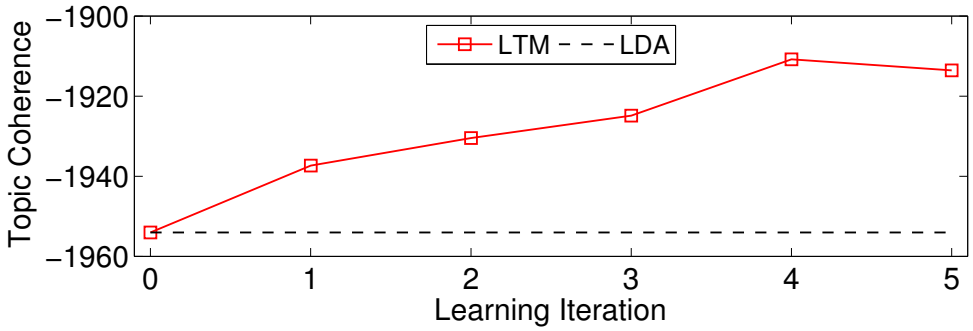

Figure 10. Average Topic Coherence values at different learning iterations over four 10K domains. The knowledge is mined from 49 domains of 1K reviews.

the four large data sets (10K reviews). Although our four large datasets are not particularly large, as it is shown in (Arora et al., 2013) that LDA using Gibbs sampling is linear in the number of documents, our results here are sufficient to show the trend.

For these experiments, we divide each of our four 10K review collections into 10 folders where each folder has 1K reviews. Then, we run the LTM model treating 10 folders as 10 domains, and evaluate both topic quality and efficiency based on Test Setting 1. Here, we also include AKL in the comparison as it gives the best Topic Coherence among baselines. Note that both PMI in LTM and co-document frequency ratio used in AKL are computed using 10K reviews. Figure 11 shows the Topic Coherence value of each model. Topic Coherence is calculated using 10K reviews. We can see that LTM achieves slightly higher Topic Coherence than LDA-10K (LDA on 10K reviews) and much higher Topic Coherence than LDA-1K (LDA on 1K reviews). AKL, however, gets the lowest Topic Coherence. We investigated its results

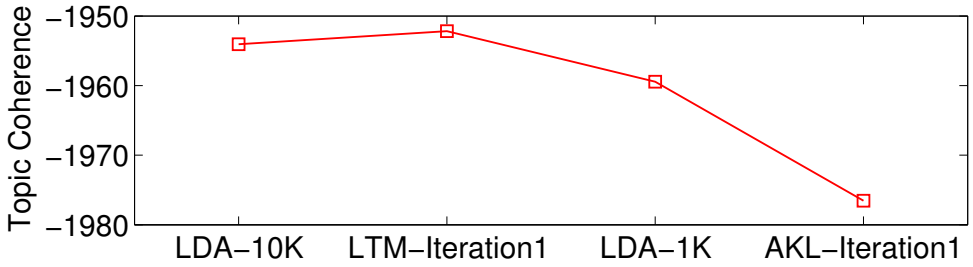

Figure 11. Average Topic Coherence values of each model when dividing big data into small data.

and found that for the noisy topics AKL tends to group them into the clusters of good topics, which lowers the quality of the mined knowledge. For AKL, we also tried different numbers of clusters with no improvements. The knowledge mining method in LTM is shown to be more effective. Since the 10 folders contain similar information, one learning iteration is sufficient (more learning iterations gave quite similar results). We also employ human labeling as in Section 4.3.3. For LTM and LDA-1K, we labeled the folder with the highest Topic Coherence value. The results are given in Figure 12 which also shows a slightly superior performance of LTM. The improvement of the labeled folder of LTM is 17 points compared with LDA-10K in terms of Topic Coherence. The topic quality improvements are not large due to the fact that the 10 small datasets are from the same domain and are thus less effective for knowledge learning.

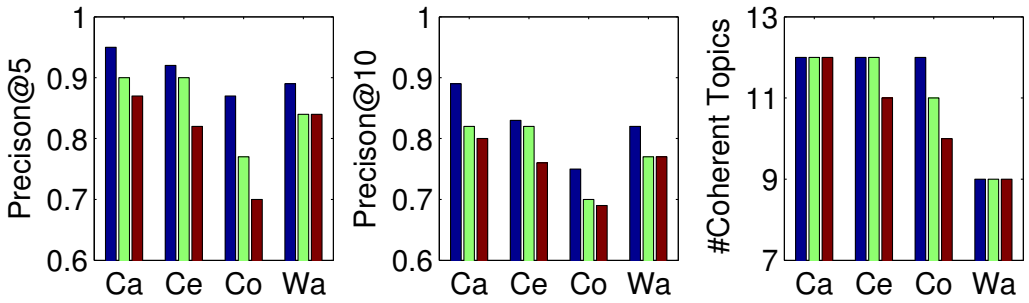

Figure 12. Human labeling of LTM, LDA-10K and LDA-1K (bars from left to right). The domains from left to rights are Camera, Cellphone, Computer, and Watch.

LTM's running time is about 31% of LDA-10K because we can run the 10 folders in parallel. Note that LTM in Figure 11 updates knowledge in every 50 iterations. LTM can be easily applied in MapReduce to further solve the memory issue with the big data.

In summary, we can conclude that with our LTM model, it is possible to run a big data set by dividing it into smaller datasets and achieve slightly better topic quality while greatly reduce the execution time.

4.4 Summary

This chapter proposed the Lifelong Topic Modeling (LTM) which combines the topic modeling with Lifelong Machine Learning (LML). LTM automatically mines the knowledge from the Topic Base generated from past domains. The knowledge is then automatically applied to a new fault-tolerant knowledge-based model. It is shown that LTM can discover topics of higher quality than traditional topic models. Also, it can be further exploited to deal with topic

modeling in big data. To summarize, we list the LML components (See Chapter 1) in the LTM model:

1. *Past Information Store* (**PIS**): It stores the Topic Base which consists of the topics discovered from past domains.
2. *Knowledge Base* (**KB**): It uses prior knowledge sets (or pk-sets).
3. *Knowledge Miner* (**KM**): It mines pk-sets using frequent itemset mining from the matching topics in the Topic Base. The matching is conducted using KL-Divergence of word distributions under the topics.
4. *Knowledge-Based Learner* (**KBL**): It proposes a new fault-tolerant knowledge-based model which combines the generalized Pólya urn (GPU) model while filtering the knowledge using Pointwise Mutual Information (PMI).

CHAPTER 5

TOPIC MODELING WITH AUTOMATICALLY GENERATED MUST-LINKS AND CANNOT-LINKS

(This chapter includes and expands on my paper previously published in *Zhiyuan Chen and Bing Liu. Mining Topics in Documents: Standing on the Shoulders of Big Data. In KDD 2014, pages 1116-1125.*)

This chapter introduces the AMC (topic modeling with Automatically generated Must-links and Cannot-links) model (Chen and Liu, 2014a) which further improves the LTM model (Chen and Liu, 2014b). AMC is superior to LTM model by extracting and incorporating the cannot-type of knowledge automatically; utilizing the multiple minimum supports frequent itemset mining algorithm (Liu et al., 1999); and constructing a graph of the must-type of knowledge to distinguish multiple senses. We will detail these in this chapter.

In the existing knowledge-based topic models, there are two existing ones, DF-LDA (Andrzejewski et al., 2009) and MC-LDA (Chen et al., 2013c), that can use both must-links and cannot-links to help generate better topics. We follow their work and use must-links and cannot-links as the knowledge. A *must-link* contains two words that should be put together in the same topic. A *cannot-link*, on the other hand, indicates that two words should not belong to the same topic. However, both DF-LDA and MC-LDA assume that the user-provided must-links and cannot-links are correct and there is no conflict among them. However, these assumptions are

violated in our case when Lifelong Machine Learning is considered because of the following issues:

1. The automatically generated must-links and cannot-links can have errors. Blindly trusting them as in DF-LDA and MC-LDA generates poor results (see Section 5.4).
2. A term may have multiple senses or meanings. This can cause the *transitivity problem*. That is, if A and B form a must-link, and B and C form a must-link, a topic model, such as DF-LDA, will put all three terms in one topic, which is clearly not always correct. For example, the term *light* can have two distinct meanings and the system may find two must-links, {light, weight} and {light, bright}. It is clearly unreasonable to put these three terms together under the same topic. MC-LDA has difficulty with this problem too because it only chooses one must-link for each term in each document and ignores the rest, which is undesirable because it can miss a lot of good must-link knowledge.

Here, we introduce our new lifelong learning approach as below, which considers both must-links and cannot-links:

Phrase 1 (*Initialization*): Given n prior document collections $\boldsymbol{D} = \{D_1, \ldots, D_n\}$, a topic model (e.g., LDA) is run on each collection $D_i \in \boldsymbol{D}$ to produce a set of prior topics (or *p-topics* for short) S_i. Let $S = \cup_i S_i$, which we call the *Topic Base*. It then mines must-links M from S using a multiple minimum supports frequent itemset mining algorithm (Liu et al., 1999).

Phase 2 (*Lifelong learning*): Given a new document collection D^t, a knowledge-based topic model (KBTM) with the must-links M is run to generate a set of topics A^t. Based on A^t, the algorithm finds a set of cannot-links C using Topic Base. The KBTM then continues, which

is now guided by both must-links M and cannot-links C, to produce the final topic set A^t. We will explain why we mine cannot-links based on A^t in Section 5.2.2. To enable LML, A^t is incorporated into S, which is used to generate a new set of must-links M.

In this chapter, we propose a new topic model, called AMC (topic modeling with Automatically generated Must-links and Cannot-links) (Chen and Liu, 2014a), whose inference can exploit the automatically mined knowledge and deal with the issues of wrong knowledge and transitivity to produce superior topics. Our experiments, using review collections from 100 domains, show that the proposed AMC model outperforms state-of-the-art baseline models significantly.

5.1 Overall Algorithm

This section introduces the proposed overall algorithm, which follows the LML idea. The algorithm consists of two phases:

Phase 1 - Initialization: Given a set of prior document collections $\boldsymbol{D} = \{D_1, \ldots, D_n\}$ from n domains, this step first runs the standard LDA on each domain collection $D_i \in \boldsymbol{D}$ to generate a set of topics S_i. The resulting topics from all n domains are unionized to produce the set of all topics S, i.e., $S = \cup_i S_i$. We call S the Topic Base which consists of *prior topics* (or *p-topics*). A set of must-links are then mined from S, which will be detailed in Section 5.2.1. Note that this initialization phase is only applied at the beginning. It will not be used for modeling of each new document collection.

Phase 2 - LML with AMC: Given a new/test document collection D^t, this phase employs the proposed AMC model to generate topics from D^t. To distinguish these topics from p-topics,

Algorithm 5 AMC(D^t, S, M)

1: $A^t \leftarrow$ GibbsSampling(D^t, N, M, $\emptyset$); // $\emptyset$: no cannot-links.
2: **for** $r = 1$ **to** R **do**
3: $\quad C \leftarrow C \cup$ MineCannotLinks(S, A^t);
4: $\quad A^t \leftarrow$ GibbsSampling(D^t, N, M, C);
5: **end for**
6: $S \leftarrow$ Incorporate(A^t, S);
7: $M \leftarrow$ MiningMustLinks(S);

we call them the *current topics* (or *c-topics* for short). AMC is given in Algorithm 5. Line 1 runs the proposed Gibbs sampler (introduced in Section 5.3.3) using only the must-links M generated from Topic Base S so far to produce a set of topics A^t, where N is the number of Gibbs sampling iterations. Line 3 mines cannot-links based on the current topics A^t and Topic Base S (see Section 5.2.2). Then line 4 uses both must-links and cannot-links to improve the resulting topics. Note that this process can run iteratively. We call these iterations the *learning iterations*, which are different from the Gibbs iterations. In each learning iteration, we hope to obtain better topic results. We will experiment with the number of learning iterations in Section 5.4. Currently, the function *Incorporate*(A^t, S) (line 6 in Algorithm 5) is very simple. If the domain of A^t exists in S, replace those topics of the domain in S with A^t; otherwise, A^t is added to S. With the updated S, a new set of must-links is mined (line 7), which will be used in the next new modeling task by calling AMC.

5.2 Mining Knowledge

In this section, we present the algorithms for mining must-links and cannot-links, which form our prior knowledge to be used to guide future modeling.

5.2.1 Mining Must-Link Knowledge

A must-link means that two terms w_1 and w_2 in it should belong to the same topic. That is, there should be some semantic correlation between them. We thus expect w_1 and w_2 to appear together in a number of p-topics in several domains due to the correlation. For example, for a must-link {*price*, *cost*}, we should expect to see *price* and *cost* as topical terms in the same topic across many domains. Note that they may not appear together in every topic about *price* due to the special context of the domain or past topic modeling errors. Thus, it is natural to use a frequency-based approach to mine frequent sets of terms (words) as reliable must-links.

Before going further, let us first discuss the representation of a topic to be used in mining. Recall that each topic generated from a topic model, such as LDA, is a distribution over terms (or words), i.e., terms with their associated probabilities. Terms are commonly ranked based on their probabilities in a descending order. In practice, top terms under a topic are expected to represent some similar semantic meaning. The lower ranked terms usually have very low probabilities due to the smoothing effect of the Dirichlet hyperparameters rather than true correlations within the topic, leading to their unreliability. Thus, in this work, only top 15 terms are employed to represent a topic. For mining the must-link and cannot-link knowledge, we use this topic representation.

Given Topic Base S, similar as LTM in Chapter 4, we find sets of terms that appear together in multiple topics using the data mining technique *frequent itemset mining* (FIM). Each itemset is simply a set of terms. The resulting frequent itemsets serve as must-links. However, this technique is insufficient due to the problem with the single minimum support threshold used in classic FIM algorithms.

A single minimum support is not appropriate because generic topics, such as *price* with topic terms like *price* and *cost*, are shared by many (even all) product review domains, but specific topics such as *screen*, occur only in product domains having such features. This means that different topics may have very different frequencies in the data. Thus, using a single minimum support threshold is unable to extract both generic and specific topics because if we set this threshold too low, the generic topics will result in numerous spurious frequent itemsets (which results in wrong must-links) and if we set it too high we will not find any must-link from less frequent topics. This is called the *rare item problem* in data mining and has been well documented in (Liu, 2007).

Due to this problem, we cannot use a traditional frequent item mining algorithm. We actually experimented with one such algorithm, but it produced very poor must-links. We thus use the multiple minimum supports frequent itemset mining (MS-FIM) algorithm in (Liu et al., 1999). MS-FIM is stated as follows: Given a set of transactions T, where each transaction $t_i \in T$ is a set of items from a global item set I, i.e., $t_i \subseteq I$. In our context, t_i is the topic vector comprising the top terms of a topic (no probability attached). An item is a term (or word). T is thus the collection of all p-topics in S and I is the set of all terms in S. In MS-FIM,

each item/term is given a minimum itemset support (MIS). The minimum support that an itemset (a set of items) must satisfy is not fixed. It depends on the MIS values of all the items in the itemset. MS-FIM also has another constraint, called the support difference constraint (SDC), expressing the requirement that the supports of the items in an itemset must not be too different. MIS and SDC together can solve the above rare item problem. For details about MS-FIM, please refer to (Liu et al., 1999).

The goal of MS-FIM is to find all itemsets that satisfy the user-specified MIS thresholds. Such itemsets are called *frequent itemsets*. In our context, a frequent itemset is a set of terms which have appeared multiple times in the p-topics of Topic Base. The frequent itemsets of length two are used as our learned must-link knowledge, e.g.,

{battery, life}, {battery, power}, {battery, charge},

{price, expensive}, {price, pricy}, {cheap, expensive}

Note that we use must-links with only two terms in each as they are sufficient to cover the semantic relationship of terms belonging to the same topic. As mentioned in Section 4.2.1, larger sets tend to contain more errors, i.e., the terms in a set may not belong to the same topic. Such errors are also harder to deal with than those in pairs. The same rationale applies to cannot-links.

