Named Entity Mining from Click-Through Data Using Weakly Supervised Latent Dirichlet Allocation

Gu Xu[1] Shuang-Hong Yang[1,2] * Hang Li[1]
[1]Microsoft Research Asia, Sigma Center, 49 Zhichun Road, Haidian, Beijing 100190, China
[2]College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA
guxu@microsoft.com, shyang@gatech.edu, hangli@microsoft.com

ABSTRACT

This paper addresses Named Entity Mining (NEM), in which we mine knowledge about named entities such as movies, games, and books from a huge amount of data. NEM is potentially useful in many applications including web search, online advertisement, and recommender system. There are three challenges for the task: finding suitable data source, coping with the ambiguities of named entity classes, and incorporating necessary human supervision into the mining process. This paper proposes conducting NEM by using click-through data collected at a web search engine, employing a topic model that generates the click-through data, and learning the topic model by weak supervision from humans. Specifically, it characterizes each named entity by its associated queries and URLs in the click-through data. It uses the topic model to resolve ambiguities of named entity classes by representing the classes as topics. It employs a method, referred to as Weakly Supervised Latent Dirichlet Allocation (WS-LDA), to accurately learn the topic model with partially labeled named entities. Experiments on a large scale click-through data containing over 1.5 billion query-URL pairs show that the proposed approach can conduct very accurate NEM and significantly outperforms the baseline.

Categories and Subject Descriptors

H.2.8 [[**Database Management**]: Data Mining—*Log Mining*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Query formulation*

General Terms

Algorithms, Experimentation

Keywords

Search Log Mining, Named Entity Recognition, Topic Model, Web Mining

1. INTRODUCTION

Named Entity Mining (NEM) is a text mining task in which the information on the named entities of a class is mined from a large amount of data. The classes can be, for example, movie, game, book and music, and the task can be to mine all the titles of movies in a textual data collection.

NEM is potentially useful in many applications such as web search, online advertisement, and recommender system. For example, if we can identify named entities in search queries using the results of NEM, then we will be able to understand users' interests and Internet trends better, and deliver search results and ads which are more relevant and interesting to the users. Although there were some studies on NEM and certain progresses were made, further investigations on the problem is still needed.

In our view, there are three challenges that stand in the way of achieving high performance NEM: (1) finding a suitable data source for NEM, because the mining task needs sufficient named entity information existing in the data, (2) effectively dealing with ambiguities in classes of named entities because the classes of named entities are highly ambiguous, and (3) properly supervising the mining process, because it would be difficult even for humans to determine the classes of named entities without additional knowledge. In this paper, we propose a new approach to NEM, which (1) makes use of click-through data, (2) employs a topic model on click-through data, (3) learns the topic model using partially labeled seed entities, to overcome the challenges described above. As far as we know, there was no previous work using click-through data and topic model for NEM.

(1) Nowadays, web search becomes the major means for people to access information. Click-through data at search engines, containing queries and clicked URLs associated with the queries, emerges as a rich data source for NEM. According to our statistics, over 70% queries contain named entities. Furthermore, in the click-through data, the contexts of named entities in queries and the websites of associated URLs provide rich clues for identifying the classes of named entities. For example, if named entities appear together with the word "trailer" in queries, then they are likely to be movies. (We refer to such kind of words as contexts in this paper.) Moreover, if named entities are associated with the website "imdb.com" in the click-through data, then they are also likely to be movies. The use of click-through data for NEM has other advantages. The scale of the data is large and thus the wisdom-of-crowds can really be leveraged. The data can be kept updated and novel knowledge on named entities can be kept discovered.

*The work was conducted at Microsoft Research Asia when the second author was intern there.

KDD'09, June 28–July 1, 2009, Paris, France.
Copyright 2009 ACM 978-1-60558-495-9/09/06 ...$5.00.

(2) In this paper, we propose employing a topic model that gives rise to the click-through data. One advantage of using topic models is their capability of modeling ambiguity. For each named entity, we consider the contexts of the named entity in the queries and websites of URLs in the clicks as two types of virtual words[1], and create a virtual document containing the virtual words. Furthermore, we represent the classes of named entity by the topics of the virtual documents. In the topic model, the classes of a named entity (the topics of the corresponding document) depend on the associated contexts and websites (the two types of words in the corresponding document).

(3) Topic model is usually constructed in an unsupervised approach. If we directly learned the topic model using click-through data, then we would not be able to make the learned model very accurate. Specifically, the topics and named entity classes might not be aligned. In this paper, we propose a new learning method called WS-LDA, which constructs the topic model by weakly supervised learning. More precisely the method exploits partially labeled data in the training process. The partially labeled data is created by merging lists of named entities, for example, a movie list and a book list. (Note that such lists can be easily obtained on the web.) The data indicates that a named entity belongs to certain classes, but not excludes the possibility that it belongs to the other classes. WS-LDA can leverage the partially labeled data and make topics and pre-defined classes aligned. The method is general and can be employed in other topic modeling tasks using partially labeled data.

Our method for NEM can be summarized as follows. Given a click-through data set, and several lists of named entities (each list corresponds to one class), it uses the data to create a topic model using WS-LDA. In the model it learns the probabilities of query contexts and click websites of named entities for each class (we call them patterns). The method then applies the acquired patterns into the click-through data to mine new named entities. It outputs the top ranked named entities for each class, as well as patterns of each class.

Our work is in part inspired by the work by Pasca [18]. They propose a method for acquiring named entities from query log data using templates. There are striking differences between their work and ours, however. They use query log data while we use click-through data which contains richer information. They employ a deterministic approach while we take a probabilistic approach which is more robust.

We conducted large-scale experiments on NEM using the proposed approach. A click-through data set containing over 1.5 billion query-URL pairs collected from a commercial search engine was used. Experimental results demonstrate that our approach achieves high performance in NEM and significantly outperforms the baseline method.

