Named Entity Disambiguation in User Queries Using Semi-Supervised Topic Modeling and Context

Jian Tang[1], Junwu Du[2], Jun Yan[3], Weiguo Fan[4], and Ming Zhang[1]

[1]School of EECS, Peking University, {tangjian, mzhang}@net.pku.edu.cn

[2]Beijing Institute of Technology, Beijing, China, du.junwu@gmail.com

[3]Microsoft Research Asia, Beijing, China, junyan@microsoft.com

[4]Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, wfan@vt.edu

ABSTRACT

The problem of Named Entity Recognition in user Query (NERQ) aims to extract the entity in the query and meanwhile classify it into the entity classes for disambiguation. It is critical in many online applications, such as Web relevance search, query suggestion and online advertising. The disambiguation of the entity in the query is challenging since (1) besides several existing target entity classes, there exist a large number of other unknown classes which have overlap with the existing ones; (2) due to the limited information contained in a single query, it is difficult to determine the entity class without other contextual information. In this paper, we propose a *Semi-Supervised Probabilistic Latent Semantic Analysis (SS-PLSA)* model to disambiguate the entities across the existing classes and against the other unknown classes simultaneously by leveraging some partially labeled and a large number of unlabeled entities. In online prediction, we further incorporate the search session context to reduce the entity ambiguities. Experiments with a commercial search engine query log demonstrate that the *SS-PLSA* model outperforms the existing NERQ approaches significantly. Moreover, the performance of the online prediction phrase is further improved by the incorporation of additional search session context information.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Query formulation.*

General Terms

Algorithms, Performance, Experimentation

Keywords

Named entity recognition in query, search session context, semi-supervised topic modeling

1. INTRODUCTION

The problem of Named Entity Recognition in user Query (NERQ) is attracting increasing attentions in both industry and academic communities recently. It is critical in many online applications, such as Web relevance search, query suggestion and online advertising [12]. According to a statistic over the user queries in a commercial search engine, 71% of the search queries contain at least one named entity [12].

Generally, the named entities in user queries are ambiguous across the entity classes, i.e. an entity may be associated with multiple entity classes. For example, the entity "harry potter" may belong to the entity class "Book", "Game" or "Movie". Thus in order to conduct NERQ, we not only have to extract the named entity in a user query but also classify it into the right entity class, which may belong to existing target entity classes or other unknown ones. Take the query "harry potter book price" as an example, the entity in the query is "harry potter" and the user is talking about the "harry potter" book, which is implied by the query context "book price".

However, the classification of the named entities in user queries is challenging, which arises from two major reasons. *First*, besides several existing target entity classes we are interested in, there exist large amounts of other unknown classes which also share a lot of entities with the existing ones. Thus we not only need to disambiguate the entities across the existing classes but also against the unknown ones. *Second*, in online prediction, due to the limited information in a single query, it is difficult, if not impossible, to determine the entity class solely based on the query itself. Take the query "harry potter review" as an example, even human editors cannot tell whether this is about the book or the movie without additional contextual information.

In this paper, we seek to address the above two challenges respectively. For the first challenge, we proposed a semi-supervised topic modeling approach called *Semi-Supervised Probabilistic Latent Semantic Analysis (SS-PLSA)* to disambiguate the entities across the existing entity classes and against the other unknown classes simultaneously. Specifically, each entity is treated as a document with its words as all the query contexts in the query log, and the entity classes are represented as topics of the model. The training data includes some partially labeled seed entities and a large number of unlabeled entities. The partially labeled entities all belong to the existing classes, which will be used to model these classes. The unlabeled entities may belong to the existing classes or other unknown ones. With the training data, the SS-PLSA model discovers the topics that are aligned with the existing classes and meanwhile discovers the topics corresponding to the unknown classes. These topics can be used to disambiguate the entities across the existing classes and against the other unknown ones. For the second challenge, as user search session context has been proved to be helpful in many query understanding tasks such as query classification [6] and

Conference'12, Month 1–2, 2012, City, State, Country.

query suggestion [7], we incorporated the context to further reduce the entity ambiguities. We performed extensive experiments on a commercial search engine query log. Experimental results show that our proposed *SS-PLSA* model significantly outperforms the baselines. Moreover, after incorporating the search session context, the performance is further improved.

The rest of this paper is organized as follows. In Section 2, we discuss the related work of this paper. Section 3 formulates the problem of NERQ. We describe our methodology in Section 4, where we present our semi-supervised topic modeling approach for offline training and context-aware NERQ for online prediction. Section 5 presents the experiment results. Finally, we conclude this paper and point out some future directions in Section 6.

2. RELATED WORK

There are three categories of previous works that have strong correlation to this paper. In terms of problem to be solved, the classical Named Entity Recognition (NER) [19] algorithms are related. In terms of solution for problem solving, the topic model [2] related studies are related. Finally, in terms of data to be used in this paper, the query log mining works [22, 23, 6, 7, 29] are strongly related. In this section, we briefly review the related studies along these three directions respectively.

