Mining Topics in Documents: Standing on the Shoulders of Big Data

Zhiyuan Chen and Bing Liu
Department of Computer Science
University of Illinois at Chicago
czyuanacm@gmail.com, liub@cs.uic.edu

ABSTRACT

Topic modeling has been widely used to mine topics from documents. However, a key weakness of topic modeling is that it needs a large amount of data (e.g., thousands of documents) to provide reliable statistics to generate coherent topics. However, in practice, many document collections do not have so many documents. Given a small number of documents, the classic topic model LDA generates very poor topics. Even with a large volume of data, unsupervised learning of topic models can still produce unsatisfactory results. In recently years, knowledge-based topic models have been proposed, which ask human users to provide some prior domain knowledge to guide the model to produce better topics. Our research takes a radically different approach. We propose to *learn as humans do*, i.e., retaining the results learned in the past and using them to help future learning. When faced with a new task, we first mine some reliable (prior) knowledge from the past learning/modeling results and then use it to guide the model inference to generate more coherent topics. This approach is possible because of the big data readily available on the Web. The proposed algorithm mines two forms of knowledge: *must-link* (meaning that two words should be in the same topic) and *cannot-link* (meaning that two words should not be in the same topic). It also deals with two problems of the automatically mined knowledge, i.e., wrong knowledge and knowledge transitivity. Experimental results using review documents from 100 product domains show that the proposed approach makes dramatic improvements over state-of-the-art baselines.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

Keywords

Topic Model; Lifelong Learning; Opinion Aspect Extraction.

1. INTRODUCTION

Topic models, such as LDA [4], pLSA [12] and their extensions, have been popularly used for topic extraction from text documents. However, these models typically need a large amount of data, e.g., thousands of documents, to provide reliable statistics for generating coherent topics. This is a major shortcoming because in practice few document collections have so many documents. For example, in the task of finding product features or aspects from online reviews for opinion mining [13, 19], most products do not even have more than 100 reviews (documents) in a review website. As we will see in the experiment section, given 100 reviews, the classic topic model LDA produces very poor results.

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 are available. However, since topic models perform unsupervised learning, if the data is small, there is simply not enough information 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., [1, 2, 9, 10, 14, 15, 26, 28]) 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 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, few things are really new because we have seen many aspects of it in the past in some other contexts. In machine learning, this paradigm is called *lifelong learning* [30, 31]. The 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. However, our approach is very different from existing lifelong learning methods (see Section 2).

Existing research has focused on the first two approaches. We believe it is high time to create algorithms and build systems that learn as humans do. Lifelong learning is possible in our context due to two key observations:

KDD'14, August 24–27, 2014, New York, NY, USA.

ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623622.

1. Although every domain is different, there is a fair amount of topic overlapping across domains. For example, every product review domain has the topic of *price*, most electronic products share the topic of *battery* and some also have the topic of *screen*. From the topics learned from these domains, we can mine frequently shared terms among the topics. For example, we may find *price* and *cost* frequently appear together in some topics, which indicates that they are likely to belong to the same topic and thus form a must-link. Note that we have the frequency requirement because we want reliable knowledge.
2. From the previously generated topics from many domains, it is also possible to find that *picture* and *price* should not be in the same topic (a cannot-link). This can be done by finding a set of topics that have *picture* as a top topical term, but the term *price* almost never appear at the top of this set of topics, i.e., they are negatively correlated.

Such knowledge can clearly help modeling in a related new domain. The observations also indicate that we need document collections from a large number of domains, which we call the *big data*, to mine enough relevant and reliable must-links and cannot-links to help topic modeling in new domains.

The proposed lifelong learning approach works as follows:

Phrase 1 (*Initialization*): Given n prior document collections $D = \{D_1, \ldots, D_n\}$, a topic model (e.g., LDA) is run on each collection $D_i \in D$ to produce a set of topics S_i. Let $S = \cup_i S_i$, which we call the *prior topics* (or *p-topics* for short). It then mines must-links M from S using a multiple minimum supports frequent itemset mining algorithm [20].