2. RELATED WORK

Technologies on Named Entity Recognition (NER) have been developed in natural language processing. Linguistic grammar-based approach (e.g.,[10]), as well as machine learning based approaches (e.g., [2, 5, 16, 9]) have been proposed. The basic idea in NER is to use context information in named entity identification, for example, to see whether "Mr." occurs before the current word when determining whether it is a personal name. A named entity recognition tool can be applied to NEM. For example, given a large-scale web page date set, one can use the tool to automatically identify named entities in the data and aggregate the results for NEM.[2] In this paper, we propose a different approach for NEM by using click-through data.

The recent years have witnessed a surge of interests in knowledge discovery from web search logs. The work most relevant to ours is that by Pasca [18, 19], in which the author proposes a bootstrapping based method for NEM from query log data (not click-through data). There are some clear differences between their work and our work, as explained. Their approach is deterministic not probabilistic. In practice, the classes of named entities are highly ambiguous. For example, "harry potter" can be names of movie, book, and even game. It is hard to deal with the ambiguity problem by taking a deterministic approach.

The use of search click-through data has been proposed in search and other applications. For example, document ranking methods by using click-through data as implicit relevance feedback from users have been proposed (e.g., [13, 1]). Click through data has also be used in query suggestion (e.g. [6, 12]), query expansion (e.g., [7]), and query substitution (e.g., [14]). As far as we know, there was no previous work on using click-through data in mining of named entities.

Topic modeling is a powerful technology for discovering hidden information from data. Topic models such as PLSI[11], LDA [3] and sLDA [4] have been successfully applied to various data mining tasks, such as topic discovery [3], document retrieval [22], citation analysis [8] and social network mining [17]. The topic model used in this paper has a different structure and learning methodology. More specifically, the model generates two types of virtual words for a virtual document, and leverages partially labeled data in learning.

Another related work is learning with only positive examples (i.e., without negative examples). Methods for learning specific types of classifiers in the setting have been proposed, mainly for multi-class classification, but not multi-label classification [23, 15]. Our method of WS-LDA can be regarded as a topic modeling approach to multi-label classification learning with partially labeled data (cf., [20, 21]). Specifically, we define the classifier as topic model, treat unlabeled classes (topics) as hidden values, and employ variational inference to learn the classifier. Note that existing methods of positive-example-based learning cannot be applied to our problem.

3. FROM CLICK-THROUGH DATA

We consider conducing Named Entity Mining from click-through data.

The click-through data is rich in named entity information. We have conducted a manual analysis on 1,000 randomly selected unique queries in a search log from a commercial web search engine. We find that named entities appear very frequently in queries and named entities usually do not occur together in queries. About 70% of the queries contain

[1] To further extend the idea, one can also use page titles or snippets in search results as virtual words.

[2] It may be difficult to extract named entities from search query data using existing NER techniques, because queries are short and not in standard forms, and thus there are not sufficient clues for NER from queries.

single named entities (e.g., "harry potter trailer") and less than 1% of the queries contain two or more named entities (e.g., "kate winslet the reader"). The named entities include people, places, organizations, movies, games, books, music, and etc.

There are strong clues for NEM in click-through data. For example, when searching the trailer of the movie "harry potter", people usually form the query "harry potter trailer", and click the results from movie sites like "movies.yahoo.com". The context "trailer" and the website "movies.yahoo.com" give us strong clues that "harry potter" is the name of a movie. One can also observe that movies usually share similar contexts and/or websites. For queries which do not have contexts (i.e., named entities only) or websites (i.e., no clicked URL), we assume that they have *null* contexts or *null* websites.

Click-through data is a useful source for NEM. The scale of click-through data is extremely large and thus the data can bring in collective knowledge of Internet users on named entities. Furthermore, click-through data keep growing rapidly and thus can provide the most up-to-date information on named entities.

4. PROBLEM FORMALIZATION

We formalize NEM from click-through data as the following problem.

The input to NEM includes a large click-through data set and lists of seed named entities. Each instance in the click-through data is a pair of query and clicked URL, e.g., ("harry potter trailer", "movies.yahoo.com/movie/. . ."). Each list of named entities corresponds to one class. For example, the list of books may contain "harry potter","the long tail", etc, while the list of movies may contain "harry potter", "kung fu panda", etc.

We first create a seed data set with the click-through data and named entity lists. Specifically, we scan the click-through data using the given named entities and obtain all the click-through data containing the named entities. We further group the obtained click-through data by named entities. From each click-through instance of a named entity, we extract the context of the named entity from the query (We employ a heuristics rule of taking the rest of the named entity as context. For example "# trailer" from "harry potter trailer", where # is a place holder), and the website from the clicked URL. Note that the context and the website can be null. The classes to which a named entity belongs are also assigned to the named entity. Note that an entity can be in multiple lists and thus belong to multiple classes. In other words, ambiguities exist in named entity classes. The seed data generation process is depicted in Fig. 1, and Table 1 shows examples of generated seed data.

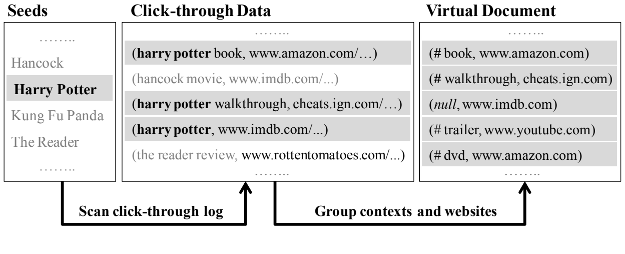
Seeds
Click-through Data
Virtual Document
……..
Hancock
Harry Potter
Kung Fu Panda
The Reader
……..
……..
(harry potter book, www.amazon.com/…)
(hancock movie, www.imdb.com/...)
(harry potter walkthrough, cheats.ign.com/…)
(harry potter, www.imdb.com/...)
(the reader review, www.rottentomatoes.com/...)
……..
……..
(# book, www.amazon.com)
(# walkthrough, cheats.ign.com)
(null, www.imdb.com)
(# trailer, www.youtube.com)
(# dvd, www.amazon.com)
……..
Scan click-through log
Group contexts and websites

Figure 1: Seed Data Generation Process

Table 1: Examples of Seed Data

Named Entity	(Context, Website)	Class
Harry Potter	(# book, amazon.com) (# walkthrough, cheats.ign.com) (*null*, imdb.com)	Book Movie Game
Kung Fu Panda	(# cheat codes, *null*) (# DVD, amazon.com) (# trailer, apple.com)	Game Movie

From the seed data, we can, *in principle*, run a bootstrapping process to mine named entity knowledge. Specifically, we repeat the following two steps: (1) mining new contexts and websites for each class from known named entities of the class, (2) mining new named entities for each class using known contexts and websites of the class. For example, "trailer" and "movies.yahoo.com" can be mined for the movie class from the seed data. The patterns can then be used to mine new movie names.