The problem of Named Entity Recognition has been widely studied in the field of text mining [19]. However, traditional named entity recognition techniques [1, 5, 18, 27,9, 24, 11] such as the commonly used sequence labeling approach Hidden Markov Models [1], Conditional Random Fields [18], which are both mainly performed on natural language texts, are not applicable for NERQ since user queries are generally short and not well formed. The features used for traditional NER are not available in user queries such as, letter case, punctuation, etc., and hence the performance would not be good enough. Recently, there are a few algorithms [12, 10] that have been proposed for the NERQ problem. In [12], the authors first proposed the NERQ problem and proposed a Weakly Supervised Latent Dirichlet Allocation (WS-LDA) model to disambiguate the entities across the existing classes. However, they do not disambiguate the entities in existing classes against the other unknown classes. In contrast to the work of Du et al. [10], though the authors have considered leveraging the session context, they do not consider disambiguate across multiple existing classes, and they also do not incorporate unlabeled data for model training.

The search engine query log mining has been extensively studied for various application scenarios such as named entity and attribute mining [20, 21, 26, 14, 22, 23], query classification [6], query suggestion [7] etc. Among the log mining related works, the search session context has been leveraged and proved to work well in various applications such as query classification [6], query suggestion [7], search results ranking [29], and user intent understanding [28], etc. For example, in [6] the authors incorporated session context information for query classification by using conditional random fields (CRF) [15]. In [7], Cao et al. proposed a general search session context-aware model for query suggestion and ranking. All these related works are showing the power of search contexts for disambiguating Web search queries. In this work, we leverage the session context for named entity recognition in query.

Topic models such as Probabilistic Latent Semantic Analysis (PLSA) [13] and Latent Dirichlet Allocation (LDA) [2], have been well applied in various applications such as latent topic discovery [21, 25], community discovery [26], link modeling [17], opinion mining [32] and social annotation [17] etc. In [32], Lu et al. also proposed a Semi-supervised Probabilistic Latent Semantic Analysis (SS-PLSA) model to learn some aspects aligned with the existing ones and other extra aspects. D-LDA [31], is an algorithm that also strongly correlates to our SS-PLSA model. Given labeled data belonging to the existing classes and unlabeled data, D-LDA aims to identify the instances belonging to the known classes and meanwhile cluster the remaining data into other meaningful groups. For both of the SS-PLSA [32] and D-LDA [31] models, the instances in the labeled training data only belong to a single class (or aspect, topic), while the instances in our labeled training data may belong to multiple classes. This requires our model to have better capability in handling ambiguities.

3. PROBLEM FORMULATION

The NERQ problem aims to extract the named entities from the user search queries and meanwhile classify these entities into the entity classes, which may belong to the existing classes $\mathcal{K}$ or other unknown classes $\mathcal{U}$, for disambiguation. To simplify the problem explanation, in [12], the authors propose to recognize at most one entity from each query. Without loss of generality, we follow the same way to configure the problem. Mathematically, for each query q, a triple (e, t, c) is used to represent it, where e stands for the entity in this query q, t stands for the query context of the entity, i.e. the remaining texts in query q after removing e from it, and c stands for the class of entity e. As an example, the query "harry potter book price" could be represented by the triple ("harry potter", "# book price", book). Same as in [12], the NERQ problem could be formulated as finding the optimum triple $(e,t,c)^*$ in a given query q such that that the joint probability $p(e,t,c)$ among all possible entities, query contexts and classes is maximized, i.e.

$$(e,t,c)^* = argmax_{(e,t,c)\epsilon G_{(q)}} p(e,t,c) \tag{1}$$

where $G_{(q)}$ is the set of all possible triples that are possible to generate query q. Notice the fact that the joint probability $p(e,t,c)$ can be factorized into:

$$\begin{aligned} p(e,t,c) &= p(e)p(c|e)p(t|e,c) \\ &= p(e)\,p(c|e)\,p(t|c) \end{aligned} \tag{2}$$

where $p(e)$ is generally the popularity of named entity e in the query log, $p(c|e)$ is the probability of e belonging to class c, and $p(t|c)$ indicates the probability of query context t belonging to entity class c. In Equation (2), we write $p(t|e,c) = p(t|c)$ because we generally assume that the context of entity only depends on entity classes and not on an individual entity.

The NERQ problem is generally considered in two different stages, which are offline training stage and online prediction stage [12]. In offline training stage, we aim to estimate the probabilities $p(e), p(c|e)$ and $p(t|c)$. Among the three probability functions to be estimated, $p(e)$ is relatively easy to be estimated by utilizing the *popularity* of e, which is defined as the number of queries containing e in the search engine query log. Thus in offline training stage, we mainly introduce how to estimate the two latter probability functions. In terms of online prediction, for each query, we aim to find the optimum triple $(e,t,c)^*$ by comparing the joint probability $p(e,t,c)$, which is calculated based on the offline training results, among all the possible triples.

4. METHODOLOGY

The problem of entity disambiguation in queries suffers from two major challenges. First, we not only have to disambiguate the entities across the existing classes but also against other unknown ones. Second, user queries are generally short (2-3 words) and hence do not contain sufficient contextual information to determine the class of the entities they contain. For addressing the first challenge, we cast the offline training of NERQ as a *semi-supervised topic modeling* problem and proposed a *Semi-supervised Probabilistic Latent Semantic Analysis (SS-PLSA)* model to disambiguate the entities across the existing classes and against the other unknown classes simultaneously. For the second challenge, we utilize the search session context of the query to reduce the entity ambiguities in online prediction.