5.2.2 Mining Cannot-Link Knowledge

Following the same intuition as must-link knowledge mining, we also utilize a frequency based approach to mine the cannot-link knowledge. However, there is a major difference. It is prohibitive to find all cannot-links based on the prior document collections $\boldsymbol{D}$. For a term

w, there are usually only a few terms w_m that share must-links with w while there are a huge number of terms w_c that can form cannot-links with w. For example, only the terms related with *price* or *money* share must-links with *expensive*, but the rest of the terms in the vocabulary of $\boldsymbol{D}$ can form potential cannot-links. Thus, in general, if there are V terms in the vocabulary, there are $O(V^2)$ potential cannot-links. However, for a new or test domain D^t, most of these cannot-links are not useful because the vocabulary size of D^t is much smaller than V. Thus, we focus only on those terms that are relevant to D^t.

Formally, given Topic Base S from all domain collections $\boldsymbol{D}$ and the current c-topics A^t from the test domain D^t, we extract cannot-links from each pair of top terms w_1 and w_2 in each c-topic $A_j^t \in A^t$. Based on this formulation, to mine cannot-links, we enumerate every pair of top terms w_1 and w_2 and check whether they form a cannot-link or not. Thus, our cannot-link mining is targeted to each c-topic with the aim to improve the c-topic using the discovered cannot-links.

To determine whether two terms form a cannot-link, if the terms seldom appear together in p-topics, they are likely to have distinct semantic meanings. Let the number of past domains that w_1 and w_2 appear in different p-topics be N_{diff} and the number of past domains that w_1 and w_2 share the same topic be N_{share}. N_{diff} should be much larger than N_{share}. We need to use two conditions or thresholds to control the formation of a cannot-link:

1. The ratio $N_{diff}/(N_{share} + N_{diff})$ (called the *support ratio*) is equal to or larger than a threshold π_c. This condition is intuitive because p-topics may contain noise due to errors of topic models.

2. N_{diff} is greater than a support threshold π_{diff}. This condition is needed because the above ratio can be 1, but N_{diff} can be very small, which may not give reliable cannot-links.

Using the above approach, we list some extracted cannot-link examples below:

{battery, money}, {life, movie}, {battery, line}

{price, digital}, {money, slow}, {expensive, simple}

5.3 AMC Model

We now present the proposed AMC model. As noted earlier, due to errors in the results of topic models, some of the automatically mined must-links and cannot-links may be wrong. AMC is capable of handling such incorrect knowledge. The idea is that the semantic relationships reflected by correct must-links and cannot-links should also be reasonably induced by the statistical information underlying the domain collection. If a piece of knowledge (a must-link or a cannot-link) is inconsistent with a domain collection, this piece of knowledge is likely to be either incorrect in general or incorrect in this particular test domain. In either case, the model should not trust or utilize such knowledge.

AMC still uses the graphical model of LDA and its generative process. Thus, we do not give the graphical model. However, the inference mechanism of AMC is entirely different from that of LDA. The inference mechanism cannot be reflected in the graphical model using the plate notation.

Below we first discuss how to handle issues with must-links and cannot-links and then put everything together to present the proposed Gibbs sampler extending the *Pólya urn* model, which we call the *multi-generalized Pólya urn* (M-GPU) model.

5.3.1 Dealing with Issues of Must-Links

There are two major challenges in incorporating the must-link knowledge:

1. A term can have multiple meanings or senses. For example, *light* may mean "something that makes things visible" or "of little weight." Different senses may lead to distinct must-links. For example, with the first sense of *light*, the must-links can be {light, bright}, {light, luminance}. In contrast, {light, weight}, {light, heavy} indicate the second sense of light. The existing knowledge-based topic model DF-LDA (Andrzejewski et al., 2009) cannot distinguish multiple senses because its definition of must-link is transitive. That is, if terms w_1 and w_2 form a must-link, and terms w_2 and w_3 form a must-link, it implies a must-link between w_1 and w_3, i.e., w_1, w_2, and w_3 should be in the same topic. We call it the *transitivity* problem. DF-LDA would incorrectly assume that *light*, *bright*, and *weight* are in the same topic. MC-LDA (Chen et al., 2013c) assumes each must-link represents a distinct sense, and thus assigns each term only one relevant must-link and ignores the rest. This misses a lot of good must-links. We propose a method in Section 5.3.1.1 to distinguish multiple senses embedded in must-links and deal with the transitivity problem.
2. Not every must-link is suitable for a domain. First, a must-link may not be correct in general due to errors in topic modeling and knowledge mining, e.g., {battery, beautiful} is not a correct must-link generally. Second, a must-link may be correct in some domains but wrong in others. For example, {card, bill} is a correct must-link in the domain of restaurant (the card here refers to credit cards), but unsuitable in the domain of camera. We will introduce a method to deal with such inappropriate knowledge in Section 5.3.1.2.

To deal with the first issue, we construct a must-link graph to distinguish multiple senses in must-links to deal with the transitivity problem. To tackle the second problem, we again utilize Pointwise Mutual Information (PMI) to estimate the word correlations of must-link terms in the domain collection. These techniques will be introduced in the next two sub-sections and incorporated in the proposed Gibbs sampler in Section 5.3.3.

5.3.1.1 Recognizing Multiple Senses

In order to handle the transitivity problem, we need to distinguish multiple senses of terms in must-links. As our must-links are automatically mined from a set of p-topics, the p-topics may also give us some guidance on whether the mined must-links share the same word sense or not. Given two must-links m_1 and m_2, if they share the same word sense, the p-topics that cover m_1 should have some overlapping with the p-topics that cover m_2. For example, must-links {light, bright} and {light, luminance} should be mostly coming from the same set of p-topics related to the semantic meaning "something that makes things visible" of *light*. On the other hand, little topic overlapping indicates likely different word senses. For example, must-links {light, bright} and {light, weight} may come from two different sets of p-topics as they usually refer to different topics.

Following this idea, we construct a must-link graph G where a must-link is a vertex. An edge is formed between two vertices if the two must-links m_1 and m_2 have a shared term. For each edge, we check how much their original p-topics overlap to decide whether the two must-

links share the same sense or not. Given two must-links m_1 and m_2, we denote the p-topics in S covering each of them as T_1 and T_2 respectively. m_1 and m_2 share the same sense if

$$\frac{\#T_1 \cap T_2}{Max(\#T_1, \#T_2)} > \pi_{overlap} \tag{5.1}$$

where $\pi_{overlap}$ is the *overlap threshold* for distinguishing senses. This threshold is necessary due to errors of topic models. The edges that do not satisfy the above inequality (Equation 5.1) are deleted.

The final must-link graph G gives us some guidance in selecting the right must-links sharing the same word sense in the Gibbs sampler in Section 5.3.3 for dealing with the transitivity problem.

5.3.1.2 Detecting Possible Wrong Knowledge

To measure the correctness of a must-link in a particular domain, we apply Pointwise Mutual Information (PMI), which is a popular measure of word associations in text. In our case, it measures the extent to which two terms tend to co-occur, which corresponds to "the higher-order co-occurrence" on which topic models are based (Heinrich, 2009). The definition of PMI was given in Equation 4.3 in Section 4.2.2.1. A positive PMI value implies a semantic correlation of terms, while a non-positive PMI value indicates little or no semantic correlation. Thus, we only consider the positive PMI values, which will be used in the proposed Gibbs sampler in Section 5.3.3.

5.3.2 Dealing with Issues of Cannot-Links

The main issue here is incorrect cannot-links. Similar to must-links, there are also two cases: a) A cannot-link contains terms that have semantic correlations. For example, {battery, charger} is not a correct cannot-link. b) A cannot-link does not fit for a particular domain. For example, {card, bill} is a correct cannot-link in the camera domain, but not appropriate for restaurants.

Wrong cannot-links can also cause conflicts with must-links. For example, the system may find two must-links {price, cost} and {price, pricy} and a cannot-link {pricy, cost}. Existing knowledge-based models, such as DF-LDA (Andrzejewski et al., 2009) and MC-LDA (Chen et al., 2013c), cannot solve these problems. A further challenge for these systems is that the number of automatically mined cannot-links is large (more than 400 cannot-links on average). Both DF-LDA and MC-LDA are incapable of using such a large amount of cannot-links. As we will see in Section 5.4, DF-LDA crashed and MC-LDA generated a large number of additional (wrong) topics with very poor results.

Wrong cannot-links are usually harder to detect and to verify than wrong must-links. Due to the power-law distribution of natural language words (Zipf, 1932), most words are rare and will not co-occur with most other words. The low co-occurrences of two words do not necessarily mean a negative correlation (cannot-link). Thus, we detect and balance cannot-links inside the sampling process. More specifically, we extend Pólya urn model to incorporate the cannot-link knowledge, and also to deal with the issues above.

5.3.3 Proposed Gibbs Sampler

This section introduces the Gibbs sampler for the proposed AMC model, which differs from LDA as AMC needs the additional mechanism to leverage the prior knowledge and to also deal with the problems with the prior knowledge during sampling. We propose the *multi-generalized Pólya urn* (M-GPU) model for the task. Below, we first introduce the Pólya urn model which serves as the basic framework to incorporate knowledge, and then enhance it to address the challenges mentioned in the above sub-sections.

5.3.3.1 Pólya Urn Model

Instead of involving only one urn at a time as in GK-LDA (Chen et al., 2013b) and LTM (Chen and Liu, 2014b), the proposed *multi-generalized Pólya urn* (M-GPU) model considers a set of urns in the sampling process simultaneously. M-GPU allows a ball to be transferred from one urn to another, enabling multi-urn interactions. Thus, during sampling, the populations of several urns will evolve even if only one ball is drawn from one urn. This capability makes the M-GPU model more powerful and suitable for solving our complex problems.

5.3.3.2 Proposed M-GPU Model

In M-GPU, when a ball is randomly drawn, certain numbers of additional balls of each color are returned to the urn, rather than just two balls of the same color as in SPU. This is inherited from GPU. As a result, the proportions of these colored balls are increased, making them more likely to be drawn in this urn in the future. We call this the *promotion* of these colored balls. Applying the idea to our case, when a term w is assigned to a topic k, each term w' that shares a must-link with w is also assigned to topic k by a certain amount, which is decided by the

matrix $\lambda_{w',w}$ (see Equation 5.2). w' is thus promoted by w. As a result, the probability of w' under topic k is also increased.

To deal with multiple senses problem in M-GPU, we exploit the fact that each term usually has only one correct sense or meaning under one topic. Since the semantic concept of a topic is usually represented by some top terms under it, we refer the word sense that is the most related to the concept as the correct sense. If a term w does not have multiple must-links, then we do not have the multiple sense problem caused by must-links. If w has multiple must-links, the rationale here is to sample a must-link (say m) that contains w to be used to represent the likely word sense from the must-link graph G (built in Section 5.3.1.1). The sampling distribution will be given in Section 5.3.3.3. Then, the must-links that share the same word sense with m, including m, are used to promote the related terms of w.

To deal with possible wrong must-links, we leverage the PMI measure (in Section 5.3.1.2) to estimate knowledge correctness in the M-GPU model. More specifically, we add a parameter factor μ to control how much the M-GPU model should trust the word relationship indicated by PMI. Formally, the amount of promotion for term w' when seen w is defined as follows:

$$\lambda_{w',w} = \begin{cases} 1 & w = w' \\ \mu \times PMI(w, w') & (w, w') \text{ is a must-link} \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

To deal with cannot-links, M-GPU defines two sets of urns which will be used in sampling in the AMC model. The first set is the set of topic urns $U^K_{d \in \{1 \ldots D^t\}}$, where each urn is for one

document and contains balls of K colors (topics) and each ball inside has a color $k \in \{1 \dots K\}$. This corresponds to the document-topic distribution in AMC. The second set of urns is the set of term urns $U^W_{k \in \{1 \dots K\}}$ corresponding to the topic-term distributions, with balls of colors (terms) $w \in \{1 \dots V\}$ in each term urn.

Based on the definition of cannot-link, two terms in a cannot-link cannot both have large probabilities under the same topic. As M-GPU allows multi-urn interactions, when sampling a ball representing term w from a term urn U^W_k, we want to transfer the balls representing the cannot-terms of w, say w_c (sharing cannot-links with w) to other urns (see Step 5 below), i.e., decreasing the probabilities of those cannot-terms under this topic while increasing their corresponding probabilities under some other topic. In order to correctly transfer a ball that represents term w_c, it should be transferred to an urn which has a higher proportion of w_c. That is, we randomly sample an urn that has a higher proportion of w_c to transfer w_c to (Step 5b below). However, there is a situation when there is no other urn that has a higher proportion of w_c. (Chen et al., 2013c) proposed to create a new urn to move w_c to under the assumption that the cannot-link knowledge is correct. As discussed in Section 5.3.2, the cannot-link knowledge may not be correct. For example, consider that the model puts battery and life in the same topic k where both *battery* and *life* have the highest probability (or proportion), a cannot-link {battery, life} wants to separate them after seeing them in the same topic. In such a case, we should not trust the cannot-link as it may split the correlated terms into different topics.

Based on all the above ideas, we now present the M-GPU sampling scheme as follows:

1. Sample a topic k from U_d^K and a term w from U_k^W sequentially, where d is the dth document in D^t.
2. Record k and w, put back two balls of color k into urn U_d^K, and two balls of color w into urn U_k^W.
3. Sample a must-link m that contains w from the prior knowledge base. Get a set of must-links $\{m'\}$ where m' is either m or a neighbor of m in the must-link graph G.
4. For each must-link $\{w, w'\}$ in $\{m'\}$, we put back $\lambda_{w',w}$ number of balls of color w' into urn U_k^W based on matrix $\lambda_{w',w}$ (in Equation 5.2).
5. For each term w_c that shares a cannot-link with w:
   (a) Draw a ball q_c of color w_c (to be transferred) from U_k^W and remove it from U_k^W. The document of ball q_c is denoted by d_c. If no ball of color w_c can be drawn (i.e., there is no ball of color w_c in U_k^W), skip steps b) and c).
   (b) Produce an urn set $\{U_{k'}^W\}$ such that each urn in it satisfies the following conditions:
   i) $k' \neq k$
   ii) The proportion of balls of color w_c in $U_{k'}^W$ is higher than that of balls of color w_c in U_k^W.
   (c) If $\{U_{k'}^W\}$ is not empty, randomly select one urn $U_{k'}^W$ from it. Put the ball q_c drawn from Step a) into $U_{k'}^W$. Also, remove a ball of color k from urn $U_{d_c}^K$ and put back a ball of k' into urn $U_{d_c}^K$. If $\{U_{k'}^W\}$ is empty, put the ball q_c back to U_k^W.

5.3.3.3 Sampling Distributions

Based on the above sampling scheme of M-GPU, this sub-section gives the final Gibbs sampler with the conditional distributions and algorithms for the AMC model. Inference of topics can be computationally expensive due to the non-exchangeability of words under the M-GPU models. We thus take the same approach as that for GPU in (Mimno et al., 2011) which approximates the true Gibbs sampling distribution by treating each word as if it were the last.