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. 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 4.2. To enable lifelong learning, A^t is incorporated into S, which is used to generate a new set of must-links M.

About knowledge-based topic models, there are two existing ones, DF-LDA [1] and MC-LDA [10], that can use both must-links and cannot-links to help generate better topics. However, both of them 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 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 6).

(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.

In this paper, we propose a new topic model, called AMC (topic modeling with Automatically generated Must-links and Cannot-links), 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.

2. RELATED WORK

Knowledge-based topic models have been proposed to incorporate prior domain knowledge from the user to improve model performance. Existing works such as [1, 9, 26] considered only the must-link type of knowledge (e.g., *price* and *cost*) while [1, 10] also used the cannot-link type of knowledge (e.g., *price* and *picture*). Most of the above models also assume the input knowledge to be correct. [9] is the first work to address the issue of wrong knowledge in topic models by using the ratio of probabilities of two words under each topic. However, [9] only assigns one piece of knowledge (in the form of link or set) to each term, which ignores many pieces of useful knowledge. As shown in Section 6, AMC outperforms it significantly. Other types of knowledge, such as document labels have also been used in [3, 29].

Our work is closely related to transfer learning and lifelong learning. Topic models have been used to help transfer learning [27, 34]. However, transfer learning in these papers is for traditional supervised classification, which is very different from our work of topic extraction. [17] 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. [35] modeled the language gap between topics using a user provided parameter indicating the degree of technicality of the domain. In contrast, our proposed AMC model is 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. In terms of lifelong learning [31, 30], LTM [7] is the first topic model that performs lifelong learning or modeling. It also improves the model in [8], which was not proposed as a lifelong learning model. However, LTM only considers must-links. AMC considers both must-links and cannot-links. AMC also has a more effective must-link mining method and deals with the transitivity or multiple sense problem, which was not tackled in [7]. As we will see in Section 6, AMC achieves dramatic improvements.

Since our experiments are carried out using product reviews, aspect extraction in opinion mining [19] is related. A topic is basically an aspect. Topic models have been used for the task by many researchers [5, 10, 16, 21, 23, 25, 26, 32, 33, 37]. However, none of these models mines must-links or cannot-links automatically to help modeling.

3. OVERALL ALGORITHM

This section introduces the proposed overall algorithm, which follows the lifelong learning idea described in the introduction section. The algorithm consists of two phases:

Phase 1 - Initialization: Given a set of prior document collections $D = \{D_1, \ldots, D_n\}$ from n domains, this step first runs the standard LDA on each domain collection $D_i \in D$ 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.,

Algorithm 1 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: $C \leftarrow C \cup$ MineCannotLinks(S, A^t);
4: $A^t \leftarrow$ GibbsSampling(D^t, N, M, C);
5: **end for**
6: $S \leftarrow$ Incorporate(A^t, S);
7: $M \leftarrow$ MiningMustLinks(S);

$S = \cup_i S_i$. We call S the *prior topic* (or *p-topic*) set. A set of must-links are then mined from S, which will be detailed in Section 4.1. 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 - Lifelong learning 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, we call them the *current topics* (or *c-topics* for short). AMC is given in Algorithm 1. Line 1 runs the proposed Gibbs sampler (introduced in Section 5.3) using only the must-links M generated from the p-topic set (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 the p-topics S (see Section 4.2). 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 6. Currently, the function *Incorporate*(A^t, S) (line 6 in Algorithm 1) 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.

4. 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.

4.1 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 hyper-parameters 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 a set of prior topics (p-topics) S, 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 [18].

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 [20]. 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 [20].

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. 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. 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.

4.2 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 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 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 p-topics S from all domain collections 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 prior domains that w_1 and w_2 appear in different p-topics be N_{diff} and the number of prior 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 0, but N_{diff} can be very small, which may not give reliable cannot-links.