4.1 Semi-Supervised Topic Modeling in Offline Training

In offline training, we aim to estimate the two probabilities $p(c|e)$ and $p(t|c)$. As mentioned in [12], the estimation of the two probabilities can be formulated as a topic modeling problem. Each entity is treated as a document with its words as all the query contexts in the query log. For example, the named entity "harry potter" corresponds to the document **t**=("# book", "# walkthrough", "# trailer"). And entity classes are treated as the hidden topics. Thus *the probability $p(c|e)$ corresponds to the document topic distribution, and $p(t|c)$ corresponds to topics over query context vocabulary.* In order to disambiguate the entities across the existing entity classes and against the other unknown classes simultaneously, the topic modeling problem here aims to achieve two goals: (1) discovering the topic aligned with each existing entity class; (2) discovering the unknown topics, each of which corresponds to an unknown class and composes the unknown classes $\mathcal{U}$. The two kinds of topics can be used to model the existing entity classes and other unknown classes respectively and hence used to disambiguate across these classes. The number of unknown classes $\mathcal{U}$ is determined by the topic number, which is a predefined parameter in the topic modeling problem.

Formally, the offline training is defined as follows: given some partially labeled seed entities $\{e_i\}_{i=1}^{n}$ belonging to the existing classes $\mathcal{K}$ and a large number of unlabeled entities $\{e_i\}_{i=n+1}^{m}$, which may belong to $\mathcal{K}$ or other unknown classes $\mathcal{U}$, the offline training of NERQ aims to discover the topics $p(t|c_k)$ that are aligned with the classes in $\mathcal{K}$ and meanwhile discover the topics $p(t|c_u)$, each of which corresponds to an unknown class in $\mathcal{U}$ and does not belong to $\mathcal{K}$. Meanwhile, the probability $p(c|e)$ of each entity *e* belonging to each class *c* can be estimated based on the discovered topics. We summarize the offline training of NERQ as a *Semi-Supervised Topic Modeling* (SSTM) problem (Figure 1).

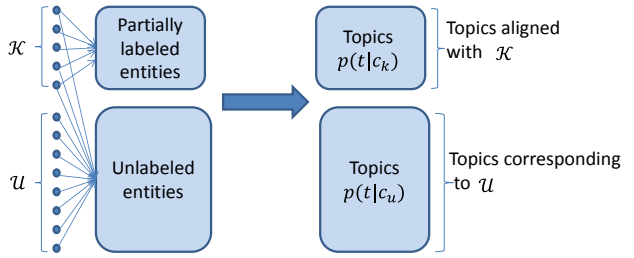

Figure 1: A description of the *Semi-Supervised Topic Modeling (SSTM)* problem. The discovered topics are separated into two parts: some topics are aligned with the existing target classes $\mathcal{K}$; the other topics correspond to the other unknown classes $\mathcal{U}$.

Thus for the SSTM problem, the training data includes two parts: (1) some partially labeled documents $D_l = \{d_1, d_2, \dots, d_n\}$, which belong to the existing classes $\mathcal{K}$ and each of which may be associated with multiple classes in $\mathcal{K}$. The assigned class labels of document d_i are represented with a $|\mathcal{K}|$ -dimensional vector $y_i = \{y_{i1}, y_{i2}, .., y_{i|\mathcal{K}|}\}$, where y_{ij}takes 1 when d_i belongs to class j in $\mathcal{K}$ and 0 otherwise. $|\mathcal{K}|$ is the number of total existing classes. (2) a large number of unlabeled documents $D_u = \{d_{n+1}, d_{n+2}, \dots, d_m\}$, which may belong to the existing classes $\mathcal{K}$ or other unknown classes $\mathcal{U}$.

In order to discover the topics aligned with $\mathcal{K}$, as similar in [12], we incorporate the following constraint on the partially labeled documents:

$$\mathrm{C}(\mathbf{y}, \Theta) = \sum_{d=1}^{n} \sum_{z=1}^{|\mathcal{K}|} y_{dz} \log p(z|d) \,^{1} \tag{6}$$

By maximizing the above constraint, we can achieve two goals: (1) the i^{th} topic is aligned to the i^{th} predefined class; and (2) the labeled document is mainly distributed over its labeled classes.

By jointly maximizing the log-likelihood of the partially labeled entities and the constraint, we can discover the topics aligned with $\mathcal{K}$and hence disambiguate the entities across $\mathcal{K}$ [12]. In order to further disambiguate the entities in $\mathcal{K}$against the other unknown classes $\mathcal{U}$, we incorporate the unlabeled entities, from which we can discover the unknown entity classes and help disambiguate these classes against the existing ones. Thus we aim to maximize the log-likelihood of the partially labeled documents, constraint (6) and the log-likelihood of the unlabeled documents simultaneously. Specifically, the final objective function is defined as a linear combination of the three terms:

$$\mathcal{O} = (1 - \alpha - \beta)\mathcal{L}(D_l, \Theta) + \alpha \mathrm{C}(\mathbf{y}, \Theta) + \beta \mathcal{L}(D_u, \Theta) \tag{7}$$

where $\mathcal{L}(D_l, \Theta)$ is the log-likelihood of the partially labeled documents and calculated by:

$$\mathcal{L}(D_l, \Theta) = \sum_{d=1}^{n} \sum_{w} n(d, w) \log \sum_{z} p(z|d) p(w|z) \tag{8}$$