For each term w_i in each document d, there are two phases corresponding to the M-GPU sampling process (Section 5.3.3.2):

Phase 1 (Steps 1-4 in M-GPU): calculate the conditional probability of sampling a topic for term w_i. We enumerate each topic k and calculate its corresponding probability, which is decided by three sub-steps:

a) Sample a must-link m_i that contains w_i, which is likely to have the word sense consistent with topic k, which is based on the following conditional distribution:

$$P(m_i = m|k) \propto P(w_1|k) \times P(w_2|k) \tag{5.3}$$

where w_1 and w_2 are the terms in must-link m and one of them is the same as w_i. $P(w|k)$ is the probability of term w under topic k given the current status of the Markov chain in the Gibbs sampler, which is defined as:

$$P(w|k) \propto \frac{\sum_{w'=1}^{V} \lambda_{w',w} \times n_{k,w'} + \beta}{\sum_{v=1}^{V}(\sum_{w'=1}^{V} \lambda_{w',v} \times n_{k,w'} + \beta)} \tag{5.4}$$

where $\lambda_{w',w}$ is the promotion matrix in Equation 5.2. $n_{k,w}$ refers to the number of times that term w appears under topic k. β is the predefined Dirichlet hyper-parameter.

b) After getting the sampled must-link m_i, we create a set of must-links $\{m'\}$ where m' is either m_i or a neighbor of m_i in the must-link graph G. The must-links in this set $\{m'\}$ are likely to share the same word sense of term w_i according to the corresponding edges in the must-link graph G.

c) The conditional probability of assigning topic k to term w_i is defined as below:

$$\begin{aligned} & p(z_i = k|\boldsymbol{z}^{-i}, \boldsymbol{w}, \alpha, \beta, \lambda) \\ & \propto \frac{n_{d,k}^{-i} + \alpha}{\sum_{k'=1}^{K}(n_{d,k'}^{-i} + \alpha)} \\ & \times \frac{\sum_{\{w',w_i\}\in\{m'\}} \lambda_{w',w_i} \times n_{k,w'}^{-i} + \beta}{\sum_{v=1}^{V}(\sum_{\{w',v\}\in\{m'_v\}} \lambda_{w',v} \times n_{k,w'}^{-i} + \beta)} \end{aligned} \tag{5.5}$$

where n^{-i} is the count excluding the current assignment of z_i, i.e., $\boldsymbol{z}^{-i}$. $\boldsymbol{w}$ refers to all the terms in all documents in the document collection D^t and w_i is the current term to be sampled with a topic denoted by z_i. $n_{d,k}$ denotes the number of times that topic k is

assigned to terms in document d. $n_{k,w}$ refers to the number of times that term w appears under topic k. α and β are predefined Dirichlet hyper-parameters. K is the number of topics, and V is the vocabulary size. $\{m'_v\}$ is the set of must-links sampled for each term v following Phase 1 a) and b), which is recorded during the iterations. $\lambda_{w',w}$ is the promotion matrix in Equation 5.2.

Phase 2 (Step 5 in M-GPU): this sampling phase deals with cannot-links. There are two sub-steps:

a) For every cannot-term (say w_c) of w_i, we sample one instance (say q_c) of w_c from topic z_i, where z_i denotes the topic assigned to term w_i in Phase 1, based on the following conditional distribution:

$$P(q = q_c|\boldsymbol{z}, \boldsymbol{w}, \alpha) \propto \frac{n_{d_c,k} + \alpha}{\sum_{k'=1}^{K}(n_{d_c,k'} + \alpha)} \tag{5.6}$$

where d_c denotes the document of the instance q_c. If there is no instance of w_c in z_i, skip step b).

b) For each drawn instance q_c from Phase 2 a), resample a topic k (not equal to z_i) based on the conditional distribution below:

$$
\begin{aligned}
& P(z_{q_c} = k | \boldsymbol{z}^{-q_c}, \boldsymbol{w}, \alpha, \beta, \lambda, q = q_c) \\
& \propto \boldsymbol{I}_{[0, p(w_c|k)]}(P(w_c|z_c)) \\
& \times \frac{n_{d_c,k}^{-q_c} + \alpha}{\sum_{k'=1}^{K}(n_{d_c,k'}^{-q_c} + \alpha)} \\
& \times \frac{\sum_{\{w', w_c\} \in \{m'_c\}} \lambda_{w', w_c} \times n_{k,w'}^{-q_c} + \beta}{\sum_{v=1}^{V}(\sum_{\{w', v\} \in \{m'_v\}} \lambda_{w', v} \times n_{k,w'}^{-q_c} + \beta)}
\end{aligned}
\tag{5.7}
$$

where z_c (the same as z_i sampled from Equation 5.5) is the original topic assignment. $\{m'_c\}$ is the set of must-links sampled for term w_c. Superscript $-q_c$ denotes the counts excluding the original assignments. $\boldsymbol{I}()$ is an indicator function, which restricts the ball to be transferred only to an urn that contains a higher proportion of term w_c. If there is no topic k has a higher proportion of w_c than z_c, then keep the original topic assignment, i.e., assign z_c to w_c.

5.4 Evaluation

This section evaluates the proposed AMC model and compares it with five state-of-the-art baseline models:

- **LDA** (Blei et al., 2003): The classic unsupervised topic model.
- **DF-LDA** (Andrzejewski et al., 2009): A knowledge-based topic model that can use both must-links and cannot-links, but it assumes all the knowledge is correct.

- **MC-LDA** (Chen et al., 2013c): A knowledge-based topic model that also uses both the must-link and the cannot-link knowledge. It assumes that all knowledge is correct as well.
- **GK-LDA** (Chen et al., 2013b): A knowledge-based topic model that uses the ratio of word probabilities under each topic to reduce the effect of wrong knowledge. However, it can only use the must-link type of knowledge. See Chapter 3.
- **LTM** (Chen and Liu, 2014b): A lifelong learning topic model that learns only the must-link type of knowledge automatically. It outperformed (Chen et al., 2014). See Chapter 4.

Note that although DF-LDA, MC-LDA and GK-LDA can take prior knowledge from the user, they cannot mine any prior knowledge, which make them not directly comparable with the proposed AMC model. We have to feed them the knowledge produced using the proposed knowledge mining algorithm. This enables us to assess the knowledge handling capability of each model. LTM uses its own way to mine and incorporate must-links.

5.4.1 Experimental Settings

Datasets. We have created two large datasets for our experiments. The first dataset contains reviews from 50 types of electronic products or domains (given in the first row of Table III). The second dataset contains reviews from 50 mixed types of non-electronic products or domains (given in the second row of Table III). Each domain has 1000 reviews. Using the first dataset, we want to show the performance of AMC when there is a reasonably large topic overlapping. Using the second dataset, we want to show AMC's performance when there is not much topic overlapping. We followed (Chen et al., 2013b) to pre-process the dataset. The datasets are publicly available online.

Alarm Clock, Amplifier, Battery, Blu-Ray Player, Cable Modem, Camcorder, Camera, Car Stereo, CD Player, Cell Phone, Computer, DVD Player, Fan, GPS, Graphics Card, Hard Drive, Headphone, Home Theater System, Iron, Keyboard, Kindle, Lamp, Laptop, Media Player, Memory Card, Microphone, Microwave, Monitor, Mouse, MP3Player, Network Adapter, Printer, Projector, Radar Detector, Remote Control, Rice Cooker, Scanner, Speaker, Subwoofer, Tablet, Telephone, TV, Vacuum, Video Player, Video Recorder, Voice Recorder, Watch, Webcam, Wireless Router, Xbox
Android Appstore, Appliances, Arts Crafts Sewing, Automotive, Baby, Bag, Beauty, Bike, Books, Cable, Care, Clothing, Conditioner, Diaper, Dining, Dumbbell, Flashlight, Food, Gloves, Golf, Home Improvement, Industrial Scientific, Jewelry, Kindle Store, Kitchen, Knife, Luggage, Magazine Subscriptions, Mat, Mattress, Movies TV, Music, Musical Instruments, Office Products, Patio Lawn Garden, Pet Supplies, Pillow, Sandal, Scooter, Shoes, Software, Sports, Table Chair, Tent, Tire, Toys, Video Games, Vitamin Supplement, Wall Clock, Water Filter

TABLE III. List of 100 domain names: electronic products (1st row) and non-electronic products (2nd row).

Parameter Setting. All models were trained using 2000 iterations with an initial burn-in of 200 iterations. The parameters of all topic models are set to $\alpha = 1$, $\beta = 0.1$, $K = 15$ (#Topics). The other parameters for the baselines were set as suggested in their original papers. For parameters of AMC, we estimated its parameters using a development set from the domain, Calculator, which was not used in the evaluation. The minimum item support count (MIS) for each term is set to Max(4, 35% of its actual support count in the data) and the support difference is 8% (Liu, 2007). The support ratio threshold (π_c) and support threshold (π_{diff}) for cannot-link mining is 80% and 10 respectively. The overlap ratio threshold $\pi_{overlap}$ for forming a must-link graph edge is 17%. The parameter μ in Equation 5.2 is set to 0.5, which determines the extent of promotion of words in must-links using the M-GPU model.

5.4.2 Topic Coherence

This sub-section evaluates the topics generated by each model based on the Topic Coherence measure in (Mimno et al., 2011). A higher Topic Coherence indicates a higher quality of topics.

In this and the next two sub-sections, we experiment with the 50 Electronics domains, which have a large amount of topic overlapping. We treat each domain as a test set (D^t) while the knowledge is mined from the rest 49 domains. Since our main aim is to improve topic modeling with small datasets, each test set consists of 100 reviews randomly sampled from the 1000 reviews of the domain. We extract knowledge from topics generated from the full data (1000 reviews) of all other 49 domains. Since we have 50 domains, we have 50 small test sets. Figure 13 shows the average Topic Coherence value of each model over the 50 test sets. From Figure 13, we can observe the following:

1. AMC performs the best with the highest Topic Coherence value. In the Figure, "AMC" refers to the AMC model with both must-links and cannot-links and "AMC-M" refers to the AMC model with must-links only. We can see that AMC-M is already better than all baseline models, showing the effectiveness of must-links. AMC is much better than AMC-M which demonstrates that cannot-links are very helpful. These results show that AMC finds higher quality topics than the baselines.

   Note that in our experiments, we found DF-LDA and MC-LDA cannot deal with a large number of cannot-links. We have more than 400 automatically mined cannot-links on average for each test set. For DF-LDA, the number of maximum cliques grows exponentially with the number of cannot-links. The program thus crashed on our data. This issue was also

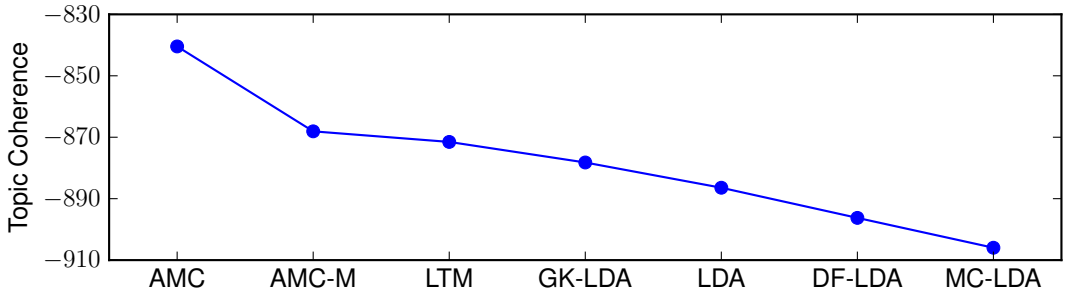

Figure 13. Average Topic Coherence of each model.

noted in (Zhai et al., 2011). For MC-LDA, it increases the number of topics whenever there is not a good topic to put a cannot-link term in. This results in a large number of topics (more than 50), which are unreasonable and give very poor results. Thus, for both DF-LDA and MC-LDA, we can only show their results with must-links,

2. LTM is better than LDA while clearly worse than AMC. The additional information from the cannot-links is shown to help produce much more coherent topics. GK-LDA is slightly better than LDA. The wrong knowledge handling method in GK-LDA can cope with some wrong knowledge, but not as effective as AMC.
3. We also notice that both DF-LDA and MC-LDA are worse than LDA. This is because they assume the knowledge to be correct and lack the necessary mechanism to deal with wrong knowledge. Also, for MC-LDA, it assumes each must-link (or must-set in (Chen et al., 2013c)) represents a distinct sense or meaning. Thus, it assigns only one must-link to each

word and ignores the rest. Then most must-links are not used. This explains also why MC-LDA is worse than DF-LDA.

Iterative improvement (lines 2-5 in Algorithm 5): We found that accumulating cannot-links iteratively is beneficial to AMC. The Topic Coherence value increases slightly from $r = 1$ to 3 and stabilizes at $r = 3$ (Algorithm 5). Figure 13 shows the AMC's result for $r = 3$.

Comparing with LTM using 1000 reviews: To further compare with LTM, we also conducted experiments in the same setting as (Chen and Liu, 2014b), i.e., each test document collection contains also 1,000 reviews (not 100 as in Figure 13). AMC still improves LTM by 47 points in Topic Coherence, showing that AMC can also produce more coherent topics with a large number of test documents.

In summary, we can say that the proposed AMC model generates more coherent topics than all baseline models. Even though DF-LDA, GK-LDA and MC-LDA used our method for knowledge mining, without an effective wrong knowledge handling method, they gave poorer results. The improvements of AMC over all baselines are significant ($p < 0.0001$) based on paired t-tests.

5.4.3 Human Evaluation

Here we want to evaluate the topics based on human judgment. Two human judges who are familiar with Amazon products and reviews were asked to label the generated topics. Since we have a large number of domains (50), we selected 10 domains for labeling. The selection was based on the knowledge of the products of the two human judges. Without enough knowledge, labeling will not be reliable. We labeled the topics generated by AMC, LTM and LDA. LDA is

the basic knowledge-free topic model and LTM is our earlier lifelong learning model that achieves the highest Topic Coherence among the baselines in Figure 13. We again use *Precision*@n (or p@n) as the evaluation measurement. Apart from p@n, we also report the number of coherent topics found by each model.

Results. Figure 14 gives the average *Precision*@5 (top chart) and *Precision*@10 (middle chart) of topical words of only coherent topics (incoherent topics are not considered) for each model in each domain. It is clear that AMC achieves the highest p@5 and p@10 values for all 10 domains. LTM is also better than LDA in general but clearly inferior to AMC. This is consistent with the Topic Coherence results in Section 5.4.2. LDA's results are very poor without a large amount of data. On average, for p@5 and p@10, AMC improves LTM by 8% and 14%, and LDA by 33% and 25% respectively. Significance testing using paired t-tests shows that the improvements of AMC are significant over LTM ($p < 0.0002$) and LDA ($p < 0.0001$) on p@5 and p@10.

The bottom chart of Figure 14 shows that AMC also discovers many more coherent topics than LTM and LDA. On average, AMC discovers 2.4 more coherent topics than LTM and 4.7 more coherent topics than LDA over the 10 domains. These results are remarkable. In many domains, LDA only finds 2-4 coherent topics and never more than 5 (out of 15), which again shows that with a small number of documents (reviews), LDA's results are very poor.

5.4.4 Example Topics

This section shows some example topics produced by AMC, LTM, and LDA in the Camera domain to give a flavor of the kind of improvements made by AMC. Each topic is shown with

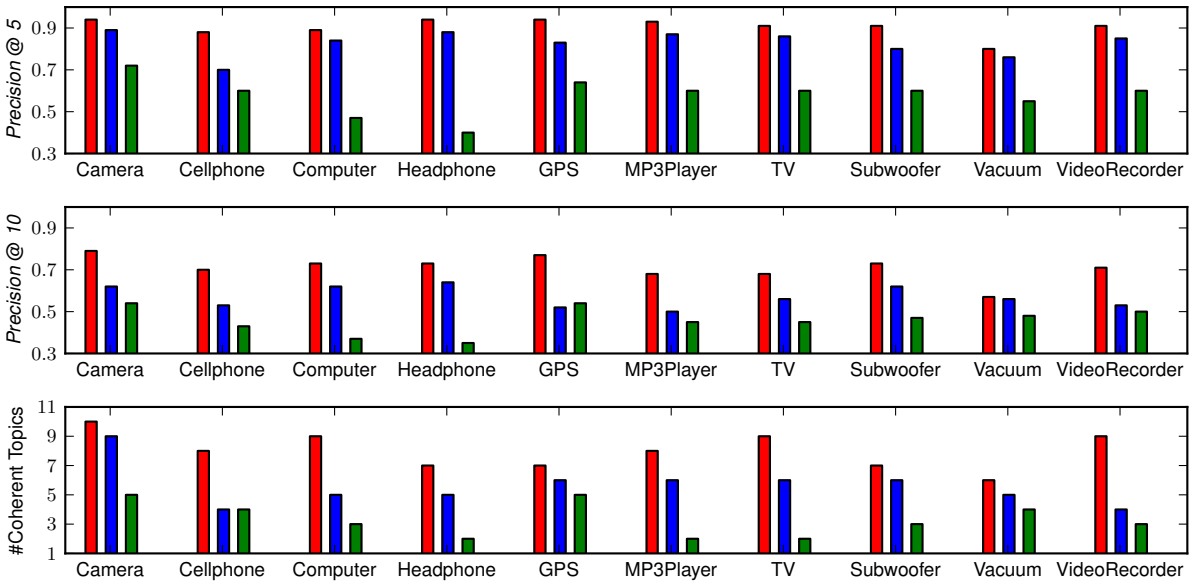

Figure 14. Top & Middle: Topical words *Precision*@5 & *Precision*@10 of coherent topics of each model respectively; Bottom: number of coherent (#Coherent) topics found by each model. The bars from left to right in each group are for AMC, LTM, and LDA.

its top 10 terms. Errors are italicized and marked in red. From Table IV, we can see that AMC discovers many more correct and meaningful topical terms at the top than the baselines. Note that for AMC's topics that were not discovered by the baseline models, we tried to find the best possible matches from the topics of the baseline models. The topic we show for LDA under "Price" is the only one that contains a "Price" related word. Here, the term *price* is mixed with other terms related to the topic "Picture Quality". From the table, we can clearly see that AMC discovers more coherent topics than LTM and LDA. In fact, the coherent topics of AMC are all better than their corresponding topics of LTM and LDA.