The biggest challenge here is how to deal with the class ambiguities of named entities. For example, "harry potter" are both book and movie, and thus the contexts and websites of both book and movie are associated with it in the click-through data. Taking a deterministic approach would be difficult to deal with the problem. In this paper, we define and utilize a topic model to overcome the challenge.

5. MODEL AND LEARNING

5.1 Topic Model of Click-through Data

Topic model usually refers to a generative model giving rise to words in documents. It is assumed that each document is associated with a probability distribution of topics, and each topic is associated with a distribution of words. Words in documents are generated through topics.

Here we define a topic model that generates click-through data. In the topic model, topics represent the classes of named entities, virtual words represent context-website pairs, and each virtual document is associated with a named entity. Note that context-website pairs are actually co-occurring two types of words. Words in a document (context website pairs associated with a named entity) is assumed to be probabilistically determined by the topics of the document (classes of the named entity). The difference between topic model of click-through data and topic model of documents is depicted in Fig. 2.

The generative process of the topic model corresponds to the process of search. The searcher first decides a named entity with specific class to search for, then formulates the query (i.e., picks up context), and clicks relevant result (i.e., selects website). For example, if a searcher looks for "harry potter" movie, he would form queries with movie contexts, such as "trailer" and "dvd", and prefers the search results from movie websites, like "imdb.com". If he is interested in "harry potter" book, he would include book contexts, such as "summary" and "notes", and tends to click the results from book websites, like "sparknotes.com". Different named entities have different distributions over classes, for example, entity "harry potter" has a high probability on "movie"; while "halo" has a very high probability on "game".

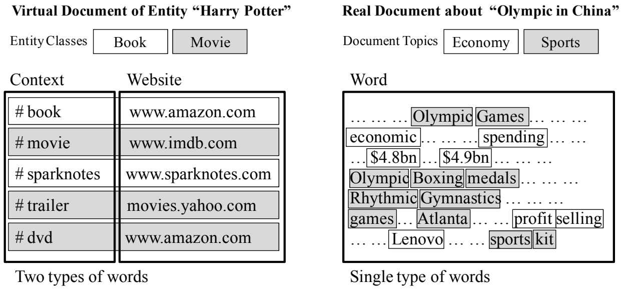

Figure 2: Difference between Topic Models of Click-through Data and Documents

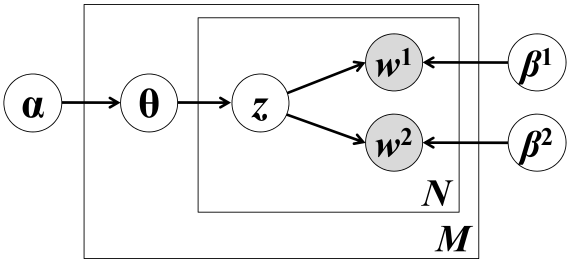

Figure 3: Graphical Representation of Topic Model of Click-through Data.

5.2 Model Definition

We first give the definition of the topic model. Suppose there is a collection of M documents $\mathcal{D} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_M\}$ sharing the set of K topics, and each document is a sequence of N word pairs denoted by $\boldsymbol{w} = \{((w^1)_i, (w^2)_i)\}$, $i = 1, \ldots, N$. Here w^1 and w^2 denote the virtual words of context and website respectively. It is assumed that a document $\boldsymbol{w}$ in the corpus $\mathcal{D}$ is created by the following generative process:

1. Draw topic distribution $\boldsymbol{\theta} \sim \text{Dirichlet}(\boldsymbol{\alpha})$
2. For each word pair
   (a) Draw topic assignment $z_n \sim \text{Multinomial}(\boldsymbol{\theta})$
   (b) Draw word $(w^1)_n \sim \text{Multinomial}(\boldsymbol{\beta^1}_{z_n})$, a multinomial probability conditioned on topic z_n
   (b) Draw word $(w^2)_n \sim \text{Multinomial}(\boldsymbol{\beta^2}_{z_n})$, a multinomial probability conditioned on topic z_n

The model is depicted with a graphical representation in Fig. 3. Given parameter $\Theta = \{\boldsymbol{\alpha}, \boldsymbol{\beta^1}, \boldsymbol{\beta^2}\}$, we obtain the probability of document $\boldsymbol{w}$:

$$p(\boldsymbol{w}|\Theta)=\int p(\boldsymbol{\theta}|\boldsymbol{\alpha})(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\boldsymbol{\theta})\prod_{c=1,2}p((w^c)_n|z_n,\boldsymbol{\beta^c}))d\boldsymbol{\theta} \quad (1)$$

where $\boldsymbol{\theta}$ is a K-dimensional Dirichlet random variable and has the following probability density

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\theta_1^{\alpha_1 - 1}\cdots\theta_K^{\alpha_K - 1}$$

Finally, taking the product of probabilities of documents, we obtain the probability of corpus $\mathcal{D}$ given parameter Θ

$$p(\mathcal{D}|\Theta)$$
$$= \prod_{d=1}^{M}\int p(\boldsymbol{\theta_d}|\boldsymbol{\alpha})(\prod_{n=1}^{N_d}\sum_{z_{dn}}p(z_{dn}|\boldsymbol{\theta_d})\prod_{c=1,2}p((w^c)_{dn}|z_{dn},\boldsymbol{\beta^c}))d\boldsymbol{\theta_d}$$

5.3 Weak Supervision

Conventional topic models are trained by unsupervised learning. Employing the unsupervised learning approach to training the topic model for NEM would not work well, however. This is because for NEM the classes need to be explicitly defined while the topics (classes) in conventional topic models are implicitly learned. There is no guarantee that the topics learned by unsupervised learning will be aligned with the classes defined for named entities. In this paper, we propose introducing supervision in the training process of the topic model for NEM. We refer to the method as WS-LDA (Weakly Supervised Latent Dirichlet Allocation).