Some extracted cannot-link examples are listed below:

{battery, money}, {life, movie}, {battery, line}
{price, digital}, {money, slow}, {expensive, simple}

5. 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.

5.1 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 [1] 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 [10] 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 5.1.1 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 5.1.2.

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 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.

5.1.1 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

$$\frac{\#T_1 \cap T_2}{Max(\#T_1, \#T_2)} > \pi_{overlap} \quad (1)$$

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 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 for dealing with the transitivity problem.

5.1.2 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 [11]. PMI of two words (or terms) is defined as follows:

$$PMI(w_1, w_2) = log \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \quad (2)$$

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

$$P(w) = \frac{\#D^t(w)}{\#D^t} \quad (3)$$

$$P(w_1, w_2) = \frac{\#D^t(w_1, w_2)}{\#D^t} \quad (4)$$

where $\#D^t(w)$ is the number of documents in D^t that contain the term w and $\#D^t(w_1, w_2)$ is the number of documents that contain both terms w_1 and w_2. $\#D^t$ is the total number of documents in D^t. 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.

5.2 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 [1] and MC-LDA [10], 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 so many cannot-links. As we will see in Section 6, 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 [38], 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 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.

5.3.1 Pólya Urn Model

Traditionally, the Pólya urn model works on colored balls and urns. In the topic model context, a term can be seen as a ball of a certain color and a topic as an urn. The distribution of a topic is reflected by the color proportions of balls in the urn. LDA follows the simple Pólya urn (SPU) model in the sense that when a ball of a particular color is drawn from an urn, the ball is put back to the urn along with a new ball of the same color. The content of the urn changes over time, which gives a self-reinforcing property known as "the rich get richer". This process corresponds to assigning a topic to a term in Gibbs sampling.

The generalized Pólya urn (GPU) model [22, 24] differs from SPU in that, when a ball of a certain color is drawn, two balls of that color are put back along with a certain number of balls of some other colors. These additional balls of some other colors added to the urn increase their proportions in the urn. This is the key technique for incorporating must-links as we will see below.

Instead of involving only one urn at a time as in the SPU and GPU model, 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.

5.3.2 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). 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 must-links, then we do not have the multiple sense problem caused by must-links. If w has 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 5.1.1). The sampling distribution will be given in Section 5.3.3. 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 5.1.2) 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:

$$\lambda_{w',w} = \begin{cases} 1 & w = w' \\ \mu \times PMI(w,w') & (w,w') \text{ is a must-link} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

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 $U^K_{d\in\{1\dots D^t\}}$, 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. [10] 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 5.2, 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^K_d and a term w from U^W_k 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^K_d, and two balls of color w into urn U^W_k.
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^W_k based on matrix $\lambda_{w',w}$ (in Equation 5).
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^W_k and remove it from U^W_k. 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^W_k), skip steps b) and c).
   (b) Produce an urn set $\{U^W_{k'}\}$ such that each urn in it satisfies the following conditions:
   i) $k' \neq k$
   ii) The proportion of balls of color w_c in $U^W_{k'}$ is higher than that of balls of color w_c in U^W_k.
   (c) If $\{U^W_{k'}\}$ is not empty, randomly select one urn $U^W_{k'}$ from it. Put the ball q_c drawn from Step a) into $U^W_{k'}$. Also, remove a ball of color k from urn $U^K_{d_c}$ and put back a ball of k' into urn $U^K_{d_c}$. If $\{U^W_{k'}\}$ is empty, put the ball q_c back to U^W_k.