Price			Size & Weight		
AMC	**LTM**	**LDA**	**AMC**	**LTM**	**LDA**
money	*shot*	*image*	size	small	*easy*
buy	money	price	small	big	small
price	*review*	*movie*	smaller	size	*canon*
range	price	*stabilization*	weight	pocket	pocket
cheap	cheap	*picture*	compact	*lcd*	*feature*
expensive	*camcorder*	*technical*	hand	*place*	*shot*
deal	*condition*	*photo*	big	*screen*	*lens*
point	*con*	*dslr*	pocket	*kid*	*dslr*
performance	*sony*	*move*	heavy	*exposure*	compact
extra	*trip*	*short*	*case*	*case*	*reduction*

TABLE IV

Example topics of AMC, LTM and LDA from the Camera domain. Errors are italicized and marked in red.

5.4.5 Experiments Using Both Datasets

The above experiments focused on 50 Electronics domains, which have a great deal of topic overlapping. Now we also want to see how AMC performs when the test domain does not have a lot of topic overlapping with the past/prior domains. We use two test data settings: the test set is from (1) an Electronics domain or (2) an non-Electronics domain. For each test set setting, we mine knowledge from topics of (a) 50 Electronics domains (E), (b) 50 non-Electronics domains (NE), and (c) all 100 domains (ALL). For each test set, we use both 100 and 1000 reviews. Figure 15 shows the performance of AMC in each of these settings compared to LDA in terms of Topic Coherence. We can clearly see that AMC performs the best with the knowledge mined from topics of all 100 domains. 50 non-Electronics domains are helpful too

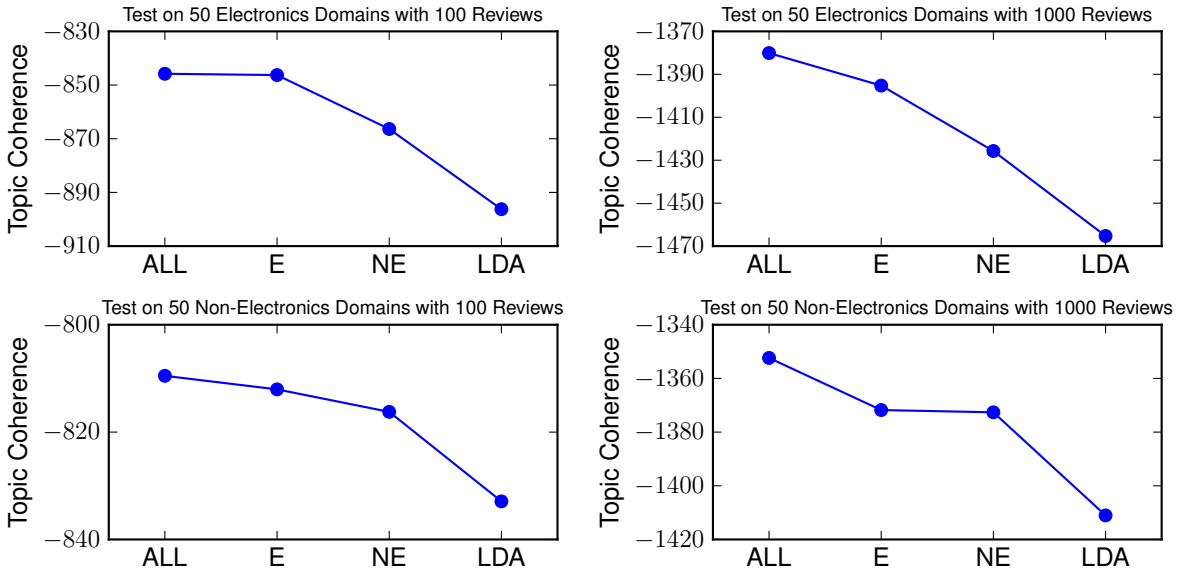

Figure 15. Average Topic Coherence of AMC compared to LDA in different settings (see Section5.4.5). ALL means Electronics (E) + Non-Electronics (NE) and LDA is equivalent to no knowledge.

because they also share some topics such as *price* and *size*. The improvement of AMC in each setting is significant over LDA using paired t-test ($p < 0.0001$). This clearly shows that AMC is able to leverage the useful knowledge from different domains even if the domains are not so related.

5.5 Summary

This chapter proposed an advanced topic model AMC that further improve topic modeling with LML. The AMC model mines two forms of prior knowledge, i.e., must-links and cannot-links, automatically from topics generated from a large number of prior document collections

(the big data). The system also identifies some issues with the automatically mined knowledge. The proposed model AMC not only can exploit the learned knowledge but also can deal with the issues of the mined knowledge to generate more accurate topics. To summarize, we list the LML components (See Chapter 1) in the LTM model:

1. *Past Information Store* (**PIS**): It stores the Topic Base which consists of the topics discovered from past domains (same as LTM model).
2. *Knowledge Base* (**KB**): It uses must-links and cannot-links.
3. *Knowledge Miner* (**KM**): It uses the multiple minimum supports frequent itemset mining (MS-FIM) algorithm
4. *Knowledge-Based Learner* (**KBL**): It proposes *multi-generalized Pólya urn* (M-GPU) model which enables multi-urn interactions. Furthermore, must-link graph is constructed to deal with multiple senses.

CHAPTER 6

LIFELONG SENTIMENT CLASSIFICATION

(This chapter includes and expands on my paper previously published in *Zhiyuan Chen, Nianzu Ma, and Bing Liu. 2015. Lifelong Learning for Sentiment Classification. In ACL 2015, pages 750-756*)

This chapter introduces the integration of lifelong machine learning on classification. In particular, we focus on sentiment classification. But the proposed approach is applicable to other classification problems.

6.1 Sentiment Classification

Sentiment classification is the task of classifying an opinion document as expressing a positive or negative sentiment. (Liu, 2012) and (Pang and Lee, 2008) provided good surveys of the existing research. In this chapter, we tackle sentiment classification from a novel angle using Lifelong Machine Learning (LML). This learning paradigm aims to learn as humans do: retaining the learned knowledge from the past and use the knowledge to help future learning.

One question is why the past learning tasks can contribute to the target domain classification given that the target domain already has labeled training data. The key reason is that the training data may not be fully representative of the test data due to the *sample selection bias* (Heckman, 1979; Shimodaira, 2000; Zadrozny, 2004). In few real-life applications, the training data are fully representative of the test data. For example, in a sentiment classification

application, the test data may contain some sentiment words that are absent in the training data of the target domain, while these sentiment words have appeared in some past domains. So the past domain knowledge can provide the prior polarity information in this situation.

Another question is why the past or source domain labeled data can help even if they are from diverse areas that are not very similar to the target domain. Part of the reason is that in sentiment classification, sentiment bearing words and expressions are largely domain independent. That is, their polarities are often shared across domains. However, simply combining data from these multiple diverse domains may not help because each specific domain also has its domain dependent sentiment terms. It has been shown by transfer learning researchers that sentiment classification is sensitive to the domain from which the training data is extracted (Blitzer et al., 2007), which indicates that a classifier trained using opinion documents from one domain often does not perform well in another domain.

Like most existing sentiment classification papers (Liu, 2012), this paper focuses on binary classification, i.e., positive (+) and negative (−) polarities. But the proposed method is also applicable to multi-class classification. To embed and use the knowledge in building the target domain classifier, we propose a novel optimization method based on the Naïve Bayesian (NB) framework and stochastic gradient descent. The knowledge is incorporated using penalty terms in the optimization formulation. The optimization is able to consider general sentiment terms as well as domain dependent sentiment terms.

6.2 Proposed LSC Technique

6.2.1 Naïve Bayesian Text Classification

Before presenting the proposed method, we briefly review the Naïve Bayesian (NB) text classification as our method uses it as the foundation.

NB text classification (McCallum and Nigam, 1998) basically computes the conditional probability of each word w given each class c_j (i.e., $P(w|c_j)$) and the prior probability of each class c_j (i.e., $P(c_j)$), which are used to calculate the posterior probability of each class c_j given a test document d (i.e., $P(c_j|d)$). c_j is either positive (+) or negative (−) in our case.

The key parameter $P(w|c_j)$ is computed as:

$$P(w|c_j) = \frac{\lambda + N_{c_j,w}}{\lambda |V| + \sum_{v=1}^{|V|} N_{c_j,v}} \tag{6.1}$$

where $N_{c_j,w}$ is the frequency of word w in documents of class c_j. $|V|$ is the size of vocabulary V and λ ($0 \leq \lambda \leq 1$) is used for smoothing.

One of the advantages of NB classification is that when new labeled data arrive, the classifier can quickly update its parameters $P(w|c_j)$ by adding the corresponding counts, without going through the past data. Motivated by this strength, we propose to represent prior knowledge using the empirical counts computed in building past NB classifiers for the source domains (Section 6.2.2).

6.2.2 Components in LSC

This subsection describes our proposed method corresponding to the proposed LML components.

1. *Past Information Store* (**PIS**): In this work, we do not store the original data used in the past learning tasks, but only their results. For each past learning task $\hat{t}$, we store a) $P^{\hat{t}}(w|+)$ and $P^{\hat{t}}(w|-)$ for each word w which are from task $\hat{t}$'s NB classifier (see Equation 6.1); and b) the number of times that w appears in a positive (+) document $N^{\hat{t}}_{+,w}$ and the number of times that w appears in a negative documents $N^{\hat{t}}_{-,w}$.
2. *Knowledge Base* (**KB**): Our knowledge base contains two types of knowledge:
   (a) Document-level knowledge $N^{KB}_{+,w}$ (and $N^{KB}_{-,w}$): number of occurrences of w in the documents of the positive (and negative) class in the past tasks, i.e., $N^{KB}_{+,w} = \sum_{\hat{t}} N^{\hat{t}}_{+,w}$ and $N^{KB}_{-,w} = \sum_{\hat{t}} N^{\hat{t}}_{-,w}$.
   (b) Domain-level knowledge $M^{KB}_{+,w}$ (and $M^{KB}_{-,w}$): number of past tasks in which $P(w|+) > P(w|-)$ (and $P(w|+) < P(w|-)$).
3. *Knowledge Miner* (**KM**). Knowledge miner is straightforward as it just performs counting and aggregation of information in PIS to generate knowledge (see 2(a) and 2(b) above).
4. *Knowledge-Based Learner* (**KBL**): This learner incorporates knowledge using regularization as penalty terms in our optimization. See the details in Section 6.2.4.

6.2.3 Objective Function

In this subsection, we introduce the objective function used in our method. The key parameters that affect NB classification results are $P(w|c_j)$ which are computed using empirical

counts of word w with class c_j, i.e., $N_{c_j,w}$ (Equation 6.1). In binary classification, they are $N_{+,w}$ and $N_{-,w}$. This suggests that we can revise these counts appropriately to improve classification. In our optimization, we denote the optimized variables $X_{+,w}$ and $X_{-,w}$ as the number of times that a word w appears in the positive and negative class. We called them *virtual counts* to distinguish them from empirical counts $N_{+,w}$ and $N_{-,w}$. For correct classification, ideally, we should have the posterior probability $P(c_j|d_i) = 1$ for labeled class c_j, and for the other class c_f, we should have $P(c_f|d_i) = 0$. Formally, given a new domain training data D^t, our objective function is:

$$\sum_{i=1}^{|D^t|} \left(P\left(c_j|d_i\right) - P\left(c_f|d_i\right)\right) \tag{6.2}$$

Here c_j is the actual labeled class of $d_i \in D^t$. In this paper, we use stochastic gradient descent (SGD) to optimize on the classification of each document $d_i \in D^t$. Due to the space limit, we only show the optimization process for a positive document (the process for a negative document is similar). The objective function under SGD for a positive document is:

$$F_{+,i} = P\left(+|d_i\right) - P(-|d_i) \tag{6.3}$$

To further save space, we omit the derivation steps and give the final derivatives below (See the detailed derivation steps in Appendix .1):

$$g\left(\boldsymbol{X}\right) = \left(\frac{\lambda\left|V\right| + \sum_{v=1}^{|V|} X_{+,v}}{\lambda\left|V\right| + \sum_{v=1}^{|V|} X_{-,v}}\right)^{|d_i|} \tag{6.4}$$

Alarm Clock	30.51	Flashlight	11.69	Home Theater System	28.84	Projector	20.24
Baby	16.45	GPS	19.50	Jewelry	12.21	Rice Cooker	18.64
Bag	11.97	Gloves	13.76	Keyboard	22.66	Sandal	12.11
Cable Modem	12.53	Graphics Card	14.58	Magazine Subscriptions	26.88	Vacuum	22.07
Dumbbell	16.04	Headphone	20.99	Movies TV	10.86	Video Games	20.93

TABLE V. Names of the 20 product domains and the proportion of negative reviews in each domain.

$$\frac{\partial F_{+,i}}{\partial X_{+,u}} = \frac{\frac{n_{u,d_i}}{\lambda+X_{+,u}} + \frac{P(-)}{P(+)} \prod_{w \in d_i} \left(\frac{\lambda+X_{-,w}}{\lambda+X_{+,w}}\right)^{n_{w,d_i}} \times \frac{\partial g}{\partial X_{+,u}}}{1 + \frac{P(-)}{P(+)} \prod_{w \in d_i} \left(\frac{\lambda+X_{-,w}}{\lambda+X_{+,w}}\right)^{n_{w,d_i}} \times g(\boldsymbol{X})} - \frac{n_{u,d_i}}{\lambda + X_{+,u}} \tag{6.5}$$

$$\frac{\partial F_{+,i}}{\partial X_{-,u}} = \frac{\frac{n_{u,d_i}}{\lambda+X_{-,u}} \times g(\boldsymbol{X}) + \frac{\partial g}{\partial X_{-,u}}}{\frac{P(+)}{P(-)} \prod_{w \in d_i} \left(\frac{\lambda+X_{+,w}}{\lambda+X_{-,w}}\right)^{n_{w,d_i}} + g(\boldsymbol{X})} \tag{6.6}$$

where n_{u,d_i} is the term frequency of word u in document d_i. $\boldsymbol{X}$ denotes all the variables consisting of $X_{+,w}$ and $X_{-,w}$ for each word w. The partial derivatives for a word u, i.e., $\frac{\partial g}{\partial X_{+,u}}$ and $\frac{\partial g}{\partial X_{-,u}}$, are quite straightforward and thus not shown here. $X^0_{+,w} = N^t_{+,w} + N^{KB}_{+,w}$ and $X^0_{-,w} = N^t_{-,w} + N^{KB}_{-,w}$ are served as a reasonable starting point for SGD, where $N^t_{+,w}$ and $N^t_{-,w}$ are the empirical counts of word w and classes $+$ and $-$ from domain D^t, and $N^{KB}_{+,w}$ and $N^{KB}_{-,w}$ are from knowledge KB (Section 6.2.2). The SGD runs iteratively using the following rules for the positive document d_i until convergence, i.e., when the difference of Equation 6.2 for

two consecutive iterations is less than $1e-3$ (same for the negative document), where γ is the learning rate:

$$X^l_{+,u} = X^{l-1}_{+,u} - \gamma \frac{\partial F_{+,i}}{\partial X_{+,u}}, X^l_{-,u} = X^{l-1}_{-,u} - \gamma \frac{\partial F_{+,i}}{\partial X_{-,u}}$$

6.2.4 Exploiting Knowledge via Penalty Terms

The above optimization is able to update the virtual counts for a better classification in the target domain. However, it does not deal with the issue of domain dependent sentiment words, i.e., some words may change the polarity across different domains. Nor does it utilize the domain-level knowledge in the knowledge base KB (Section 6.2.2). We thus propose to add penalty terms into the optimization to accomplish these.