The supervision is performed by providing the labels of the possible classes of each seed named entity. That is to say, the classes are not exclusive. For example, "harry potter" may have three classes, "Movie", "Book", and "Game". The weak supervision is equivalent to assuming that the entity (document) has high probabilities on *labeled classes* (labeled topics), but low probabilities on *unlabeled classes* (unlabeled topics). This setting makes WS-LDA unique, and it is also different from the so-called supervised LDA [4].

Given document $\boldsymbol{w}$, the assigned topics are represented as $\boldsymbol{y} = \{y_1, ..., y_K\}$, where y_i takes 1 or 0 when the i-th topic is or is not assigned to the document, and K denotes the number of topics. This weak supervision information will be used as constraints in learning of WS-LDA. WS-LDA tries to maximize the likelihood of data with respect to the model, and at the same time to maximally satisfy the constraints. The constraints are specifically defined as follows.

$$\mathcal{C}(\boldsymbol{y}, \Theta) = \sum_{i=1}^{K} y_i \bar{z}_i \quad (2)$$

Here we define $\bar{z}_i = \frac{1}{N}\sum_{n=1}^{N} z_n^i$, where z_n^i is 1 or 0 when the i-th topic is or is not assigned to the n-th word. That is to say, $\bar{z}_i$ represents the empirical probability of the i-th topic in document $\boldsymbol{w}$. To satisfy the constraints above actually requires the model to meet the following two objectives at the same time: (1) the i-th topic is aligned to the i-th class, and (2) the document $\boldsymbol{w}$ is mainly distributed over labeled topics.

We can combine the likelihood maximization and constraint satisfaction into a single optimization problem with the following objective function.

$$\mathcal{O}(\boldsymbol{w}|\boldsymbol{y}, \Theta) = \log p(\boldsymbol{w}|\Theta) + \lambda \mathcal{C}(\boldsymbol{y}, \Theta) \quad (3)$$

where the likelihood function $p(\boldsymbol{w}|\Theta)$ and constraint function $\mathcal{C}(\boldsymbol{y}, \Theta)$ are represented as in Eqn. (1) and (2) respectively, and λ is coefficient. If λ equals 0, WS-LDA will degenerate to the conventional LDA learning.

Finally, substituting Eqn. (1) and (2) into Eqn. (3) and taking the sum over all documents, we obtain the following total objective function:

$$\mathcal{O}(\mathcal{D}|\mathcal{Y}, \Theta) = \sum_{d=1}^{M}\mathcal{O}(\boldsymbol{w_d}|\boldsymbol{y_d}, \Theta)$$
$$=\sum_{d=1}^{M}\log\int p(\boldsymbol{\theta_d}|\boldsymbol{\alpha})(\prod_{n=1}^{N_d}\sum_{z_{dn}}p(z_{dn}|\boldsymbol{\theta_d})\prod_{c=1,2}p((w^c)_{dn}|z_{dn},\boldsymbol{\beta^c}))d\boldsymbol{\theta_d}$$
$$+\sum_{d=1}^{M}\lambda\sum_{i=1}^{K}y_{di}\bar{z}_{di} \quad (4)$$

5.4 Learning

The learning of WS-LDA is equivalent to maximizing the objective function in Eqn. (4). There might be no analytic solution to the problem as in conventional LDA learning. We employ a variational method similar to that used in [3] to approximate the posterior distribution of the latent variables. The approximate distribution is characterized by the following variational distribution:

$$q(\boldsymbol{\theta}, \boldsymbol{z}|\Lambda) = q(\boldsymbol{\theta}|\boldsymbol{\gamma}) \prod_{n=1}^{N} q(z_n|\phi_n)$$

where $\Lambda = \{\boldsymbol{\gamma}, \phi_{1:N}\}$ are variational parameters. Here, $\boldsymbol{\gamma}$ is the Dirichlet parameter and $\phi_{1:N}$ are the multi-nominal parameters.

Therefore, the objective function for a single document can be derived as follows.

$$\mathcal{O}(\boldsymbol{w}|\boldsymbol{y}, \Theta) = L(\Lambda; \Theta) + D(q(\boldsymbol{\theta}, \boldsymbol{z}|\Lambda)||p(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{w}, \Theta)) \qquad (5)$$

where

$$\begin{aligned} L(\Lambda; \Theta) \quad &= \quad \int \sum_{\boldsymbol{z}} q(\boldsymbol{\theta}, \boldsymbol{z}|\Lambda) \log \frac{p(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{w}|\Theta)}{q(\boldsymbol{\theta}, \boldsymbol{z}|\Lambda)} d\boldsymbol{\theta} \\ &+ \quad \int \sum_{\boldsymbol{z}} q(\boldsymbol{\theta}, \boldsymbol{z}|\Lambda) \lambda \mathcal{C}(\boldsymbol{y}, \Theta) d\boldsymbol{\theta} \end{aligned}$$

Minimizing the KL divergence between the variational posterior probability distribution and the true posterior probability distribution, denoted as $D(q(\boldsymbol{\theta}, \boldsymbol{z}|\Lambda)||p(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{w}, \Theta))$, gives a good approximate distribution of $p(\boldsymbol{\theta}, \boldsymbol{z}|\boldsymbol{w}, \Theta)$. From Eqn. (5) we can see, this is equivalent to maximizing the lower bound $L(\Lambda; \Theta)$ on the objective function $\mathcal{O}(\boldsymbol{w}|\boldsymbol{y}, \Theta)$ with respect to Λ which has the form

$$\begin{aligned} \mathcal{O}(\boldsymbol{w}|\boldsymbol{y}, \Theta) \geq \; & L(\Lambda; \Theta) \\ = \; & E_q[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha})] + E_q[\log p(\boldsymbol{z}|\boldsymbol{\theta})] \\ & + \textstyle\sum_{(c=1,2)} E_q[\log p(\boldsymbol{w^c}|\boldsymbol{z}, \boldsymbol{\beta^c})] \\ & - E_q[\log q(\boldsymbol{\theta})] - E_q[\log q(\boldsymbol{z})] + E_q[\lambda \mathcal{C}(\boldsymbol{y}, \Theta)] \end{aligned}$$