$\mathcal{L}(\mathrm{D_u}, \Theta)$ is the log-likelihood of the unlabeled documents and calculated by:

$$\mathcal{L}(D_u, \Theta) = \sum_{d=n+1}^{m} \sum_{w} n(d, w) \log \sum_{z} p(z|d) p(w|z) \tag{9}$$

α and β are two parameters used to control the supervision of partially labeled documents and unlabeled documents respectively.

[1] A more intuitive constraint should be $\mathrm{C}(\mathbf{y}, \Theta) = \sum_{d=1}^{n} \sum_{z=1}^{|\mathcal{K}|} y_{dz} p(z|d)$, and the reason that we use $\log p(z|d)$ instead of $p(z|d)$ here is for computation simplicity in the final EM updating procedure. In the experiment, we prove this constraint can also achieve the same effects.

For the optimization algorithm of objective function (7), we also adopt the EM algorithm. We omit the detailed derivation procedure and only present the final update equation as below:

- **E-step:** computing the posterior of the latent variable $p(z|d,w)$,

$$p(z|d,w) = \frac{p(z|d)p(w|z)}{\sum_{z'} p(z'|d)p(w|z')} \tag{10}$$

- **M-step:** updating the model parameters based on the results of E-step,

  For partially labeled documents $1 \le d \le n$,

$$p(z|d) \propto (1-\alpha-\beta)\sum_{w} n(d,w)p(z|d,w) + \alpha y_{dz}$$

  For unlabeled documents $n+1 \le d \le m$,

$$p(z|d) \propto (1-\alpha-\beta)\sum_{w} n(d,w)p(z|d,w)$$

$$p(w|z) \propto (1-\alpha-\beta)\sum_{d=1}^{n} n(d,w)p(z|d,w) + \beta \sum_{d=n+1}^{m} n(d,w)p(z|d,w) \tag{11}$$

Prediction: The SS-PLSA is also used for predicting the probability $p(c|e)$ for unseen named entity *e*. This corresponds to estimate the entity associated document topic proportion $p(z|d)$. The estimation is the same as conventional prediction in the standard PLSA, where in E-step we update $p(z|d,w)$ using Equation (10) and in M-step we update $p(z|d)$ using Equation (11) while holding the value of $p(w|z)$.

4.2 Context-aware NERQ in Online Prediction

In online prediction stage, we aim to detect the named entities in queries and classify the entities into either the predefined entity classes $\mathcal{K}$ or other unknown classes $\mathcal{U}$. However, since the queries are generally short, sometimes it is difficult or even impossible to determine the entity classes solely based on a few terms in the queries. For example, it is hard to judge the class of entity "harry potter" in query "harry potter review" since the entity could belong to the class of movie or book or others.

Fortunately, many search engine users generally search different aspects of the same entity in a user search session [23]. Thus we propose to consider the sequential user search behaviors in a user search session for named entity recognition and entity class disambiguation. For example, in the first example of Table 1, it is difficult to determine the class of the entity "harry potter" based on the current query "harry potter review". However, according to the previous query "harry potter book", we can know that the user is searching the book of "harry potter". Similarly, in the second example of Table 1, we can determine the user is searching the movie of "harry potter". From the two examples we can see that user search session context can help reduce the entity ambiguities.

Table 1: Examples of Search Session Query Sequences

ID	User Session
1	"harry potter book" → "harry potter review"
2	" movie" → "harry potter "
3	"apple computer" → "apple"

Formally, given a user search session $q_1 q_2 \dots q_{T-1} q_T$, we recognize the named entity from the current query q_T by utilizing the search session context. In this paper, for simplicity we only consider q_{T-1} as the session context. Specifically, we jointly recognize the named entities in q_{T-1} and q_T. Then the query context in q_{T-1} can be used to help determine the entity class in q_T. For example, in the sequence "harry potter book price" → "harry potter review", the query context of previous query "# book price" can help determine the entity class of current query, which is talking about the "harry potter" book.

In order to jointly recognize the named entities in the previous and current query and meanwhile utilize the previous query context to help determine the entity class of current query, we aim to maximize the following objective function:

$$\Lambda = p_{T-1}(e_{T-1}, t_{T-1}, c_{T-1}) + p_T(e_{T-1}, e_T, t_{T-1}, t_T, c_T) \tag{12}$$

where (e_T, t_T, c_T) is the possible triple generated by query q_T, $p_{T-1}(e_{T-1}, t_{T-1}, c_{T-1})$ is defined as in Equation (2), and $p_T(e_{T-1}, e_T, t_{T-1}, t_T, c_T)$ is defined as:

$$p_{T-1}(e_{T-1}, e_T, t_{T-1}, t_T, c_T) = p(e_T)\, p(c_T|e_T) \left(p(t_T|c_T) + f(e_{T-1}, e_T) p(t_{T-1}|c_T)\right)$$

where

$$f(e_{T-1}, e_T) = \begin{cases} 1, & if\ e_{T-1} = e_T \\ 0.5, & \text{otherwise} \end{cases}$$

The function $f(e_{T-1}, e_T)$ is used to determine the weight of the previous query context in influencing the current entity class. If the entities in previous and current query are the same, the weight should be larger.