5.3.3 Sampling Distributions

Based on the above sampling scheme of M-GPU, this subsection 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 [24] 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 5.3.2):

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:

$$P(m_i = m|k) \propto P(w_1|k) \times P(w_2|k) \quad (6)$$

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:

$$P(w|k) \propto \frac{\sum_{w'=1}^{V} \lambda_{w',w} \times n_{k,w'} + \beta}{\sum_{v=1}^{V}(\sum_{w'=1}^{V} \lambda_{w',v} \times n_{k,w'} + \beta)} \quad (7)$$

where $\lambda_{w',w}$ is the promotion matrix in Equation 5. $n_{k,w}$ 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:

$$p(z_i = k|\boldsymbol{z}^{-i}, \boldsymbol{w}, \alpha, \beta, \lambda) \propto \frac{n_{d,k}^{-i} + \alpha}{\sum_{k'=1}^{K}(n_{d,k'}^{-i} + \alpha)} \times \frac{\sum_{\{w',w_i\}\in\{m'\}} \lambda_{w',w_i} \times n_{k,w'}^{-i} + \beta}{\sum_{v=1}^{V}(\sum_{\{w',v\}\in\{m'_v\}} \lambda_{w',v} \times n_{k,w'}^{-i} + \beta)} \quad (8)$$

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.

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:

$$P(q = q_c|\boldsymbol{z}, \boldsymbol{w}, \alpha) \propto \frac{n_{d_c,k} + \alpha}{\sum_{k'=1}^{K}(n_{d_c,k'} + \alpha)} \quad (9)$$

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:

$$P(z_{q_c} = k|\boldsymbol{z}^{-q_c}, \boldsymbol{w}, \alpha, \beta, \lambda, q = q_c) \propto \boldsymbol{I}_{[0,p(w_c|k))]}(P(w_c|z_c)) \times \frac{n_{d,k}^{-q_c} + \alpha}{\sum_{k'=1}^{K}(n_{d,k'}^{-q_c} + \alpha)} \times \frac{\sum_{\{w',w_i\}\in\{m'_c\}} \lambda_{w',w_i} \times n_{k,w'}^{-q_c} + \beta}{\sum_{v=1}^{V}(\sum_{\{w',v\}\in\{m'_v\}} \lambda_{w',v} \times n_{k,w'}^{-q_c} + \beta)} \quad (10)$$

where z_c (the same as z_i sampled from Equation 8) 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.

6. EVALUATION

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

LDA [4]: The classic unsupervised topic model.

DF-LDA [1]: A knowledge-based topic model that can use both must-links and cannot-links, but it assumes all the knowledge is correct.

MC-LDA [10]: A knowledge-based topic model that also use both the must-link and the cannot-link knowledge. It assumes that all knowledge is correct as well.

GK-LDA [9]: 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.

LTM [7]: A lifelong learning topic model that learns only the must-link type of knowledge automatically. It outperformed [8].

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.

6.1 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 1). The second dataset contains reviews from 50 mixed types of non-electronic products or domains (given in the second row of Table 1). 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 [9] to pre-process the dataset. The datasets are publicly available at the authors' websites.

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 $\alpha = 1$, $\alpha = 0.1$, $K = 15$ (#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% [18]. 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 is set to 0.5, which determines the extent of promotion of words in must-links using the M-GPU model.

6.2 Topic Coherence

This sub-section evaluates the topics generated by each model based on the Topic Coherence measure in [24]. Traditionally, topic models are evaluated using perplexity. However, as shown in [6], perplexity does not reflect the semantic coherence of individual topics. It can sometimes be contrary to human judgments. The Topic Coherence measure [24] was proposed as a better alternative for assessing topic quality. It was shown in [24] that Topic Coherence correlates well with human expert labeling. A higher Topic Coherence indicates a higher quality of topics.

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

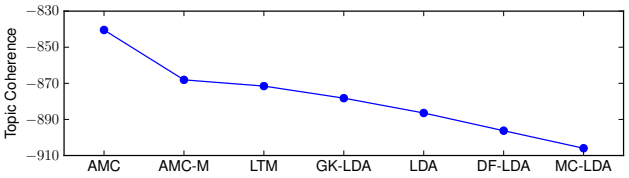

Figure 1: Average Topic Coherence of each model.

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 1 shows the average Topic Coherence value of each model over the 50 test sets. From Figure 1, 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 noted in [36]. 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 [10]) 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 1): 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 1). Figure 1 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 [7], i.e., each test document collection contains also 1,000 reviews (not 100 as in Figure 1). 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.