The intuition here is that if a word w can distinguish classes very well from the target domain training data, we should rely more on the target domain training data in computing counts related to w. So we define a set of words V_T that consists of distinguishable target domain dependent words. A word w belongs to V_T if $P(w|+)$ is much larger or much smaller than $P(w|-)$ in the target domain, i.e., $\frac{P(w|+)}{P(w|-)} \geq \sigma$ or $\frac{P(w|-)}{P(w|+)} \geq \sigma$, where σ is a parameter. Such words are already effective in classification for the target domain, so the virtual counts in optimization should follow the empirical counts ($N^t_{+,w}$ and $N^t_{-,w}$) in the target domain, which are reflected in the L2 regularization penalty term below (α is the regularization coefficient):

$$\frac{1}{2}\alpha \sum_{w \in V_T} \left(\left(X_{+,w} - N^t_{+,w}\right)^2 + \left(X_{-,w} - N^t_{-,w}\right)^2 \right) \tag{6.7}$$

NB-T	NB-S	NB-ST	SVM-T	SVM-S	SVM-ST	CLF	**LSC**
56.21	57.04	60.61	57.82	57.64	61.05	12.87	**67.00**

TABLE VI. Natural class distribution: Average F1-score of the negative class over 20 domains. Negative class is the minority class and thus harder to classify.

NB-T	NB-S	NB-ST	SVM-T	SVM-S	SVM-ST	CLF	**LSC**
80.15	77.35	80.85	78.45	78.20	79.40	80.49	**83.34**

TABLE VII. Balanced class distribution: Average accuracy over 20 domains for each system.

To leverage domain-level knowledge (the second type of knowledge in KB in Section 6.2.2), we want to utilize only those reliable parts of knowledge. The rationale here is that if a word only appears in one or two past domains, the knowledge associated with it is probably not reliable or it is highly specific to those domains. Based on it, we use domain frequency to define the reliability of the domain-level knowledge. For w, if $M^{KB}_{+,w} \geq \tau$ or $M^{KB}_{-,w} \geq \tau$ (τ is a parameter), we regard it as appearing in a reasonable number of domains, making its knowledge reliable. We denote the set of such words as V_S. Then we add the second penalty term as follows:

$$\frac{1}{2}\alpha \sum_{w \in V_S} \left(X_{+,w} - R_w \times X^0_{+,w}\right)^2 + \frac{1}{2}\alpha \sum_{w \in V_S} \left(X_{-,w} - (1 - R_w) \times X^0_{-,w}\right)^2 \tag{6.8}$$

where the ratio R_w is defined as $M^{KB}_{+,w}/(M^{KB}_{+,w} + M^{KB}_{-,w})$. $X^0_{+,w}$ and $X^0_{-,w}$ are the starting points for SGD (Section 6.2.3). Finally, we revise the partial derivatives in Eqs. 4-6 by adding the corresponding partial derivatives of Eqs. 7 and 8 to them.

6.3 Experiments

Datasets. We created a large corpus containing reviews from 20 types of diverse products or domains crawled from Amazon.com (i.e., 20 datasets). The names of product domains are listed in Table V. Each domain contains 1,000 reviews. Following the existing work of other researchers (Blitzer et al., 2007; Pang et al., 2002), we treat reviews with rating > 3 as positive and reviews with rating < 3 as negative.

Natural class distribution: We can see from Table V that every dataset is skewed with significantly more positive reviews, which reflect the natural (or skewed) distribution of the positive and negative class reviews in the real world. We want to experiment with this natural class distribution because it reflects the real-life situation. F1-score is used due to the imbalance.

Balanced class distribution: Since most existing papers on sentiment classification use balanced class data (Blitzer et al., 2007; Pang et al., 2002), we also created a balance dataset with 200 reviews (100 positive and 100 negative) in each domain dataset. This set is smaller because many domains have a very small number of negative reviews due to their highly skewed class distributions. Accuracy is used for evaluation in this balanced setting.

We used unigram features with no feature selection in classification. To deal with negation words (such as "not", "isn't"), we follow (Pang et al., 2002) and add the tag NOT_ to every word between a negation word and the first punctuation following the negation word. For

evaluation, each domain is treated as the target domain with the rest 19 domains as the past domains. All the models are evaluated using 5-fold cross validation.

Baselines. Although there are many transfer learning methods (see Section 2.2.1), they assume there are no or little labeled examples in the target domain but there are a large number of unlabeled examples, which are different from our setting and our goal of improving sentiment classification when good training data in the target domain is available. In our experiments, we compare our proposed LSC model with Naïve Bayes (NB), SVM[1], and CLF (Li and Zong, 2008). Note that NB and SVM can only work on a single domain data. To have a comprehensive comparison, they are fed with three types of training data:

a) labeled training data from the target domain only, denoted by NB-T and SVM-T;

b) labeled training data from all past source domains only, denoted by NB-S and SVM-S;

c) merged (labeled) training data from all past domains and the target domain, referred to as NB-ST and SVM-ST.

For LSC, we empirically set $\sigma = 6$ and $\tau = 6$. The learning rate λ and regularization coefficient α are set to 0.1 empirically. λ is set to 1 for (Laplace) smoothing.

Table VI shows the average F1-scores for the negative class in the natural class distribution, and Table VII shows the average accuracies in the balanced class distribution. We can clearly see that our proposed model LSC achieves the best performance in both cases. In general, NB-S (and SVM-S) are worse than NB-T (and SVM-T), both of which are worse than NB-ST

---

[1]http://www.csie.ntu.edu.tw/~cjlin/libsvm/

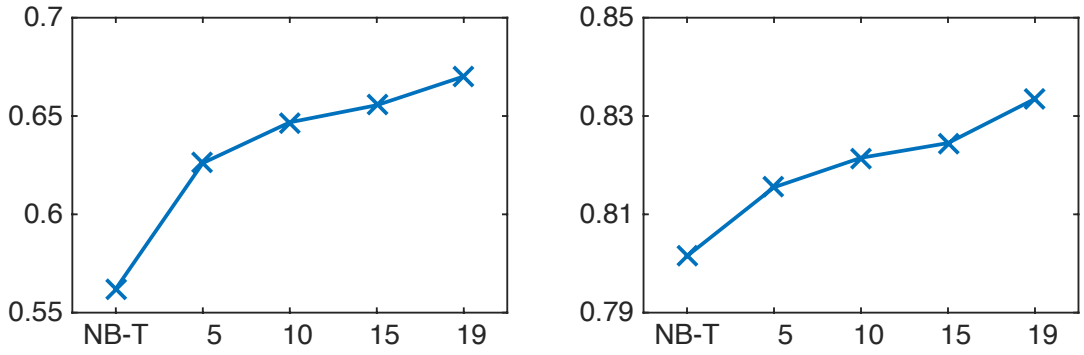

Figure 16. (Left): Negative class F1-score of LSC with #past domains in natural class distribution. (Right): Accuracy of LSC with #past domains in balanced class distribution.

(and SVM-ST). This shows that simply merging both past domains and the target domain data is slightly beneficial. Note that the average F1-score for the positive class is not shown as all classifiers perform very well because the positive class is the majority class (while our model performs slightly better than the baselines). The improvements of the proposed LSC model over all baselines in both cases are statistically significant using paired t-test ($p < 0.01$ compared to NB-ST and CLF, $p < 0.0001$ compared to the others). In the balanced class setting (Table VII), CLF performs better than NB-T and SVM-T, which is consistent with the results in (Li and Zong, 2008). However, it is still worse than our LSC model.

Effects of #Past Domains. Figure 16 shows the effects of our model using different number of past domains. We clearly see that LSC performs better with more past domains, showing it indeed has the ability to accumulate knowledge and use the knowledge to build better classifiers.

6.4 Summary

This chapter proposed a new optimization method LSC that can make use of the past domains to help build a more accurate sentiment classifier for the target domain in the LML manner. The optimization method is based on stochastic gradient descent in the framework of Bayesian probability. Our experimental results using 20 diverse product domains demonstrate the effectiveness of the method. LSC performs significantly better than several baseline methods. We believe that the proposed research which uses the past learning data and results to help new learning is an important research direction as it is analogous to human learning. Without this LML capability, a computer system will not be intelligent.

CHAPTER 7

CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, we studied Lifelong Machine Learning (LML) which extracts and accumulates knowledge automatically from previous domains/tasks and leverages the knowledge to enhance the performance of future learning. We introduced our work using LML on both topic modeling (unsupervised learning) and classification (supervised learning).

7.1 Summary of Contributions

This thesis made the following significant contributions:

1. It proposed the GK-LDA model (Chapter 3) which has three major contributions:
   (a) It proposed the idea of exploiting the general knowledge of lexical semantic relations in topic models to produce coherent topics automatically. To the best of our knowledge, this is the first work that systematically studies such domain independent knowledge in topic models with the aim to provide a general platform to be used in any application domain.
   (b) It proposed a knowledge estimating mechanism in GK-LDA that makes GK-LDA the first knowledge-based topic model that tries to explicitly deal with the problem of wrong input knowledge for an application domain.

(c) A comprehensive evaluation has been conducted to compare GK-LDA with several state-of-the-art baselines based on various qualitative and quantitative measures. Both the codes and the datasets are publically available online.

2. It proposed the LTM model (Chapter 4) which has three main contributions:
   (a) It proposed a novel LML approach to exploit text collections from many domains to learn prior knowledge to guide model inference in order to generate more coherent topics. The process is fully automatic. To our knowledge, it is the first LML method for topic modeling. It also helps deal with big data as seen in Section 4.3.
   (b) It proposed an effective method to mine/learn quality knowledge dynamically from topic matching with raw topics produced using text data from a large number of domains. It proposed a new knowledge estimation method using PMI which is shown to be more effective than that in GK-LDA.
   (c) It created a corpus of 50 product domains for experimentation. Experimental results show that the proposed LTM model achieves significant improvements over state-of-the-art baselines. Both the codes and the datasets are publically available online.
3. It proposed the AMC model (Chapter 5) which has three main contributions:
   (a) It proposed to automatically extract and incorporate cannot-links (in addition to must-links in LTM) into topic modeling with LML.
   (b) It proposed the *multi-generalized Pólya urn* (M-GPU) model which enables the interactions among urns. Such an ability makes it possible to leverage cannot-links discriminatively.

(c) It created a large corpus of 100 product domains and conducted extensive experiments, which showed the superior performance of AMC model. Both the codes and the datasets are publically available online.

4. It proposed the LSC model (Chapter 6) for lifelong sentiment classification, which has three major contributions:
   (a) It proposed a novel lifelong machine learning approach to sentiment classification. To the best our knowledge, this is the first work using LML on the task of sentiment classification.
   (b) It proposed an optimization method that uses penalty terms to embed the knowledge gained in the past and to deal with domain dependent sentiment words to build a better classifier.
   (c) It created a large corpus containing reviews from 20 diverse product domains for extensive evaluation. The experimental results demonstrate the superiority of the proposed LSC model.
5. Last but not the least, it comprehensively studied and discussed lifelong machine learning (LML), including its definitions and related learning paradigms. Although there are much fewer works on LML than on traditional one-shot learning, the benefits of LML for both unsupervised topic modeling and supervised classification shown in this thesis demonstrate its superiority. Furthermore, these promising studies should encourage significantly more participation in this research area.

7.2 Future Directions

There are many interesting future directions to explore in lifelong machine learning, especially in the era of big data. We highlight some of them below:

1. *Knowledge Conflict*: The automatically generated knowledge may contain noise. Such noise can cause conflicts inside the knowledge base. For example, in the problem of topic modeling, we may have a must-link $\{w_1, w_2\}$ stating that two words w_1 and w_2 should be in the same topic while another cannot-link $\{w_1, w_2\}$ may exist and indicate the exact opposite. More sophisticated situations also involve knowledge transitivity. For example, must-links $\{w_1, w_2\}$ and $\{w_2, w_3\}$ do not always indicate must-links $\{w_1, w_3\}$. But in general, words w_1 and w_3 are more likely to share similar semantic meaning than two random words. In the above example, it gets harder when a cannot-link $\{w_1, w_3\}$ is discovered and added into knowledge base. In the problem of sentiment classification, a word w could indicate positive polarity in some domains while expressing negative polarity in some other domains. It is hard to quantify the effects of such word in a new domain. A more accurate estimation method is needed to decide which piece of knowledge the model should trust.
2. *Domain Selection*: Our approach utilizes a large number of domains. But given a huge number of domains, say a million, a strategy for domain selection is necessary for the following two reasons: 1) Mining knowledge from these domains is time-consuming; 2) Many domains may be irrelevant, and thus providing useless or even harmful knowledge. For example, when modeling in the domain "Camera", we prefer knowledge from domains such as "Cellphone" and "Computer" rather than "Diaper".

3. *Big Data*: In the big data era, it is intriguing to apply the proposed technique to a bigger data in terms of both the number of documents and the number of domains. It is crucial to address the challenges of bigger data including efficiency and scalability. Combing MapReduce with the proposed framework is promising to explore. Given a specific task, it is challenging to collect the dataset from a large number of domains, as well as define domains and domain similarity properly.
4. *Lifelong Machine Learning*: This thesis demonstrates the benefits of using the lifelong learning idea to improve topic modeling. We can apply this idea to other machine learning problems, such as clustering. There are several fundamental questions here: Is the information from the other domains helpful in the task of the new domain? What's the representation scheme of knowledge? How can the knowledge be learned and accumulated automatically? Solving these problems will further advance the research in machine learning and data mining.

APPENDICES

.1 Appendix A

This appendix includes the detailed derivation steps for the proposed Lifelong Sentiment Classification (LSC) model.

Recall that our objective function under stochastic gradient descent for each target domain training document d_i is defined as below:

$$F_{+,i} = P\left(c_j|d_i\right) - P(c_f|d_i) \tag{.1}$$

where c_j is the correct label of document d_i and c_f is the wrong label of document d_i. Equation .1 is written as below after plugging probabilities from Naïve Bayesian text classification:

$$\frac{P(c_j)\prod_{w\in d_i} P(w|c_j)^{n_{w,d_i}}}{\sum_{r=1}^{|C|} P(c_r)\prod_{w\in d_i} P(w|c_r)^{n_{w,d_i}}} - \frac{P(c_f)\prod_{w\in d_i} P(w|c_f)^{n_{w,d_i}}}{\sum_{r=1}^{|C|} P(c_r)\prod_{w\in d_i} P(w|c_r)^{n_{w,d_i}}} \tag{.2}$$

Below, we first work on the derivation for a positive document d_i, i.e., $c_j = +$ and $c_f = -$ for document d_i, which gives us:

$$P(+)\prod_{w\in d_i} P(w|+)^{n_{w,d_i}} - P(-)\prod_{w\in d_i} P(w|-)^{n_{w,d_i}} \tag{.3}$$

Here we leave out the denominator of Equation .2 for the time being and work only on the numerators (we will bring the denominator back in Equation .5).