The procedures of finding the optimum triples $(e_{T-1}, t_{T-1}, c_{T-1})^*$ and $(e_T, t_T, c_T)^*$, which maximizes the objective function (12) are separated into two steps: (1) First, we search all possible triples $(e_{T-1}, t_{T-1}, c_{T-1})$ that maximizes the first term $p_{T-1}(e_{T-1}, t_{T-1}, c_{T-1})$. This can be finished by iterating each substring of q_{T-1} as e_{T-1}, the rest part of q_{T-1} as t_{T-1} and each entity class as c_{T-1}. Then the triple with the highest probability is outputted as $(e_{T-1}, t_{T-1}, c_{T-1})^*$. The time complexity of this part is $O(Kn^2)$, where *K* is the number of topics in SS-PLSA, and *n* is the length of query q_{T-1}. Both K and n are quite small, so this part can be finished immediately; (2) Second, after finding ${t_{T-1}}^*$, we optimize the second term $p_T(e_{T-1}, e_T, e_T, {t_{T-1}}^*, t_T, c_T)$, which can be done similarly as done in the first part by finding all the possible (e_T, t_T, c_T). The time complexity is also the same as the first part. Thus the total time complexity is also $O(Kn^2)$, and the prediction can be efficiently finished as user queries are quite short.

5. EXPERIMENTS

In this section, we introduce our empirical study for verifying our proposed algorithm in dealing with the NERQ problem. We first introduce the datasets and then describe our evaluation metrics for offline training and online prediction respectively. For the baselines, in offline training we compare our proposed SS-PLSA model with WS-LDA [12], and the Determine approach used in [12], and in online prediction we compare the performances of these methods with or without search session context.

5.1 Datasets

We use a three-month search query log of a commercial search engine. Queries that are non-English are all removed and finally the total number of queries that we get is about 10 billion and the number of unique ones is about 1.5 billion. As for the existing target entity classes, we study the same ones as in [12], which are "Book", "Game", "Movie" and "Music", and we use the same seed named entities in [12], which contain 120 unique named entities belonging to the four entity classes. Each of the named entities may belong to multiple classes of the four classes. The dataset has two characteristics: (1) the overlap ratios between classes vary according to class pairs. For example, the overlap ratios of the "Movie" and "Book" class pair and the "Movie" and "Game" class pair are higher than 20%. (2) The number of the named entities in each class is proportional to the size of their search traffic in the query log. For example, the number of named entities belonging to "Game" or class "Movie" is larger than that belonging to "Book" or "Music". Note we do not contain any named entities that from other unknown classes among the seed named entities.

For the test named entities, which are used for the offline training evaluation (introduced in the next subsection), we randomly choose 30 named entities that belong to other unknown classes (all are labeled with the "Others" class) besides the 60 named entities used as test entities in [12], which all belong to the four target classes. The detailed statistics of the training and testing named entities are summarized in Table 2.

Table 2: Statistics of the labeled entities for training and testing

	Book	Game	Movie	Music	Others
Training	34	37	59	30	0
Testing	11	15	23	11	30

Starting with the 120 seed named entities, we scan the search query log three times to collect our training data. First, we scan the query log the first time and collect the queries that contain the named entities. In each query, the remaining text after the entity is removed and treated as query context. Then each entity is represented with their query contexts. Second, we scan the query log again and collect the queries that contain the query contexts collected in the first step. In each query, the remaining text after the query context is removed is treated as an entity. In this step, we collect the unlabeled entities, which may belong to the existing classes or other unknown classes. Third, we scan the query log the third time and collect the queries containing the unlabeled entities. We remove the unlabeled entities in the queries and get the query contexts. By this step, all the unlabeled entities are represented with their query contexts. We remove the query contexts that appear in less than five entities and the entities with document size less than five. Finally, we get 32,234 contexts and about 1 million unlabeled named entities.

For online prediction, we use another month of query log for evaluation. As done in the previous work [6], queries appear within 30 minutes are segmented into a user session. We perform NERQ on the user search sessions, in which the last query is considered as target query. Then we randomly sampled 1000 user queries from the recognition results for evaluation. Each recognition result was manually labeled as correct and incorrect. A result is treated as correct if and only if the detected entity and its associated class are both correct. We summarize the number of the named entities in each entity class among the 1000 queries in Table 3.

Table 3: Statistics of the test queries for online NERQ

	Book	Game	Movie	Music	Others
Number	187	293	343	75	102

5.2 Evaluation Metrics

The evaluation is done for offline training and online entity recognition respectively. In offline training, we evaluate the accuracy of estimated probabilities $p(c|e)$for the unlabeled named entities and test entities, and in online prediction we measure the accuracy of the online entity recognition result.