Let $(\beta^c)_{iv}$ be $p((w^c)_n^v = 1|z^i = 1)$ for word v. Each of above terms can be expressed in the following equations (6)$\sim$(11):

$$\begin{aligned} L(\Lambda; \Theta) = & \log \Gamma(\textstyle\sum_{j=1}^K \alpha_j) - \sum_{i=1}^K \log \Gamma(\alpha_i) \\ & + \textstyle\sum_{i=1}^K (\alpha_i - 1)(\Psi(\gamma_i) - \Psi(\sum_{j=1}^K \gamma_j)) && (6) \\ & + \textstyle\sum_{n=1}^N \sum_{i=1}^K \phi_{ni}(\Psi(\gamma_i) - \Psi(\sum_{j=1}^K \gamma_j)) && (7) \\ & + \textstyle\sum_{(c=1,2)} \sum_{n=1}^N \sum_{i=1}^K \sum_{v=1}^V \phi_{ni}(w^c)_n^v \log(\beta^c)_{iv} && (8) \\ & - 2\log \Gamma(\textstyle\sum_{j=1}^K \gamma_j) + 2\sum_{i=1}^K \log \Gamma(\gamma_i) \\ & - \textstyle\sum_{i=1}^K (\gamma_i - 1)(\Psi(\gamma_i) - \Psi(\sum_{j=1}^K \gamma_j)) && (9) \\ & - \textstyle\sum_{n=1}^N \sum_{i=1}^K \phi_{ni} \log \phi_{ni} && (10) \\ & + \frac{\lambda}{N} \textstyle\sum_{n=1}^N \sum_{i=1}^K y_i \phi_{ni} && (11) \end{aligned}$$

Notice that

$$E_q[\bar{z}_i] = E_q[\frac{1}{N} \sum_{n=1}^N z_n^i] = \frac{1}{N} \sum_{n=1}^N E_q[z_n^i] = \frac{1}{N} \sum_{n=1}^N \phi_{ni}$$

is used for the derivation of (11).

A variational expectation-maximization (EM) algorithm is then employed to estimate the model parameters Θ.

E-step:

$$\gamma_i = \alpha_i + \textstyle\sum_{n=1}^N \phi_{ni}$$

$$\phi_{ni} \propto (\beta^1)_{iv} (\beta^2)_{iv} \exp(\Psi(\gamma_i) - \Psi(\textstyle\sum_{j=1}^K \gamma_j) + \frac{\lambda}{N} y_i)$$

M-step:

$$(\beta^1)_{ij} \propto \textstyle\sum_{d=1}^M \sum_{n=1}^{N_d} \phi_{dni} (w^1)_{dn}^j$$

$$(\beta^2)_{ij} \propto \textstyle\sum_{d=1}^M \sum_{n=1}^{N_d} \phi_{dni} (w^2)_{dn}^j$$

Dirichlet parameter $\boldsymbol{\alpha}$ can be updated in the M-step by using an efficient Newton-Raphson method in which the inverted Hessian can be computed in linear time.

5.5 Prediction

The topic model learned by WS-LDA is also used in prediction. Specifically, we calculate the probability $P(c|e)$ for unseen named entities in NEM. This corresponds to estimating the probability of topic given a new document $\boldsymbol{w}$ with the already estimated model Θ. The estimation is then equivalent to approximating the posterior topic distribution $\boldsymbol{\theta}$ of the new document $\boldsymbol{w}$ using the variational inference procedure. It can be easily derived from the variational inference for conventional LDA (cf., [3]).

6. OUR MINING METHOD

In this section, we give a detailed explanation to our proposed method for NEM, as described in Algorithm 1.

The input includes a number of named entity classes, click-through data, and seed named entities and their classes (they are compiled from class-based lists of seed entities). The method first scans the click-through data with the seed named entities to create training data. For each query-URL pair that contains a named entity seed, it extracts the context and website from the pair. It then puts the newly obtained context and website pair to the document associated with the related named entity. After that, it trains a topic model, using WS-LDA. Next, the method tries to find new named entities (beyond the seeds) from the click-through data. Specifically, it scans the click-through data again to find new named entities by using the contexts of classes. For each class, the method then sorts the named entities based on the probability $P(e, c)$ estimated by WS-LDA and outputs the top ranked named entities.

7. EXPERIMENTS

We conducted experiments to verify the effectiveness of the proposed method. In this section, we first introduce the data sets used in the experiments. Then we show both qualitative and quantitative evaluation results by our method and the baseline. Finally, we experimentally demonstrate the importance of using click information as well as the supervision information in our method.

7.1 Baseline and Parameter

As baseline for NEM, we used the method of using query log data and a deterministic approach, proposed in [18]. We refer to it as *QueDet*. Note that our method uses click-through and a probabilistic approach.

In the experiments, the parameter λ in WS-LDA was set to 1 by default.