Now we plug 1 (in the submitted paper) into Equation .3:

$$\frac{P\left(+\right)\prod_{w\in d_i}\left(\lambda+X_{+,w}\right)^{n_{w,d_i}}}{\left(\lambda\left|V\right|+\sum_{v=1}^{|V|}X_{+,v}\right)^{|d_i|}}-\frac{P\left(-\right)\beta^{|d_i|}\prod_{w\in d_i}\left(\lambda+X_{-,w}\right)^{n_{w,d_i}}}{\left(\lambda\left|V\right|+\sum_{v=1}^{|V|}X_{+,v}\right)^{|d_i|}} \tag{.4}$$

where $|d_i|$ is the number of words in d_i and $\beta=(\lambda\left|V\right|+\sum\limits_{v=1}^{|V|}X_{+,v})/(\lambda\left|V\right|+\sum\limits_{v=1}^{|V|}X_{-,v})$.

Now let us bring back the denominator in Equation .2, which is nothing but Equation .4 except that instead of subtraction, it uses summation. After canceling the common denominator, we obtain:

$$\frac{P(+)\prod_{w\in d_i}\left(\lambda+X_{+,w}\right)^{n_{w,d_i}}-P(-)\beta^{|d_i|}\prod_{w\in d_i}\left(\lambda+X_{-,w}\right)^{n_{w,d_i}}}{P(+)\prod_{w\in d_i}\left(\lambda+X_{+,w}\right)^{n_{w,d_i}}+P(-)\beta^{|d_i|}\prod_{w\in d_i}\left(\lambda+X_{-,w}\right)^{n_{w,d_i}}} \tag{.5}$$

To make sure Equation .5 gives a positive value for taking log, we first add 1 to it. Then we take the log. These do not change the maximization solution. Last, we negate the equation to make it a minimization problem for gradient descent:

$$\begin{aligned} &log\Bigg(P(+)\prod_{w\in d_i}\left(\lambda+X_{+,w}\right)^{n_{w,d_i}}+P(-)\beta^{|d_i|}\prod_{w\in d_i}\left(\lambda+X_{-,w}\right)^{n_{w,d_i}}\Bigg)\\ &-log\Bigg(2\times P(+)\prod_{w\in d_i}\left(\lambda+X_{+,w}\right)^{n_{w,d_i}}\Bigg) \end{aligned} \tag{.6}$$

Eq. Equation .6 is the objective function that we want to minimize for a positive training document d_i. Note that this objective function is not convex.

We now compute the gradients by taking partial derivatives on Equation .6. We define $g(\boldsymbol{X})$, a function of $\boldsymbol{X}$ where $\boldsymbol{X}$ is a vector consisting of $X_{+,w}$ and $X_{-,w}$ of each word w:

$$g\left(\boldsymbol{X}\right)=\beta^{|d_i|}=\left(\frac{\lambda\left|V\right|+\sum_{v=1}^{|V|}X_{+,v}}{\lambda\left|V\right|+\sum_{v=1}^{|V|}X_{-,v}}\right)^{|d_i|} \tag{.7}$$

The partial derivatives for a word u, i.e., $\frac{\partial g}{\partial X_{+,u}}$ and $\frac{\partial g}{\partial X_{-,u}}$, are quite straightforward and thus not shown here. The final partial derivatives for a word u on Equation .6 is shown below:

$$\frac{\partial F_{+,i}}{\partial X_{+,u}}=\frac{\frac{n_{u,d_i}}{\lambda+X_{+,u}}+\frac{P(-)}{P(+)}\prod_{w\in d_i}\left(\frac{\lambda+X_{-,w}}{\lambda+X_{+,w}}\right)^{n_{w,d_i}}\times\frac{\partial g}{\partial X_{+,u}}}{1+\frac{P(-)}{P(+)}\prod_{w\in d_i}\left(\frac{\lambda+X_{-,w}}{\lambda+X_{+,w}}\right)^{n_{w,d_i}}\times g(\boldsymbol{X})}-\frac{n_{u,d_i}}{\lambda+X_{+,u}} \tag{.8}$$

$$\frac{\partial F_{+,i}}{\partial X_{-,u}}=\frac{\frac{n_{u,d_i}}{\lambda+X_{-,u}}\times g(\boldsymbol{X})+\frac{\partial g}{\partial X_{-,u}}}{\frac{P(+)}{P(-)}\prod_{w\in d_i}\left(\frac{\lambda+X_{+,w}}{\lambda+X_{-,w}}\right)^{n_{w,d_i}}+g(\boldsymbol{X})} \tag{.9}$$

Negative document. We can follow the same process and get the corresponding objective function $F_{-,i}$ for a negative document. Then the final partial derivatives can be obtained following the same process. We gave the final results directly:

$$\frac{\partial F_{-,i}}{\partial X_{+,u}}=\frac{\frac{n_{u,d_i}}{\lambda+X_{+,u}}+\frac{P(-)}{P(+)}\prod_{w\in d_i}\left(\frac{\lambda+X_{-,w}}{\lambda+X_{+,w}}\right)^{n_{w,d_i}}\times\frac{\partial g}{\partial X_{+,u}}}{1+\frac{P(-)}{P(+)}\prod_{w\in d_i}\left(\frac{\lambda+X_{-,w}}{\lambda+X_{+,w}}\right)^{n_{w,d_i}}\times g(\boldsymbol{X})}-\frac{\partial g}{\partial X_{+,u}}\times\frac{1}{g(\boldsymbol{X})} \tag{.10}$$

$$\frac{\partial F_{-,i}}{\partial X_{-,u}}=\frac{\frac{n_{u,d_i}}{\lambda+X_{-,u}}\times g(\boldsymbol{X})+\frac{\partial g}{\partial X_{-,u}}}{\frac{P(+)}{P(-)}\prod_{w\in d_i}\left(\frac{\lambda+X_{+,w}}{\lambda+X_{-,w}}\right)^{n_{w,d_i}}+g(\boldsymbol{X})}-\frac{\frac{n_{u,d_i}}{\lambda+X_{-,u}}\times g(\boldsymbol{X})+\frac{\partial g}{\partial X_{-,u}}}{g(\boldsymbol{X})} \tag{.11}$$

.2 Appendix B

ASSOCIATION FOR COMPUTING MACHINERY, INC. LICENSE TERMS AND CONDITIONS

Oct 13, 2015

This is a License Agreement between Zhiyuan Chen ("You") and Association for Computing Machinery, Inc. ("Association for Computing Machinery, Inc.") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Association for Computing Machinery, Inc., and the payment terms and conditions.

License Number	3727410086395
License date	Oct 13, 2015
Licensed content publisher	Association for Computing Machinery, Inc.
Licensed content publication	Proceedings of the 22nd ACM international conference on Conference on information & knowledge management
Licensed content title	Discovering coherent topics using general knowledge
Licensed content author	Zhiyuan Chen, et al
Licensed content date	Oct 27, 2013
Type of Use	Thesis/Dissertation
Requestor type	Author of this ACM article
Is reuse in the author's own new work?	Yes
Format	Electronic
Portion	Full article
Will you be translating?	No
Order reference number	None
Title of your thesis/dissertation	Lifelong Machine Learning for Topic Modeling and Classification
Expected completion date	Dec 2015
Estimated size (pages)	150
Billing Type	Credit Card
Credit card info	Visa ending in 2515
Credit card expiration	10/2017
Total	8.00 USD
Terms and Conditions	

Rightslink Terms and Conditions for ACM Material

1. The publisher of this copyrighted material is Association for Computing Machinery, Inc. (ACM). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at).

2. ACM reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.
3. ACM hereby grants to licensee a non-exclusive license to use or republish this ACM-copyrighted material* in secondary works (especially for commercial distribution) with the stipulation that consent of the lead author has been obtained independently. Unless otherwise stipulated in a license, grants are for one-time use in a single edition of the work, only with a maximum distribution equal to the number that you identified in the licensing process. Any additional form of republication must be specified according to the terms included at the time of licensing.
*Please note that ACM cannot grant republication or distribution licenses for embedded third-party material. You must confirm the ownership of figures, drawings and artwork prior to use.
4. Any form of republication or redistribution must be used within 180 days from the date stated on the license and any electronic posting is limited to a period of six months unless an extended term is selected during the licensing process. Separate subsidiary and subsequent republication licenses must be purchased to redistribute copyrighted material on an extranet. These licenses may be exercised anywhere in the world.
5. Licensee may not alter or modify the material in any manner (except that you may use, within the scope of the license granted, one or more excerpts from the copyrighted material, provided that the process of excerpting does not alter the meaning of the material or in any way reflect negatively on the publisher or any writer of the material).
6. Licensee must include the following copyright and permission notice in connection with any reproduction of the licensed material: "[Citation] © YEAR Association for Computing Machinery, Inc. Reprinted by permission." Include the article DOI as a link to the definitive version in the ACM Digital Library. Example: Charles, L. "How to Improve Digital Rights Management," Communications of the ACM, Vol. 51:12, © 2008 ACM, Inc. http://doi.acm.org/10.1145/nnnnnn.nnnnnn (where nnnnnn.nnnnnn is replaced by the actual number).
7. Translation of the material in any language requires an explicit license identified during the licensing process. Due to the error-prone nature of language translations, Licensee must include the following copyright and permission notice and disclaimer in connection with any reproduction of the licensed material in translation: "This translation is a derivative of ACM-copyrighted material. ACM did not prepare this translation and does not guarantee that it is an accurate copy of the originally published work. The original intellectual property contained in this work remains the property of ACM."
8. You may exercise the rights licensed immediately upon issuance of the license at the end of the licensing transaction, provided that you have disclosed complete and accurate details of your proposed use. No license is finally effective unless and until full payment is received from you (either by CCC or ACM) as provided in CCC's Billing and Payment terms and conditions.
9. If full payment is not received within 90 days from the grant of license transaction, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted.
10. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.
11. ACM makes no representations or warranties with respect to the licensed material and

adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.
12. You hereby indemnify and agree to hold harmless ACM and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.
13. This license is personal to the requestor and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.
14. This license may not be amended except in a writing signed by both parties (or, in the case of ACM, by CCC on its behalf).
15. ACM hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and ACM (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.
16. This license transaction shall be governed by and construed in accordance with the laws of New York State. You hereby agree to submit to the jurisdiction of the federal and state courts located in New York for purposes of resolving any disputes that may arise in connection with this licensing transaction.
17. There are additional terms and conditions, established by Copyright Clearance Center, Inc. ("CCC") as the administrator of this licensing service that relate to billing and payment for licenses provided through this service. Those terms and conditions apply to each transaction as if they were restated here. As a user of this service, you agreed to those terms and conditions at the time that you established your account, and you may see them again at any time at http://myaccount.copyright.com
18. Thesis/Dissertation: This type of use requires only the minimum administrative fee. It is not a fee for permission. Further reuse of ACM content, by ProQuest/UMI or other document delivery providers, or in republication requires a separate permission license and fee. Commercial resellers of your dissertation containing this article must acquire a separate license.
Special Terms:

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

ASSOCIATION FOR COMPUTING MACHINERY, INC. LICENSE TERMS AND CONDITIONS

Oct 13, 2015

This is a License Agreement between Zhiyuan Chen ("You") and Association for Computing Machinery, Inc. ("Association for Computing Machinery, Inc.") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Association for Computing Machinery, Inc., and the payment terms and conditions.

License Number	3727410311381
License date	Oct 13, 2015
Licensed content publisher	Association for Computing Machinery, Inc.
Licensed content publication	Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining
Licensed content title	Mining topics in documents: standing on the shoulders of big data
Licensed content author	Zhiyuan Chen, et al
Licensed content date	Aug 24, 2014
Type of Use	Thesis/Dissertation
Requestor type	Author of this ACM article
Is reuse in the author's own new work?	Yes
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Order reference number	None
Title of your thesis/dissertation	Lifelong Machine Learning for Topic Modeling and Classification
Expected completion date	Dec 2015
Estimated size (pages)	150
Billing Type	Credit Card
Credit card info	Visa ending in 2515
Credit card expiration	10/2017
Total	8.00 USD
Terms and Conditions	

Rightslink Terms and Conditions for ACM Material

1. The publisher of this copyrighted material is Association for Computing Machinery, Inc. (ACM). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at).

2. ACM reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.
3. ACM hereby grants to licensee a non-exclusive license to use or republish this ACM-copyrighted material* in secondary works (especially for commercial distribution) with the stipulation that consent of the lead author has been obtained independently. Unless otherwise stipulated in a license, grants are for one-time use in a single edition of the work, only with a maximum distribution equal to the number that you identified in the licensing process. Any additional form of republication must be specified according to the terms included at the time of licensing.
*Please note that ACM cannot grant republication or distribution licenses for embedded third-party material. You must confirm the ownership of figures, drawings and artwork prior to use.
4. Any form of republication or redistribution must be used within 180 days from the date stated on the license and any electronic posting is limited to a period of six months unless an extended term is selected during the licensing process. Separate subsidiary and subsequent republication licenses must be purchased to redistribute copyrighted material on an extranet. These licenses may be exercised anywhere in the world.
5. Licensee may not alter or modify the material in any manner (except that you may use, within the scope of the license granted, one or more excerpts from the copyrighted material, provided that the process of excerpting does not alter the meaning of the material or in any way reflect negatively on the publisher or any writer of the material).
6. Licensee must include the following copyright and permission notice in connection with any reproduction of the licensed material: "[Citation] © YEAR Association for Computing Machinery, Inc. Reprinted by permission." Include the article DOI as a link to the definitive version in the ACM Digital Library. Example: Charles, L. "How to Improve Digital Rights Management," Communications of the ACM, Vol. 51:12, © 2008 ACM, Inc. http://doi.acm.org/10.1145/nnnnnn.nnnnnn (where nnnnnn.nnnnnn is replaced by the actual number).
7. Translation of the material in any language requires an explicit license identified during the licensing process. Due to the error-prone nature of language translations, Licensee must include the following copyright and permission notice and disclaimer in connection with any reproduction of the licensed material in translation: "This translation is a derivative of ACM-copyrighted material. ACM did not prepare this translation and does not guarantee that it is an accurate copy of the originally published work. The original intellectual property contained in this work remains the property of ACM."
8. You may exercise the rights licensed immediately upon issuance of the license at the end of the licensing transaction, provided that you have disclosed complete and accurate details of your proposed use. No license is finally effective unless and until full payment is received from you (either by CCC or ACM) as provided in CCC's Billing and Payment terms and conditions.
9. If full payment is not received within 90 days from the grant of license transaction, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted.
10. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.
11. ACM makes no representations or warranties with respect to the licensed material and

adopts on its own behalf the limitations and disclaimers established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.
12. You hereby indemnify and agree to hold harmless ACM and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.
13. This license is personal to the requestor and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.
14. This license may not be amended except in a writing signed by both parties (or, in the case of ACM, by CCC on its behalf).
15. ACM hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and ACM (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.
16. This license transaction shall be governed by and construed in accordance with the laws of New York State. You hereby agree to submit to the jurisdiction of the federal and state courts located in New York for purposes of resolving any disputes that may arise in connection with this licensing transaction.
17. There are additional terms and conditions, established by Copyright Clearance Center, Inc. ("CCC") as the administrator of this licensing service that relate to billing and payment for licenses provided through this service. Those terms and conditions apply to each transaction as if they were restated here. As a user of this service, you agreed to those terms and conditions at the time that you established your account, and you may see them again at any time at http://myaccount.copyright.com
18. Thesis/Dissertation: This type of use requires only the minimum administrative fee. It is not a fee for permission. Further reuse of ACM content, by ProQuest/UMI or other document delivery providers, or in republication requires a separate permission license and fee. Commercial resellers of your dissertation containing this article must acquire a separate license.
Special Terms:

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Copyright policy for the Association for Computational Linguistics (ACL) [1]:

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

- All proprietary rights other than copyright and publication rights transferred to ACL;
- **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;**
- The right to make oral presentation of the material in any forum;
- 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`

Copyright policy of INTERNATIONAL JOINT CONFERENCES ON ARTIFICIAL INTELLIGENCE:

IJCAI hereby grants to the above authors, and the employers for whom the work was performed, royalty-free permission to:

1. retain all proprietary rights (such as patent rights) other than copyright and the publication rights transferred to IJCAI;
2. **personally reuse all or portions of the paper in other works of their own authorship;**
3. make oral presentation of the material in any forum;
4. reproduce, or have reproduced, the above paper for the authors personal use, or for company use provided that IJCAI copyright and the source are indicated, and that the copies are not used in a way that implies IJCAI endorsement of a product or service of an employer, and that the copies per se are not offered for sale. The foregoing right shall not permit the posting of the paper in electronic or digital form on any computer network, except by the author or the authors employer, and then only on the authors or the employers own World Wide Web page or ftp site. Such Web page or ftp site, in addition to the aforementioned requirements of this Paragraph, must provide an electronic reference or link back to the IJCAI electronic server (http://www.ijcai.org), and shall not post other IJCAI copyrighted materials not of the authors or the employers creation (including tables of contents with links to other papers) without IJCAIs written permission;
5. make limited distribution of all or portions of the above paper prior to publication.