Offline Evaluation

We use two metrics, which are the same as [12], to evaluate the accuracy of the probability $p(c|e)$ for the unlabeled entities and test entities respectively. The first one is P@N, which is firstly to rank the unlabeled entities for each target class c according to $p(c|e)$ and then calculate the precision of the top N entities, i.e.

$$\text{P@N} = \frac{\#entities\ belonging\ to\ class\ c}{N}$$

The second one is the Average Class Likelihood, which is done on the test entities and calculated by:

$$\text{Average Class Likelihood} = \frac{1}{m}\sum_{e=1}^{m}\sum_{c} p(c|e)y_{ec}$$

where m is the total number of test named entities, and y_{ec} equals to 1 if entity e belongs to class c otherwise 0. The average class likelihood measures how consistent the algorithm predictions are with human labels and hence the larger the better.

Online Prediction

Given a user search session $q_1 q_2 \ldots q_T$, we take the last query q_T as the test query and the previous query q_{T-1} as its search context for simplicity. To compare the performances of the comparative algorithms, we use the commonly used metric F1 in information retrieval field. We calculate the F1 value for each entity class and the overall F1 value by averaging over all the entity classes.

5.3 Comparative Algorithms

We compare the offline training performances of the following algorithms:

(1) *Determine:* The determine approach was used as a baseline in [12], which learns the query contexts of each

existing entity class by simply aggregating all the contexts of the named entities that belong to the class.

(2) *WS-LDA:* The probabilistic approach proposed in [12], which learns the contexts of each class by using the partially labeled seed named entities in the existing target classes. As in [12], the parameter λ in the WS-LDA model is set to 1 by default.

(3) *SS-PLSA:* Our proposed semi-supervised topic modeling approach introduced in Section 4. It learns the contexts of each class with both partially labeled seed entities and unlabeled named entities. In offline training, we use 10,000 unlabeled entities for training by default, and the default values for α, β and the number of topics are set as 0.96, 0.01, 20 respectively.

For online prediction, besides the three algorithms above, we further compare the performances of these algorithms incorporating search session context, which are denoted by adding a "+ Context" after their names.

5.4 Experimental Results

In this subsection, we present the experimental results on NERQ. We first present the offline training results, followed by online prediction results. Finally, we give a sensitivity analysis with respect to the parameters in the SS-PLSA model.

5.4.1 Offline Evaluation

In offline evaluation, we compare the performances of the algorithms on estimating the two probabilities $p(t|c), p(c|e)$, i.e. the probability of query context *t* belonging to class *c* and the probability of entity *e* belonging to class *c*.

In Table 3, we compare the learned query contexts of the four existing classes with the three comparative algorithms. We list the top ten query contexts, which are ranked according to the probability $p(t|c)$ for each of our target class *c*. Due to the ambiguities of the seed named entities, the learned query contexts in each class by the *Determine* approach are often mixed with some query contexts in other existing classes. Take the "Book" class as an example, the query contexts "# torrent", "# movi" are ranked top in the "Book" class while the two contexts belong to the "Movie" class. This is because the entities from "Book" and "Movie" have a big overlap. We learned better query contexts with the *WS-LDA* and *SS-PLSA* models because topic model can better deal with entity ambiguities. Furthermore, by incorporating large number of unlabeled entities, *SS-PLSA* model discover better topics than *WS-LDA*. Take the "Game" class as an example, the query contexts "# download" and "# wiki" are ranked top using the *WS-LDA* model, which learns with the partially labeled seed entities belonging to the four classes. However, the two query contexts often more frequently appear in other unknown entity classes such as "Software" class. The problem lies in that *WS-LDA* learns the topics only with the entities belonging to the four existing classes. Our *SS-PLSA* model can effectively address this problem by learning with a large number of unlabeled entities which contain lots of entities that belong to other unknown entity classes and hence can help disambiguate the existing classes against the other unknown classes.

Table 3: Comparisons on learned query contexts of each target class. We list the top ten contexts of each target class ranked according to the probabilities $p(t|c)$.

Book			**Game**		
Determine	**WS-LDA**	**SS-PLSA**	**Determine**	**WS-LDA**	**SS-PLSA**
# book	# book	# book	# torrent	# cheat	# cheat
# quot	# quot	quot from #	# wiki	descargar #	descargar #
# torrent	# torrent	who wrote #	# game	# game	# pc
who wrote #	quot from #	summari of #	# pc	# download	# walkthrough
quot from #	who wrote #	# lesson plan	# cheat	# pc	# ps2
# review	# review	# summari	# download	# torrent	# ps3
# movi	# summari	# chapter summari	descargar #	# wiki	plai #
# summari	# chapter summari	the book #	# 2	# walkthrough	# onlin game
# author	# author	book #	# walkthrough	# patch	# cheat code
# chapter summari	summari of #	# pdf	# pictur	# ps3	game #
Movie			**Music**		
Determine	**WS-LDA**	**SS-PLSA**	**Determine**	**WS-LDA**	**SS-PLSA**
# torrent	# movie	# cast	# lyric	# lyric	# lyric
# movie	# torrent	# torrent	# song	# song	lyric to #
# movi	# cast	# soundtrack	# lyrics	# lyrics	# song
# cast	# soundtrack	# movie	lyric to #	lyric to #	lyric #
# soundtrack	cast of #	# imdb	lyric #	lyric #	# lyrics
# imdb	# movi	movi #	# youtub	letra de #	# chord
# quot	# imdb	cast of #	youtub #	# chord	song #
cast of #	# wiki	# quot	# chord	song #	letra de #
movi #	# dvd	# the movi	letra de #	# youtub	# music video
# wiki	# quot	# movi	you tube #	youtub #	letra #

Table 4: Precision of learned named entities in each class (P@N).