6.3 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 1. For labeling, we followed the instructions in [24].

Topic Labeling. We first asked the judges to label each topic as *coherent* or *incoherent*. The models that generated the topics for labeling were obscure to the judges. In general, a topic was labeled as coherent if its topical words/terms are semantically *coherent* and together represent a semantic concept; otherwise *incoherent*.

Word Labeling. The topics that were labeled as coherent by both judges were used for word labeling. Each topical word was labeled as *correct* if it was coherently related to the concept represented by the topic (identified in the topic labeling step); otherwise *incorrect*.

The Cohen's Kappa agreement scores for topic labeling and word labeling are 0.873 and 0.860 respectively.

Evaluation Measures. Since topics are rankings of words based on their probabilities, without knowing the exact num-

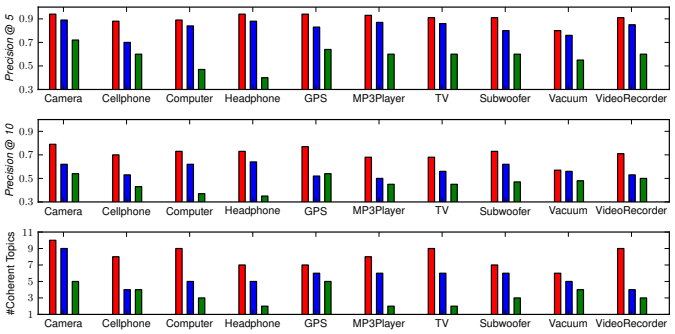

Figure 2: Top & Middle: Topical words *Precision*@5 & *Precision*@10 of coherent topics of each model respectively; Bottom: number of coherent (#Coherent) topics found by each model. The bars from left to right in each group are for AMC, LTM, and LDA.

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

ber of correct topical words/terms, a natural way to evaluate these rankings is to use *Precision*@n (or p@n) which was also used by other researchers, e.g., [9, 37], where n is a rank position. Apart from p@n, we also report the number of coherent topics found by each model.

Results. Figure 2 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 6.2. 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 2 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.

6.4 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 its top 10 terms. Errors are italicized and marked in red. From Table 2, 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.

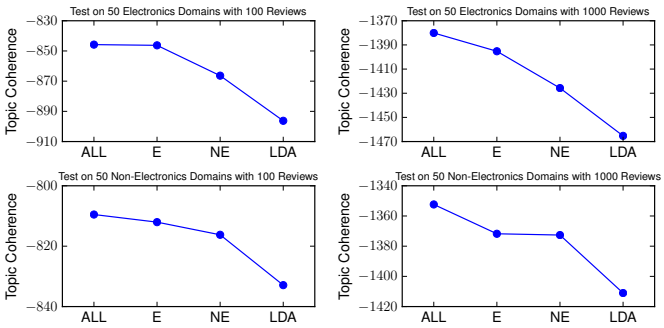

Figure 3: Average Topic Coherence of AMC compared to LDA in different settings (see Section 6.5). ALL means Electronics (E) + Non-Electronics (NE) and LDA is equivalent to no knowledge.

6.5 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 3 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 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.

7. CONCLUSIONS

This paper proposed an advanced topic model AMC that is able to perform lifelong learning. For such learning, it mines prior knowledge from the results of past modeling and uses the knowledge to help future modeling. Our system 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. Experimental results using review collections from 100 domains showed that the proposed AMC model outperforms existing state-of-the-art models significantly. In our future work, we plan to study other aspects of lifelong learning in the topic modeling context, e.g., how to maintain the prior topics and how to incrementally update the must-links knowledge when new topics are added to the prior topic set.

8. ACKNOWLEDGMENTS

This work was supported in part by a grant from National Science Foundation (NSF) under grant no. IIS-1111092.

References

[1] D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via Dirichlet Forest priors. In *ICML*, pages 25–32, 2009.