7.2 Data Set

Four classes were considered in our experiments, including "Movie", "Game", "Book", and "Music". For each class, we collected a set of named entities from some authoritative web

Algorithm 1 Named Entity Mining Algorithm

```
-Input: named entity classes $\mathcal{C} = \{c_1, \ldots, c_K\}$,
        click through data $\Omega = \{(q, u)\}$,
        named-entities and their classes $\mathcal{S} = \{(e, C_e)\}$
-Output: top ranked new entities for each class
-Variable: $\mathcal{T}$= context vocabulary
           $\mathcal{W}$= website vocabulary
           $\mathcal{D}$= corpus of documents
           $\mathcal{E}$= named entity set
Initialize $\mathcal{T} = \varnothing$, $\mathcal{W} = \varnothing$, $\mathcal{D} = \varnothing$, $\mathcal{E} = \varnothing$
for each item $(q, u)$ in $\Omega$
    for each seed $(e, C_e)$ in $\mathcal{S}$
        if ($q$ contains $e$)
            $t$=ExtractContext($q, e$)
            $w$=ExtractWebsite($u$)
            $\mathcal{T} = \mathcal{T} \cup t$
            $\mathcal{W} = \mathcal{W} \cup w$
            Put $(t, w)$ into document $W_e \in \mathcal{D}$
Train topic model using $\mathcal{D}$
for each item $(q, u)$ in $\Omega$
    for each context $t$ in $\mathcal{T}$
        if ($q$ contains $t$)
            $e$=ExtractEntity($q, t$)
            $w$=ExtractWebsite($u$)
            $\mathcal{E} = \mathcal{E} \cup e$
            put $(t, w)$ into document $W_e \in \mathcal{D}$
for each candidate $e$ in $\mathcal{E}$
    estimate $P(e|c)$
Sort $e \in \mathcal{E}$ for each $c$ according to $P(e, c)$
Output the top ranked named entities in each $c$.
```

Table 2: Training Data for Topic Model Learning

	Ave.	Min.	Max.
Unique Contexts in Doc.	94.55	1	1506
Unique Websites in Doc.	216.87	1	6153
Doc. Length (With Freq.)	6271.84	1	1.69×10^6

sources (e.g., "top movies" on www.imdb.com, and "best-sellers in books" on amazon.com). Finally, we had about 1,000 seeds for each class, and 3,767 unique named entities in total.

The experiments were conducted on a repository of click-through data randomly sampled from a commercial web search engine. The click-through data consists of about *1.5 billion* query URL pairs, *240 million* unique queries, and *17 million* unique URLs.

7.3 Experiment on NEM

7.3.1 Experiment Procedure

We applied our NEM method described in Section 6 to the click-through and seed named entity data. We first created a topic model using WS-LDA. The training data contains 3,246 named entities, as well as 3,293 contexts and 6,891 websites. Statistics on the training data of the topic model is shown in Table 2. We then applied the topic model to the click-through data to extract new named entities with the learned patterns (contexts and websites).

Next, we applied QueDet to the query part of the click-through data to extract named entities and patterns for NEM. Except the URL part, QueDet exactly utilized the same information as our method.

7.3.2 Named Entities

Table 3 shows the top 20 named entities for each class selected by our method and QueDet. We can see that our method worked well and the top ranked 20 named entities are very reasonable. DueDet ranked some entities at the top which are correct but not very popular, like "lego batman" in the game class. Also, we can observe some errors in DueDet's results, like taking music type "hip pop" as a title of music.

To make a comparison, we also conducted a quantitative experiment on top 250 results. The top 250 results were manually judged as "correct", "partially correct" and "incorrect" by human annotators. "Partially correct" was used for the cases in which the mined named entities contain additional information, like "the latest harry potter" or "harry potter movie" for named entity "harry potter", or "chris brown with you" for the music "with you".

The performance was evaluated in terms of Precision@N (or P@N) where P@N is defined as follows:

$$P@N = \frac{N_c + 0.5 * N_{pc}}{N}$$

where N_c denotes the number of results labeled as "correct" in top ranked N results, and N_{pc} denotes the number of results labeled as "partially correct" in top ranked N results.

Table 4 shows the P@N scores by our method and QueDet for each of the classes. We can see that our method significantly outperforms QueDet (The t-test shows that the improvement is statistically significant $p < 0.01$).

7.3.3 Contexts and Websites

The top contexts and top websites mined by our method and QueDet are shown in Table 5 and Table 6, and the errors have been marked by underline. In Table 5, # denotes a place holder for named entities.

We can see that our method gives more reasonable results than QueDet. The contexts belonging to different classes are slightly mixed-up in QueDet, e.g. "# lyric" in the movie class. This might be due to its low ability in resolving ambiguities. In contrast, it seems that our method, which is based on a probabilistic approach is more robust against ambiguities and can generate more accurate results.

QueDet does not mine from websites. We can take websites as contexts, apply QueDet to the data, and rank websites for each class. We refer to it as QueDet-URL here. The top websites mined by our method also appear to be reasonable and interesting. One may think that some results are incorrect, e.g. "www.download.com" for the "game" class. Actually they are not. The website "www.download.com" provides download services for various software tools including games on different platforms. We can see that the website information provides useful clues for mining of named entities.

From the results, we can also observe some interesting patterns of searches. For example, most searchers search songs for lyrics, because "# lyrics" is the most popular context used for searching songs, and most of the top web sites are lyrics web sites. Furthermore, we can also see that most game fans use search engines to look for "cheat codes".

7.4 Experiment on Our Method

In this section, we compared the performances of our method under different settings.

Table 3: Top Named Entities for Each Class.

Movie		Game	
QueDet	Our Method	QueDet	Our Method
the dark knight	the dark knight	call of duty 4	call of duty 4
twilight	dark knight	lego batman	need for speed carbon
dark knight	tropic thunder	spore	sims 2
tropic thunder	titanic	sims 2	call of duty
iron man	pineapple express	need for speed carbon	mercenaries 2
transformers	step brothers	call of duty	need for speed most wanted
the notebook	300	need for speed most wanted	call of duty 2
pineapple express	what happens in vegas	mercenaries 2	the sims 2
titanic	eagle eye	lego indiana jones	spore
kung fu panda	the house bunny	oblivion	oblivion
mamma mia	death race	halo	madden 08
batman	prom night	fable	crysis
saw 5	beverly hills chihuahua	the sims 2	guitar hero 3
mama mia	house bunny	madden 09	star wars the force unleashed
star wars	stepbrothers	dead space	bioshock
finding nemo	shrek	call of duty 2	need for speed pro street
high school musical 3	the happening	bioshock	grand theft auto
step brothers	wanted	star wars the force unleashed	fallout 3
wanted	mama mia	madden 08	dead space
juno	quarantine	guitar hero 3	half life 2