	Book			Game			Movie			Music			Average		
	Determine	WSLDA	SSPLSA	Determine	WSLDA	SSPLSA	Determine	WSLDA	SSPLSA	Determine	WSLDA	SSPLSA	Determine	WSLDA	SSPLSA
P@25	0.88	0.92	0.96	0.68	0.76	0.86	1	1	1	1	1	1	0.89	0.92	**0.955**
P@50	0.84	0.9	0.95	0.66	0.8	0.93	0.96	0.984	1	0.984	1	1	0.861	0.921	**0.97**
P@100	0.78	0.886	0.945	0.69	0.85	0.93	0.96	0.97	0.995	0.97	0.99	1	0.85	0.924	**0.9675**
P@150	0.773	0.88	0.92	0.68	0.85	0.94	0.927	0.95	0.99	0.964	0.988	1	0.836	0.917	**0.9625**
P@250	0.726	0.85	0.89	0.636	0.88	0.94	0.882	0.928	0.988	0.96	0.984	1	0.801	0.9105	**0.9545**

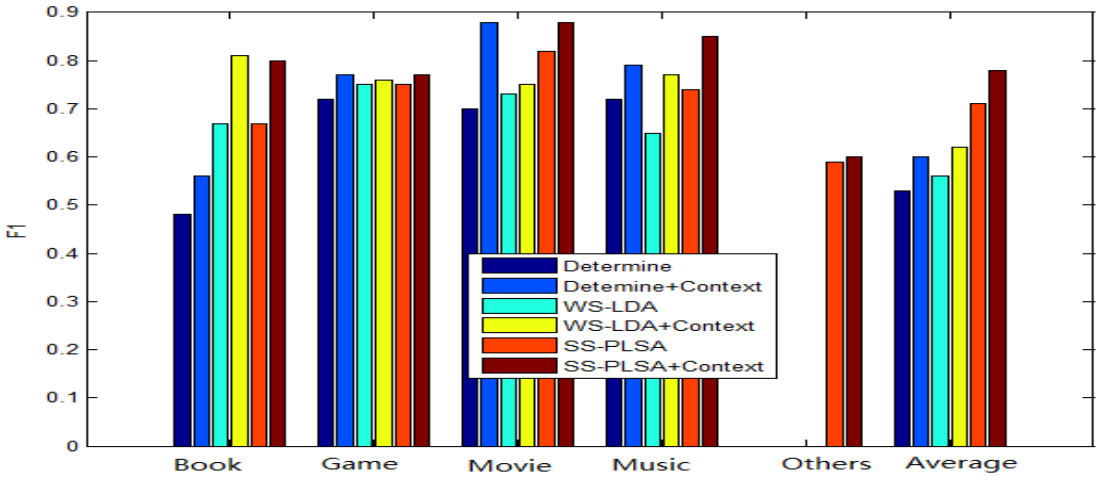

Figure 4: Online recognition results. (F1 measure)

In Table 4, we present the precision of the top named entities ranked according to the probabilities p(c|e) in each existing target classes. Overall, the topic model based approach performs consistently better than the *Determine* approach. Furthermore, *SS-PLSA* outperforms *WS-LDA* in all the domains due to the incorporation of unlabeled entities. The reasons are the same as stated in the above paragraph, which means that topic models better address the problem of entity ambiguities and incorporating unlabeled entities help disambiguate the entities in existing target classes against other unknown classes. Among the four target classes, the precision of "Music" class is the highest while "Game" class is the lowest. The reason is that the "Music" class has less overlap with other unknown classes, while the "Game" class shares a lot of query contexts with other unknown classes, among which the most similar one is "Software" class by our case study. For example, "# download", "descargar #" are frequently shared by the two classes.

We further compare the average class likelihood of the test named entities with the *WS-LDA* and *SS-PLSA* models. We repeat the experiments ten times and the result is presented in Figure 3. We can see that the average class likelihood with *SS-PLSA* is consistently better than the one with *WS-LDA*, which indicates that *SS-PLSA* performs better in estimating entity class probability p(c|e).

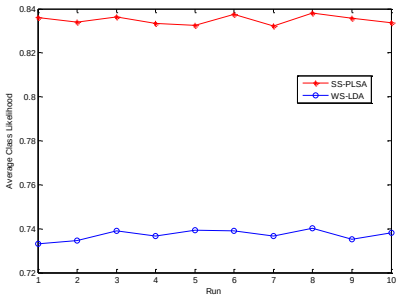

Figure 3: Comparisons of the average class likelihood of the test entities with *WS-LDA* and *SS-PLSA*.