6. In the case of work performed under U.S. Government contract, IJCAI grants the U.S. Government royalty-free permission to reproduce all or portions of the above paper, and to authorize others to do so, for U.S. Government purposes. In the event the above paper is not accepted and published by IJCAI, or is withdrawn by the author(s) before acceptance by IJCAI, this agreement becomes null and void.

CITED LITERATURE

[Agrawal and Srikant, 1994] Rakesh Agrawal and Ramakrishnan Srikant. 1994. Fast Algorithms for Mining Association Rules. In *VLDB*, pages 487–499.

[Ammar et al., 2014] Haitham B Ammar, Eric Eaton, Paul Ruvolo, and Matthew Taylor. 2014. Online Multi-Task Learning for Policy Gradient Methods. In *ICML*, pages 1206–1214.

[Ando and Zhang, 2005] Rie Kubota Ando and Tong Zhang. 2005. A High-performance Semi-supervised Learning Method for Text Chunking. In *ACL*, pages 1–9.

[Andreevskaia and Bergler, 2008] Alina Andreevskaia and Sabine Bergler. 2008. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. In *ACL*, pages 290–298.

[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.

[Andrzejewski et al., 2011] David Andrzejewski, Xiaojin Zhu, Mark Craven, and Benjamin Recht. 2011. A framework for incorporating general domain knowledge into latent Dirichlet allocation using first-order logic. In *IJCAI*, pages 1171–1177, jul.

[Argyriou et al., 2008] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex Multi-task Feature Learning. *Machine Learning*, 73(3):243–272.

[Arora et al., 2013] Sanjeev Arora, Rong Ge, Yonatan Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. 2013. A Practical Algorithm for Topic Modeling with Provable Guarantees. In *ICML*, pages 280–288.

[Aue and Gamon, 2005] Anthony Aue and Michael Gamon. 2005. Customizing Sentiment Classifiers to New Domains: A Case Study. In *RANLP*.

[Barker, 2006] Chris Barker. 2006. *Lexical Semantics*. Encyclopedia of Cognitive Science.

[Baxter, 2000] Jonathan Baxter. 2000. A Model of Inductive Bias Learning. *Journal of Artificial Intelligence Research*, 12:149–198.

[Ben-David and Schuller, 2003] Shai Ben-David and Reba Schuller. 2003. Exploiting Task Relatedness for Multiple Task Learning. In *COLT*.

[Bengio, 2011] Yoshua Bengio. 2011. Deep Learning of Representations for Unsupervised and Transfer Learning. *JMLR: Workshop and Conference Proceedings 7*, 7:1–20.

[Bickel et al., 2007] Steffen Bickel, Michael Brückner, and Tobias Scheffer. 2007. Discriminative Learning for Differing Training and Test Distributions. In *ICML*, pages 81–88.

[Blei and McAuliffe, 2010] David M. Blei and Jon D. McAuliffe. 2010. Supervised Topic Models. In *NIPS*, pages 121–128.

[Blei et al., 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022.

[Blitzer et al., 2006] John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain Adaptation with Structural Correspondence Learning. In *EMNLP*, pages 120–128.

[Blitzer et al., 2007] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL*, pages 440–447.

[Blum, 1998] Avrim Blum. 1998. *On-line algorithms in machine learning*. Springer.

[Bollegala et al., 2011] Danushka Bollegala, David J Weir, and John Carroll. 2011. Using Multiple Sources to Construct a Sentiment Sensitive Thesaurus for Cross-Domain Sentiment Classification. In *ACL HLT*, pages 132–141.

[Bonilla et al., 2008] Edwin V Bonilla, Kian M Chai, and Christopher Williams. 2008. Multi-task Gaussian Process Prediction. In *NIPS*, pages 153–160.

[Boyd-Graber et al., 2007] Jordan L Boyd-Graber, David M. Blei, and Xiaojin Zhu. 2007. A Topic Model for Word Sense Disambiguation. In *EMNLP-CoNLL*, pages 1024–1033.

[Branavan et al., 2008] S R K Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. 2008. Learning Document-Level Semantic Properties from Free-Text Annotations. In *ACL*, pages 263–271.

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

[Carenini et al., 2005] Giuseppe Carenini, Raymond T Ng, and Ed Zwart. 2005. Extracting knowledge from evaluative text. In *K-CAP*, pages 11–18.

[Carlson et al., 2010] Andrew Carlson, Justin Betteridge, and Bryan Kisiel. 2010. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, pages 1306–1313.

[Caruana, 1997] Rich Caruana. 1997. Multitask Learning. *Machine learning*, 28(1):41–75.

[Chang et al., 2009] Jonathan Chang, Jordan Boyd-Graber, Wang Chong, Sean Gerrish, 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.

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

[Chen et al., 2011] Jianhui Chen, Jiayu Zhou, and Jieping Ye. 2011. Integrating low-rank and group-sparse structures for robust multi-task learning. In *KDD*, pages 42–50.

[Chen et al., 2013a] Zhiyuan Chen, Bing Liu, and M Hsu. 2013a. Identifying Intention Posts in Discussion Forums. In *NAACL-HLT*, number June, pages 1041–1050.

[Chen et al., 2013b] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013b. Discovering Coherent Topics Using General Knowledge. In *CIKM*, pages 209–218.

[Chen et al., 2013c] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013c. Exploiting Domain Knowledge in Aspect Extraction. In *EMNLP*, pages 1655–1667.

[Chen et al., 2013d] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013d. Leveraging Multi-Domain Prior Knowledge in Topic Models. In *IJCAI*, pages 2071–2077.

[Chen et al., 2014] Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect Extraction with Automated Prior Knowledge Learning. In *ACL*, pages 347–358.

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

[Cheng et al., 2015] Hao Cheng, Hao Fang, and Mari Ostendorf. 2015. Open-Domain Name Error Detection using a Multi-Task RNN. In *EMNLP*, pages 737–746.

[Choi and Cardie, 2010] Yejin Choi and Claire Cardie. 2010. Hierarchical Sequential Learning for Extracting Opinions and their Attributes. In *ACL*, number July, pages 269–274.

[Church and Hanks, 1990] Kenneth Ward Church and Patrick Hanks. 1990. Word Association Norms, Mutual Information, and Lexicography. *Comput. Linguist.*, 16(1):22–29, mar.

[Crammer et al., 2006] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585.

[Dai et al., 2007a] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2007a. Co-clustering Based Classification for Out-of-domain Documents. In *KDD*, pages 210–219.

[Dai et al., 2007b] Wenyuan Dai, Gui-rong Xue, Qiang Yang, and Yong Yu. 2007b. Transferring naive bayes classifiers for text classification. In *AAAI*.

[Dai et al., 2007c] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. 2007c. Boosting for Transfer Learning. In *ICML*, pages 193–200.

[Daume III, 2007] Hal Daume III. 2007. Frustratingly Easy Domain Adaptation. In *ACL*, pages 256–263.

[Daumé III, 2009] Hal Daumé III. 2009. Bayesian Multitask Learning with Latent Hierarchies. In *UAI*, pages 135–142.

[Dekel et al., 2006] Ofer Dekel, Philip M Long, and Yoram Singer. 2006. Online multitask learning. In *Learning Theory*, pages 453–467. Springer.

[Dekel et al., 2007] Ofer Dekel, Philip M Long, and Yoram Singer. 2007. Online Learning of Multiple Tasks with a Shared Loss. *Journal of Machine Learning Research*, 8(10).

[Eidelman et al., 2012] Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic Models for Dynamic Translation Model Adaptation. In *ACL*, pages 115–119.

[Eisenstein et al., 2010] Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric P Xing. 2010. A Latent Variable Model for Geographic Lexical Variation. In *EMNLP*, EMNLP '10, pages 1277–1287.

[Evgeniou and Pontil, 2004] Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized Multi–task Learning. In *KDD*, pages 109–117.

[Fang and Huang, 2012] Lei Fang and Minlie Huang. 2012. Fine Granular Aspect Analysis using Latent Structural Models. In *ACL*, pages 333–337.

[Fei et al., 2012] Geli Fei, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2012. A Dictionary-Based Approach to Identifying Aspects Implied by Adjectives for Opinion Mining. In *COLING*, pages 309–318.

[Fei et al., 2014] Geli Fei, Zhiyuan Chen, and Bing Liu. 2014. Review Topic Discovery with Phrases using the Pólya Urn Model. In *COLING*, pages 667–676.

[Foster and Vohra, 1999] Dean P Foster and Rakesh Vohra. 1999. Regret in the on-line decision problem. *Games and Economic Behavior*, 29(1):7–35.

[Gao and Li, 2011] Sheng Gao and Haizhou Li. 2011. A cross-domain adaptation method for sentiment classification using probabilistic latent analysis. In *CIKM*, pages 1047–1052.

[Gao et al., 2008] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. 2008. Knowledge Transfer via Multiple Model Local Structure Mapping. In *KDD*, pages 283–291.

[Gong et al., 2012] Pinghua Gong, Jieping Ye, and Changshui Zhang. 2012. Robust Multi-task Feature Learning. In *KDD*, pages 895–903.

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

[Guo et al., 2009] Honglei Guo, Huijia Zhu, Zhili Guo, Xiaoxun Zhang, and Zhong Su. 2009. Product feature categorization with multilevel latent semantic association. In *CIKM*, pages 1087–1096.

[He et al., 2011] Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically Extracting Polarity-Bearing Topics for Cross-Domain Sentiment Classification. In *ACL*, pages 123–131.

[Heckman, 1979] James J Heckman. 1979. Sample selection bias as a specification error. *Econometrica: Journal of the econometric society*, pages 153–161.

[Heinrich, 2009] Gregor Heinrich. 2009. A Generic Approach to Topic Models. In *ECML PKDD*, pages 517 – 532.

[Hofmann, 1999] Thomas Hofmann. 1999. Probabilistic Latent Semantic Analysis. In *UAI*, pages 289–296.

[Hu and Liu, 2004] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *KDD*, pages 168–177.

[Hu et al., 2009] Chonghai Hu, Weike Pan, and James T Kwok. 2009. Accelerated gradient methods for stochastic optimization and online learning. In *NIPS*, pages 781–789.

[Hu et al., 2011] Yuening Hu, Jordan Boyd-Graber, and Brianna Satinoff. 2011. Interactive Topic Modeling. In *ACL*, pages 248–257.

[Huang et al., 2013] Yan Huang, Wei Wang, Liang Wang, and Tieniu Tan. 2013. Multi-task deep neural network for multi-label learning. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 2897–2900.

[Jacob et al., 2009] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. 2009. Clustered Multi-Task Learning: A Convex Formulation. In *NIPS*, pages 745–752.

[Jagarlamudi et al., 2012] Jagadeesh Jagarlamudi, Hal Daumé III, and Raghavendra Udupa. 2012. Incorporating Lexical Priors into Topic Models. In *EACL*, pages 204–213.

[Jakob and Gurevych, 2010] Niklas Jakob and Iryna Gurevych. 2010. Extracting Opinion Targets in a Single- and Cross-Domain Setting with Conditional Random Fields. In *EMNLP*, number October, pages 1035–1045.

[Jiang and Zhai, 2007] Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *ACL*, volume 7, pages 264–271.

[Jiang, 2008] Jing Jiang. 2008. A literature survey on domain adaptation of statistical classifiers. Technical report.

[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, feb.

[Joshi et al., 2012] Mahesh Joshi, William W Cohen, Mark Dredze, and Carolyn P Rosé. 2012. Multi-domain Learning: When Do Domains Matter? In *EMNLP*, EMNLP-CoNLL '12, pages 1302–1312.

[Kaelbling et al., 1996] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, pages 237–285.

[Kamar et al., 2013] Ece Kamar, Ashish Kapoor, and Eric Horvitz. 2013. Lifelong Learning for Acquiring the Wisdom of the Crowd. In *IJCAI*, pages 2313–2320.

[Kang et al., 2011] Zhuoliang Kang, Kristen Grauman, and Fei Sha. 2011. Learning with Whom to Share in Multi-task Feature Learning. In *ICML*, pages 521–528.

[Kang et al., 2012] Jeon-hyung Kang, Jun Ma, and Yan Liu. 2012. Transfer Topic Modeling with Ease and Scalability. In *SDM*, pages 564–575.

[Kapoor and Horvitz, 2009] Ashish Kapoor and Eric Horvitz. 2009. Principles of lifelong learning for predictive user modeling. In *User Modeling*, pages 37–46.

[Kawamae, 2010] Noriaki Kawamae. 2010. Latent Interest-Topic Model: Finding the Causal Relationships behind Dyadic Data. In *CIKM*, pages 649–658, oct.

[Kim et al., 2013] Suin Kim, Jianwen Zhang, Zheng Chen, Alice Oh, and Shixia Liu. 2013. A Hierarchical Aspect-Sentiment Model for Online Reviews. In *AAAI*, pages 526–533.

[Kobayashi et al., 2007] Nozomi Kobayashi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting Aspect-Evaluation and Aspect-of Relations in Opinion Mining. In *EMNLP*, number June, pages 1065–1074.

[Ku et al., 2006] Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. 2006. Opinion Extraction, Summarization and Tracking in News and Blog Corpora. In *AAAI-CAAW*, pages 100–107.

[Ku et al., 2009] Lun-Wei Ku, Ting-Hao Huang, and Hsin-Hsi Chen. 2009. Using morphological and syntactic structures for Chinese opinion analysis. In *EMNLP*, pages 1260–1269.

[Kumar et al., 2012] Abhishek Kumar, Hal Daum, and Hal Daume Iii. 2012. Learning Task Grouping and Overlap in Multi-task Learning. In *ICML*, pages 1383–1390.

[Lawrence and Platt, 2004] Neil D Lawrence and John C Platt. 2004. Learning to Learn with the Informative Vector Machine. In *ICML*.

[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*, pages 1630–1639.

[Lee et al., 2007] Su-In Lee, Vassil Chatalbashev, David Vickrey, and Daphne Koller. 2007. Learning a Meta-level Prior for Feature Relevance from Multiple Related Tasks. In *ICML*, pages 489–496.

[Li and Zong, 2008] Shoushan Li and Chengqing Zong. 2008. Multi-domain sentiment classification. In *ACL HLT*, pages 257–260.

[Li et al., 2010] Shoushan Li, Sophia Yat Mei Lee, Ying Chen, Chu-Ren Huang, and Guodong Zhou. 2010. Sentiment Classification and Polarity Shifting. In *COLING*, pages 635–643.

[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 *EMNLP*, pages 1137–1146.

[Li et al., 2012] Fangtao Li, Sinno Jialin Pan, Ou Jin, Qiang Yang, and Xiaoyan Zhu. 2012. Cross-domain Co-extraction of Sentiment and Topic Lexicons. In *ACL*, pages 410–419.