Book		Music	
QueDet	Our Method	QueDet	Our Method
to kill a mockingbird	lord of the flies	crazy	tattoo
beowulf	animal farm	obama	viva la vida
lord of the flies	to kill a mockingbird	tattoo	crazy
animal farm	scarlet letter	amazing grace	paper planes
the crucible	huckleberry finn	disturbia	i kissed a girl
hamlet	the scarlet letter	angel	i m yours
scarlet letter	the great gatsby	iron man	i can only imagine
pride and prejudice	the crucible	destiny	lollipop
of mice and men	macbeth	wallpaper	no air
the scarlet letter	brave new world	gospel	apologize
macbeth	fahrenheit 451	sweet home alabama	hey there delilah
great expectations	the odyssey	hurricane	this is me
the outsiders	great expectations	definition	better in time
the great gatsby	of mice and men	hip hop	smoke on the water
1984	things fall apart	viva la vida	all summer long
huckleberry finn	catcher in the rye	i kissed a girl	im yours
romeo and juliet	jane eyre	american pie	one step at a time
the kite runner	the giver	paper planes	somewhere over the rainbow
jane eyre	wuthering heights	halo	bubbly
brave new world	a separate peace	superman	house of the rising sun

We evenly partitioned the training data (containing 3,246 seed named entities) into 10 folds. Then we conducted 10-fold cross-validation on prediction of named entity class using likelihood. The class likelihood with respect to named entity e is defined as $\sum_{i=1}^{K} y_i P(c_i|e)$, where y_i takes 1 or 0 when the i-th class is or is not assigned to e. It measures how consistent the predictions are with human labels.

7.4.1 *Click v.s. No-click*

Our method makes use of click-through data for NEM, rather than just query log. If we ignore the click information in the click-through data, and apply the same approach, then we end up with a new setting for our method which only resorts to context information. We refer to it as WS-LDA (No-click). Fig. 4 shows the class likelihood in the 10 runs. We can see that without using website information, the performances of WS-LDA deteriorate significantly. The average class likelihood changes from 0.65 to 0.62 (The t-test results show that the difference is statistically significant. $p < 0.01$). This result indicates that using click through data for NEM is superior to only using query log data.

7.4.2 *Supervision v.s. No-supervision*

Our method leverages supervision and conducts weakly supervised learning (WS-LDA). We compared the performances of WS-LDA and the case in which supervision is withheld. We refer to the setting as WS-LDA (No-supervision). Fig. 4 shows the class likelihood in the 10 runs. From the result, we can see that WS-LDA significantly outperforms WS-LDA (No-supervision) ($p < 0.01$). WS-LDA (No-supervision) even significantly underperforms WS-LDA (No-click) ($p < 0.01$). The result demonstrates the importance of supervision in the topic model learning.

Note that in WS-LDA (No-supervision) there is actually no explicit label output. We enumerate all the $K!$ possible alignments of the classes with the exiting classes in each run and report the highest accuracies achieved by the method here. It can be considered as an upper-bound of WS-LDA (No-supervision).

8. CONCLUSION

In this paper, we have proposed a new method for mining named entities from a large amount of data, specifically,

Table 4: Precision of Named Entity Indentification for Each Class (P@N)

	Movie		Game		Book		Music		Average-Class	
	QueDet	Ours	QueDet	Ours	QueDet	Ours	QueDet	Ours	QueDet	Ours
P@50	0.990	1.000	1.000	1.000	1.000	1.000	0.960	0.990	0.988	0.998
P@100	0.990	1.000	0.980	1.000	0.980	1.000	0.935	0.985	0.971	0.996
P@150	0.993	0.993	0.963	1.000	0.957	1.000	0.947	0.983	0.965	0.994
P@200	0.993	0.995	0.958	1.000	0.942	0.995	0.935	0.985	0.957	0.994
P@250	0.990	0.996	0.954	0.992	0.921	0.988	0.932	0.970	0.949	0.987

Table 5: Top Ranked Contexts for Each Class.

Movie		Game	
QueDet	Our Method	QueDet	Our Method
# movie	# movie	# cheats	# cheats
# soundtrack	# soundtrack	# walkthrough	# walkthrough
# lyrics	movie #	# cheat codes	# cheat codes
# trailer	# trailer	# demo	# download
# games	# dvd	# download	# demo
# cast	# cast	# game	# game
# game	# the movie	cheats for #	cheats for #
movie #	# imdb	# patch	# patch
# dvd	# film	download #	download #
# the movie	watch #	# soundtrack	play #
Book		**Music**	
QueDet	Our Method	QueDet	Our Method
# movie	# summary	# lyrics	# lyrics
# summary	# book	lyrics to #	lyrics to #
# lyrics	# quotes	lyrics #	lyrics #
# book	summary of #	# song	# song
# quotes	# sparknotes	song #	song #
summary of #	# spark notes	lyrics for #	lyrics for #
# spark notes	# novel	# movie	# video
# sparknotes	sparknotes #	# chords	# chords
movie #	# cliff notes	# video	# tabs
# dvd	# author	# tabs	# tab

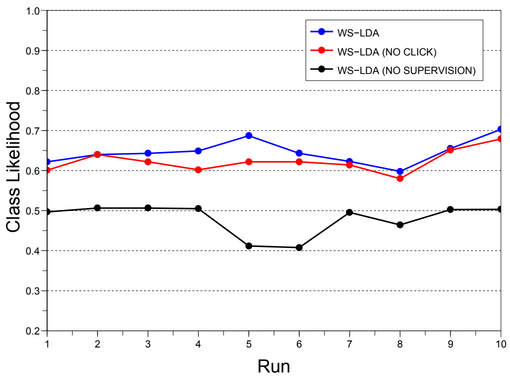

Figure 4: Comparison between Different Settings.

discovering the named entities in given classes. Our method is unique in that it uses click-through data as data source, it employs a topic model as model, and it conducts weakly supervised learning to create the topic model. Experimental results on large scale data shows that our method is very effective for named entity mining. Previous work on this problem resorted to query log data and a deterministic approach. Our method outperforms the existing method because click-through data is richer than query log data and the probabilistic approach is more robust against ambiguities than the deterministic approach. Our proposed learning method called WS-LDA (Weakly Supervised Latent Dirichlet Allocation) is an extension of the existing LDA method. It is a general method and can be used in other multi-label classification learning tasks using partially labeled data.