[2] D. Andrzejewski, X. Zhu, M. Craven, and B. Recht. A framework for incorporating general domain knowledge into latent Dirichlet allocation using first-order logic. In *IJCAI*, pages 1171–1177, 2011.

[3] D. M. Blei and J. D. McAuliffe. Supervised Topic Models. In *NIPS*, pages 121–128, 2007.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[5] S. R. K. Branavan, H. Chen, J. Eisenstein, and R. Barzilay. Learning Document-Level Semantic Properties from Free-Text Annotations. In *ACL*, pages 263–271, 2008.

[6] J. Chang, J. Boyd-Graber, W. Chong, S. Gerrish, and D. M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In *NIPS*, pages 288–296, 2009.

[7] Z. Chen and B. Liu. Topic Modeling using Topics from Many Domains, Lifelong Learning and Big Data. In *ICML*, 2014.

[8] Z. Chen, A. Mukherjee, and B. Liu. Aspect Extraction with Automated Prior Knowledge Learning. In *ACL*, pages 347–358, 2014.

[9] Z. Chen, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Discovering Coherent Topics Using General Knowledge. In *CIKM*, pages 209–218, 2013.

[10] Z. Chen, A. Mukherjee, B. Liu, M. Hsu, M. Castellanos, and R. Ghosh. Exploiting Domain Knowledge in Aspect Extraction. In *EMNLP*, pages 1655–1667, 2013.

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

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

[13] M. Hu and B. Liu. Mining and Summarizing Customer Reviews. In *KDD*, pages 168–177, 2004.

[14] Y. Hu, J. Boyd-Graber, and B. Satinoff. Interactive Topic Modeling. In *ACL*, pages 248–257, 2011.

[15] J. Jagarlamudi, H. D. III, and R. Udupa. Incorporating Lexical Priors into Topic Models. In *EACL*, pages 204–213, 2012.

[16] Y. Jo and A. H. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM*, pages 815–824, Feb. 2011.

[17] J.-h. Kang, J. Ma, and Y. Liu. Transfer Topic Modeling with Ease and Scalability. In *SDM*, pages 564–575, 2012.

[18] B. Liu. *Web data mining*. Springer, 2007.

[19] B. Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[20] B. Liu, W. Hsu, and Y. Ma. Mining association rules with multiple minimum supports. In *KDD*, pages 337–341. ACM, 1999.

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

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

[23] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW*, pages 171–180, 2007.

[24] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, pages 262–272, 2011.

[25] S. Moghaddam and M. Ester. The FLDA Model for Aspect-based Opinion Mining: Addressing the Cold Start Problem. In *WWW*, pages 909–918, 2013.

[26] A. Mukherjee and B. Liu. Aspect Extraction through Semi-Supervised Modeling. In *ACL*, pages 339–348, 2012.

[27] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

[28] J. Petterson, A. Smola, T. Caetano, W. Buntine, and S. Narayanamurthy. Word Features for Latent Dirichlet Allocation. In *NIPS*, pages 1921–1929, 2010.

[29] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*, pages 248–256, 2009.

[30] D. L. Silver, Q. Yang, and L. Li. Lifelong Machine Learning Systems: Beyond Learning Algorithms. In *AAAI Spring Symposium: Lifelong Machine Learning*, 2013.

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

[32] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120, 2008.

[33] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *KDD*, pages 783–792, 2010.

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

[35] S. H. Yang, S. P. Crain, and H. Zha. Bridging the Language Gap: Topic Adaptation for Documents with Different Technicality. In *AISTATS*, volume 15, pages 823–831, 2011.

[36] Z. Zhai, B. Liu, H. Xu, and P. Jia. Constrained LDA for grouping product features in opinion mining. In *PAKDD*, pages 448–459, May 2011.

[37] W. X. Zhao, J. Jiang, H. Yan, and X. Li. Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. In *EMNLP*, pages 56–65, 2010.

[38] G. K. Zipf. *Selective Studies and the Principle of Relative Frequency in Language*. Harvard University Press, 1932.