[Li et al., 2013] Shoushan Li, Yunxia Xue, Zhongqing Wang, and Guodong Zhou. 2013. Active learning for cross-domain sentiment classification. In *AAAI*, pages 2127–2133.

[Liao et al., 2005] Xuejun Liao, Ya Xue, and Lawrence Carin. 2005. Logistic Regression with an Auxiliary Data Source. In *ICML*, pages 505–512.

[Lin and He, 2009] Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *CIKM*, pages 375–384, nov.

[Liu et al., 1999] Bing Liu, Wynne Hsu, and Yiming Ma. 1999. Mining association rules with multiple minimum supports. In *KDD*, pages 337–341. ACM.

[Liu et al., 2013] Kang Liu, Liheng Xu, and Jun Zhao. 2013. Syntactic Patterns versus Word Alignment: Extracting Opinion Targets from Online Reviews. *ACL*, pages 1754–1763.

[Liu et al., 2015] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. 2015. Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval. In *NAACL*.

[Liu, 2007] Bing Liu. 2007. *Web data mining*. Springer.

[Liu, 2012] Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

[Lu and Zhai, 2008] Yue Lu and Chengxiang Zhai. 2008. Opinion integration through semi-supervised topic modeling. In *WWW*, pages 121–130.

[Lu et al., 2009] Yue Lu, ChengXiang Zhai, and Neel Sundaresan. 2009. Rated aspect summarization of short comments. In *WWW*, pages 131–140.

[Lu et al., 2011] Bin Lu, Myle Ott, Claire Cardie, and Benjamin K Tsou. 2011. Multi-aspect Sentiment Analysis with Topic Models. In *ICDM Workshops*, pages 81–88.

[Lu et al., 2012] Yue Lu, Hongning Wang, ChengXiang Zhai, and Dan Roth. 2012. Unsupervised discovery of opposing opinion networks from forum discussions. In *CIKM*, pages 1642–1646.

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

[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 *WWW*, pages 171–180.

[Miller, 1995] George A Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

[Mimno et al., 2011] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *EMNLP*, pages 262–272.

[Mitchell et al., 2015] T Mitchell, W Cohen, E Hruschka, P Talukdar, J Betteridge, A Carlson, B Dalvi, M Gardner, B Kisiel, J Krishnamurthy, N Lao, K Mazaitis, T Mohamed, N Nakashole, E Platanios, A Ritter, M Samadi, B Settles, R Wang, D Wi-

jaya, A Gupta, X Chen, A Saparov, M Greaves, and J Welling. 2015. Never-Ending Learning. In *AAAI*.

[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*, WWW '13, pages 909–918.

[Mukherjee and Liu, 2012a] Arjun Mukherjee and Bing Liu. 2012a. Aspect Extraction through Semi-Supervised Modeling. In *ACL*, pages 339–348.

[Mukherjee and Liu, 2012b] Arjun Mukherjee and Bing Liu. 2012b. Mining contentions from discussions and debates. In *KDD*, pages 841–849, aug.

[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 *JCDL*, pages 215–224, jun.

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

[Pan et al., 2010] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *WWW*, pages 751–760.

[Pang and Lee, 2008] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

[Pang et al., 2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *EMNLP*, pages 79–86.

[Pentina and Lampert, 2014] Anastasia Pentina and Christoph H Lampert. 2014. A PAC-Bayesian Bound for Lifelong Learning. In *ICML*, pages 991–999.

[Petterson et al., 2010] James Petterson, Alex Smola, Tibério Caetano, Wray Buntine, and Shravan Narayanamurthy. 2010. Word Features for Latent Dirichlet Allocation. In *NIPS*, pages 1921–1929.

[Popescu and Etzioni, 2005] AM Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *HLT*, pages 339–346.

[Qiu et al., 2011] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*, 37(1):9–27.

[Raina et al., 2007] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. 2007. Self-taught Learning : Transfer Learning from Unlabeled Data. In *ICML*, pages 759–766.

[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.

[Robert and Casella, 2004] Christian P. Robert and George Casella. 2004. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc. Secaucus, NJ, USA, jun.

[Rosen-Zvi et al., 2010] Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. 2010. Learning author-topic models from text corpora. *ACM Transactions on Information Systems*, 28(1):1–38, jan.

[Ruvolo and Eaton, 2013a] Paul Ruvolo and Eric Eaton. 2013a. ELLA: An efficient lifelong learning algorithm. In *ICML*, pages 507–515.

[Ruvolo and Eaton, 2013b] Paul Ruvolo and Eric Eaton. 2013b. Online Multi-Task Learning based on K-SVD. *ICML 2013 Workshop on Theoretically Grounded Transfer Learning.*

[Ruvolo and Eaton, 2014] Paul Ruvolo and Eric Eaton. 2014. Online multi-task learning via sparse dictionary optimization. In *AAAI.*

[Sauper and Barzilay, 2013] Christina Sauper and Regina Barzilay. 2013. Automatic Aggregation by Joint Modeling of Aspects and Values. *J. Artif. Intell. Res. (JAIR)*, 46:89–127.

[Schwaighofer et al., 2004] Anton Schwaighofer, Volker Tresp, and Kai Yu. 2004. Learning Gaussian process kernels via hierarchical Bayes. In *NIPS*, pages 1209–1216.

[Shalev-Shwartz and Srebro, 2008] Shai Shalev-Shwartz and Nathan Srebro. 2008. SVM optimization: inverse dependence on training set size. In *ICML*, pages 928–935.

[Shimodaira, 2000] Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244.

[Silver et al., 2013] Daniel L Silver, Qiang Yang, and Lianghao Li. 2013. Lifelong Machine Learning Systems: Beyond Learning Algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*, pages 49–55.

[Silver, 2013] Daniel L Silver. 2013. On Common Ground: Neural-Symbolic Integration and Lifelong Machine Learning. In *9th International Workshop on Neural-Symbolic Learning and Reasoning NeSy13*, pages 41–46.

[Somasundaran and Wiebe, 2009] Swapna Somasundaran and J Wiebe. 2009. Recognizing stances in online debates. In *ACL*, number August, pages 226–234.

[Stevens and Buttler, 2012] Keith Stevens and PKDAD Buttler. 2012. Exploring Topic Coherence over many models and many topics. In *EMNLP-CoNLL*, number July, pages 952–961.

[Sugiyama et al., 2008] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul V Buenau, and Motoaki Kawanabe. 2008. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, pages 1433–1440.

[Tan et al., 2007] Songbo Tan, Gaowei Wu, Huifeng Tang, and Xueqi Cheng. 2007. A novel scheme for domain-transfer problem in the context of sentiment analysis. In *CIKM*, pages 979–982.

[Teh et al., 2006] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1–30.

[Thrun and Mitchell, 1995] Sebastian Thrun and Tom M Mitchell. 1995. *Lifelong robot learning*. Springer.

[Thrun, 1995] Sebastian Thrun. 1995. Lifelong Learning: A Case Study. Technical report.

[Thrun, 1996a] S Thrun. 1996a. *Explanation-Based Neural Network Learning: A Lifelong Learning Approach*. Kluwer Academic Publishers.

[Thrun, 1996b] Sebastian Thrun. 1996b. Is learning the n-th thing any easier than learning the first? In *NIPS*, pages 640–646.

[Thrun, 1998] Sebastian Thrun. 1998. Lifelong Learning Algorithms. In S Thrun and L Pratt, editors, *Learning To Learn*, pages 181–209. Kluwer Academic Publishers.

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

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

[Wang and Mahadevan, 2008] Chang Wang and Sridhar Mahadevan. 2008. Manifold Alignment Using Procrustes Analysis. In *ICML*, pages 1120–1127.

[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.

[Wei and Croft, 2006] Xing Wei and W Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *SIGIR*, pages 178–185.

[Wu et al., 2009] Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *EMNLP*, number August, pages 1533–1541.

[Xia and Zong, 2011] Rui Xia and Chengqing Zong. 2011. A POS-based Ensemble Model for Cross-domain Sentiment Classification. In *IJCNLP*, pages 614–622. Citeseer.

[Xu et al., 2013] Liheng Xu, Kang Liu, Siwei Lai, Yubo Chen, and Jun Zhao. 2013. Mining Opinion Words and Opinion Targets in a Two-Stage Framework. In *ACL*, pages 1764–1773.

[Xue et al., 2007] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. *Journal of Machine Learning Research*, 8:35–63.

[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*, pages 1640–1649, aug.

[Yang et al., 2006] Hui Yang, Luo Si, and Jamie Callan. 2006. Knowledge Transfer and Opinion Detection in the TREC 2006 Blog Track. In *TREC*.

[Yang et al., 2010] Haiqin Yang, Zenglin Xu, Irwin King, and Michael R Lyu. 2010. Online learning for group lasso. In *ICML*, pages 1191–1198.

[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 *AISTATS*, pages 823–831.

[Yim et al., 2015] Junho Yim, Heechul Jung, ByungIn Yoo, Changkyu Choi, Dusik Park, and Junmo Kim. 2015. Rotating Your Face Using Multi-Task Deep Neural Network. In *CVPR*.

[Yoshida et al., 2011] Yasuhisa Yoshida, Tsutomu Hirao, Tomoharu Iwata, Masaaki Nagata, and Yuji Matsumoto. 2011. Transfer Learning for Multiple-Domain Sentiment Analysis-Identifying Domain Dependent/Independent Word Polarity. In *AAAI*, pages 1286–1291.

[Yu et al., 2005] Kai Yu, Volker Tresp, and Anton Schwaighofer. 2005. Learning Gaussian Processes from Multiple Tasks. In *ICML*, pages 1012–1019.

[Yu et al., 2007] Shipeng Yu, Volker Tresp, and Kai Yu. 2007. Robust Multi-task Learning with T-processes. In *ICML*, pages 1103–1110.

[Yu et al., 2011] Jianxing Yu, Zheng-Jun Zha, Meng Wang, and Tat-Seng Chua. 2011. Aspect Ranking: Identifying Important Product Aspects from Online Consumer Reviews. In *ACL*, pages 1496–1505.

[Zadrozny, 2004] Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *ICML*, page 114. ACM.

[Zhai et al., 2011] Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. 2011. Constrained LDA for grouping product features in opinion mining. In *PAKDD*, pages 448–459, may.

[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, oct.

[Zhao et al., 2012] Yanyan Zhao, Bing Qin, and Ting Liu. 2012. Collocation polarity disambiguation using web-based pseudo contexts. In *EMNLP-CoNLL*, pages 160–170. Association for Computational Linguistics.

[Zhou et al., 2013] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2013. Collective Opinion Target Extraction in Chinese Microblogs. In *EMNLP*, pages 1840–1850, oct.

[Zhuang et al., 2006] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *CIKM*, pages 43–50. ACM Press.

[Zipf, 1932] George Kingsley Zipf. 1932. *Selected Papers of the Principle of Relative Frequency in Language*. Harvard University Press.

VITA

Education

Ph.D., Computer Science, University of Illinois at Chicago, Chicago, Illinois, 2015.

Bachelor of Engineering, Software Engineering, Dalian, China, 2011.

Honors

Fifty For The Future®, Illinois Technology Foundation, 2015.

Dean's Scholar Awards, University of Illinois at Chicago, 2015.

Working Experience

Data Science Intern at Quora Inc., Mountain View, CA, USA. May 2015 - July 2015.

Software Engineering Intern at Twitter Inc., Boston, MA, USA. May 2014 - Aug 2014.

Research Intern at Microsoft Research, Redmond, WA, USA. May 2012 - Aug 2012.

Research Intern at Microsoft Research Asia, Beijing, China. Sept 2010 - May 2011.

Publications

- **Zhiyuan Chen**, Nianzu Ma, and Bing Liu. "Lifelong Learning for Sentiment Classification". *Short Paper*. In Proceedings of **ACL 2015**.
- **Zhiyuan Chen**. "Lifelong Machine Learning for Topic Modeling and Beyond". *Thesis Proposal*. In Proceedings of **NAACL 2015 SRW**.
- Huayi Li, **Zhiyuan Chen**, Arjun Mukherjee, Bing Liu, and Jidong Shao. "Analyzing and Detecting Opinion Spam on a Large-scale Dataset via Temporal and Spatial Patterns". *Short Paper*. In Proceedings of **ICWSM 2015**.

- Yueshen Xu, **Zhiyuan Chen**, Jianwei Yin, Zizheng Wu, and Taojun Yao. "Learning to Recommend with User Generated Content." *Oral Presentation.* In Proceedings of **WAIM 2015**.
- **Zhiyuan Chen** and Bing Liu. "Mining Topics in Documents: Standing on the Shoulders of Big Data". *Oral Presentation.* In Proceedings of **KDD 2014**.
- **Zhiyuan Chen** and Bing Liu. "Topic Modeling using Topics from Many Domains, Lifelong Learning and Big Data". *Oral Presentation.* In Proceedings of **ICML 2014**.
- Geli Fei, **Zhiyuan Chen**, and Bing Liu. "Review Topic Discovery with Phrases using the Polya Urn Model". *Oral Presentation.* In Proceedings of **COLING 2014**.
- **Zhiyuan Chen**, Arjun Mukherjee, and Bing Liu. "Aspect Extraction with Automated Prior Knowledge Learning". *Oral Presentation.* In Proceedings of **ACL 2014**.
- Huayi Li, **Zhiyuan Chen**, Bing Liu, Xiaokai Wei, and Jidong Shao. "Spotting Fake Reviews via Collective Positive-Unlabeled Learning. Oral Presentation." *Short Paper.* In Proceedings of **ICDM 2014**.
- **Zhiyuan Chen**, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. "Exploiting Domain Knowledge in Aspect Extraction". *Oral Presentation.* In Proceedings of **EMNLP 2013**.
- **Zhiyuan Chen**, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. "Discovering Coherent Topics Using General Knowledge". *Oral Presentation.* In Proceedings of **CIKM 2013**.

- **Zhiyuan Chen**, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. "Leveraging Multi-Domain Prior Knowledge in Topic Models". In Proceedings of **IJCAI 2013**.
- **Zhiyuan Chen**, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. "Identifying Intention Posts in Discussion Forums". *Oral Presentation.* In Proceedings of **NAACL-HLT 2013**.
- Yunbo Cao, **Zhiyuan Chen**, Jiamin Zhu, Pei Yue, Chin-Yew Lin, and Yong Yu. "Leveraging Unlabeled Data to Scale Blocking for Record Linkage". In Proceedings of **IJCAI 2011**.

Professional Services

Conferences

- Program committee member in Conference KDD 2016, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
- Program committee member in Conference WWW 2016, International World Wide Web Conference.
- Program committee member in Conference NAACL 2016, Annual Conference of the North American Chapter of the Association for Computational Linguistics.
- Program committee member in Conference WWW 2015, International World Wide Web Conference.
- Program committee member in Conference IJCAI 2015, International Joint Conferences on Artificial Intelligence.

- Program committee member in Conference ICDM 2015, Conference on Empirical Methods in Natural Language Processing.
- Program committee member in Conference EMNLP 2015, Conference on Empirical Methods in Natural Language Processing.
- Program committee member in Conference CIKM 2015, ACM International Conference on Information and Knowledge Management.
- Reviewer in Conference KDD 2015, ACM SIGKDD Conference on Knowledge Discovery and Data Mining.
- Reviewer in Conference AAAI 2015, AAAI Conference on Artificial Intelligence.
- Program committee member in Conference SKG 2015, International Conference on Semantics, Knowledge & Grids.
- Program committee member in Conference CIKM 2014, ACM International Conference on Information and Knowledge Management.
- Program committee member in Conference COLING 2014, International Conference on Computational Linguistics.
- Reviewer in Conferences AAAI 2014, AAAI Conference on Artificial Intelligence.

Journals

- Reviewer in Journal TKDE, Transactions on Knowledge and Data Engineering, 2015.
- Reviewer in Journal PLOS ONE, 2015.
- Reviewer in Journal DMKD, Data Mining and Knowledge Discovery, 2014, 2015.
- Reviewer in Journal TWEB, ACM Transactions on the Web, 2014 (Twice), 2012 (Once).