As future work, we plan to apply the proposed method to applications such as web search; we also want to further investigate the statistical properties of the WS-LDA method.

9. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais. Improving web search ranking by incorporating user behavior information. In Prof. of SIGIR'06, pp. 19-26, 2006.

[2] D. M. Bikel, S. Miller, R. Schwartz and R. Weischedel. Nymble: a high-performance learning name-finder. In Proc. of the 5th conference on applied natural language processing, pp.194–201, 1997.

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

[4] D. Blei and J. McAuliffe. Supervised topic models. In Advances in Neural Information Processing Systems 21 (NIPS'07), MIT Press, 2007.

[5] A. E. Borthwick. A maximum entropy approach to named entity recognition. PhD thesis, New York, NY, USA, 1999.

Table 6: Top Ranked Websites for Each Class.

Movie		Game	
QueDet-URL	Our Method	QueDet-URL	Our Method
www.imdb.com	www.imdb.com	www.gamespot.com	www.gamespot.com
us.imdb.com	us.imdb.com	cheats.ign.com	cheats.ign.com
imdb.com	imdb.com	www.gamefaqs.com	www.gamefaqs.com
movies.yahoo.com	movies.yahoo.com	www.download.com	www.download.com
www.gamespot.com	www.rottentomatoes.com	www.1up.com	faqs.ign.com
www.rottentomatoes.com	disney.go.com	faqs.ign.com	www.1up.com
www.movieweb.com	search.ebay.com	www.imdb.com	www.cheatscodesguides.com
disney.go.com	www.myspace.com	www.gamerankings.com	www.gamerankings.com
movies.ign.com	www.movieweb.com	xbox360.ign.com	www.cheatcc.com
www.apple.com	www.moviefone.com	www.cheatscodesguides.com	ps2.ign.com
Book		**Music**	
QueDet-URL	Our Method	QueDet-URL	Our Method
www.imdb.com	www.sparknotes.com	www.metrolyrics.com	www.metrolyrics.com
us.imdb.com	www.bookrags.com	www.imdb.com	www.lyrics007.com
www.bookrags.com	search.barnesandnoble.com	www.lyrics007.com	www.azlyrics.com
www.sparknotes.com	www.online-literature.com	www.azlyrics.com	www.lyricsfreak.com
imdb.com	profile.myspace.com	www.lyricsfreak.com	www.lyricsdepot.com
search.barnesandnoble.com	www.gradesaver.com	www.lyricsdepot.com	www.stlyrics.com
www.gradesaver.com	www.enotes.com	www.last.fm	music.yahoo.com
www.myspace.com	www.myspace.com	www.stlyrics.com	www.cowboylyrics.com
www.enotes.com	www.cliffsnotes.com	www.songfacts.com	www.sing365.com
www.online-literature.com	www.gutenberg.org	music.yahoo.com	www.mtv.com

[6] H. Cao, D. Jiang, J. Pei, Q. He, Z. Liao, E. Chen and H. Li. Context-aware query suggestion by mining click-through and session data, In Proc. ACM SIG KDD'08, pp.875–883, 2008.

[7] H. Cui, J. Wen, J. Nie, W. Ma. Probabilistic query expansion using query logs. In Proc. of WWW'02, pp.325-332, 2002.

[8] L. Dietz, S. Bickel and T. Scheffer. Unsupervised prediction of citation influences. In Proc. of ICML'07, pp.233–240, 2007.

[9] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: an experimental study. Artificial Intelligence, 165:91–134, June 2005.

[10] R. Gaizauskas, K. Humphreys, H. Cunningham, and Y. Wilks. University of sheffield: description of the lasie system as used for muc-6. In Proc. of MUC6'95: Proceedings of the 6th conference on Message understanding, pp.207–220, 1995.

[11] T. Hofmann. Probabilistic latent semantic indexing. In Proc. of SIGIR'99, pp.50–57, 1999.

[12] C. K. Huang, L. Chien, Y. Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. J. Am. Soc. Inf. Sci. Technol., 54(7):638–649, 2003.

[13] T. Joachims. Optimizing search engines using clickthrough data. In Proc. of KDD'02, pp. 133-142, 2002.

[14] R. Jones, B. Rey, O. Madani, W. Greiner. Generating query substitutions. In Proc. of WWW'06, 2006.

[15] B. Liu, Y. Dai, X. Li, W. S. Lee and P. Yu. Building Text Classifiers Using Positive and Unlabeled Examples. In Proc. of ICDM'03, 2003.

[16] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Proc. of HLT-NAACL 2003, pp.188–191, 2003.

[17] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In Proc. of WWW'08, pp.101–110, 2008.

[18] M. Paşca. Organizing and searching the world wide web of facts ? step two: harnessing the wisdom of the crowds. In Proc. of WWW'07, pp.101–110, 2007.

[19] M. Paşca. Weakly-supervised discovery of named entities using web search queries. In Proc. of CIKM '07, pages 683–690, 2007.

[20] I. Sato and H. Nakagawa. Knowledge discovery of multiple-topic document using parametric mixture model with dirichlet prior. In Proc. of ACM KDD'07, pp.590–598, 2007.

[21] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In Advances in Neural Information Processing Systems 15 (NIPS'03), pp.721–728. MIT Press, 2003.

[22] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In Proc. of SIGIR'06, pp.178–185, 2006.

[23] H. Yu, J. Han and K.Chang: PEBL: Positive Example Based Learning for Web Page Classification Using SVM. In Proc. of KDD'02, 2002.