5.4.2 Online Prediction

Figure 4 presents the online entity recognition results. We list the F1 value in each entity class and the overall average F1 value. Overall speaking, the *WS-LDA* and *SS-PLSA* approaches outperform the *Determine* approach and *SS-PLSA* performs the best. Our SS-PLSA model is effective in recognizing the named entities in each class while both the *Determine* and *WS-LDA* approaches fail in the "Others" class. The reason is that both the two approaches classify all the possible named entities into the four existing classes and hence all the entities from the "Others" class will be misclassified into one of the four classes. The F1 value of the "Others" class with the SS-PLSA model is lower than the four classes. This is because the entities belonging to the "Others" class in our training data is much larger than the four target classes and hence the training data will be biased towards the "Others" class, which result that a lot of entities from the target classes are misclassified into the "Others" class.

Besides, after incorporating the search session context, the performances of all three approaches further improved. As mentioned previously, this is because the search session context can resolve the entity ambiguities and hence improves the entity classification result.

5.4.3 Parameter Sensitivity Analysis

In this part, we give a sensitive analysis on the average class likelihood of the test named entities with respect to the parameters in the *SS-PLSA* model.

First, we vary the value of the parameter α, which is used to control the supervision from partially labeled seed entities. The result is presented in Figure 5. When the value of α equals to 0, which means we do not incorporate any supervised information, the class likelihood is quite low. As α increases, the class likelihood also increases. This indicates increasing supervised information can help predict class labels more accurately. However, when α increases to larger than 0.95, the class likelihood begin to decrease. This is because the supervised information is over emphasized.

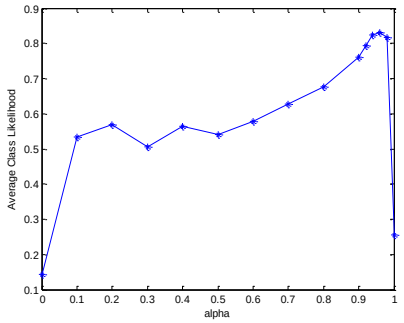

Figure 5: Sensitivity analysis with respect to the parameter α in the *SS-PLSA* model

Second, we investigate how the parameter β, which is used to control the influence of the unlabeled entities, affect the performances of the *SS-PLSA* model. The result is presented in Figure 6. From Figure 6, we can see that as the value of β increases from 0, the performance increases significantly, which proves that adding unlabeled entities for training can improve the performance. This is because adding unlabeled entities can help disambiguate the entities in the target classes from the other unknown classes. However, as we further increase β, the performance begins to drop. The reason is that increasing β will decrease the weight of supervised information, which has been proved quite helpful from Figure 5.

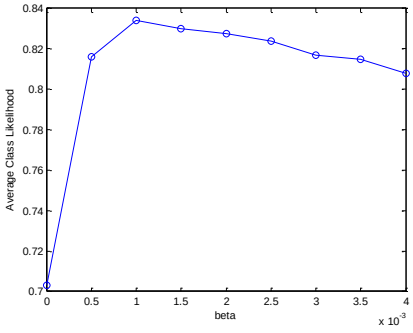

Figure 6: Sensitivity analysis with respect to the parameter β in the *SS-PLSA* model

Third, we investigate how the number of unlabeled entities for training influences the performance. We present the result in Figure 7. As we increase the number of unlabeled entities, the performance improves. This is because the unlabeled entities contain a large number of unlabeled entities from other unknown classes. These entities can be used to model the other unknown classes and help us disambiguate the entities in the target existing classes against the other unknown classes. As the number of entities in the other unknown classes is quite large, increasing the number of unlabeled entities will help model these classes better and hence improve the performance.

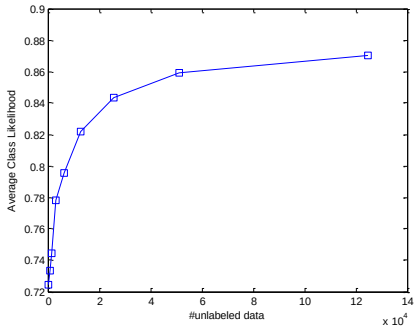

Figure 7: Sensitivity analysis with respect to the number of unlabeled entities for training in the *SS-PLSA* model

In Figure 8, we present the results of the average class likelihood with different number of topics in the *SS-PLSA* model. The number of topics determines the possible number of unknown entity classes in the unlabeled entities. When the number of topic is small, which may not match the real entity class number in the unlabeled entities, the performance is not good enough. As we increase the topic number, the performance also increases. However, when the topic number is larger than 30, the performance begin to stabilize. This suggests that using 30 topics is enough to model the entity classes in the unlabeled entities.

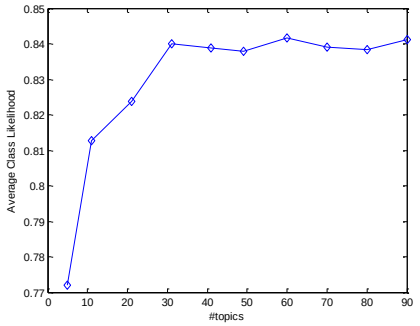

Figure 8: Sensitivity analysis with respect to the number of topic numbers in the *SS-PLSA* model

Finally, we summarize the experimental results presented in this Section. We have done the evaluation both for the offline training and online recognition with all the comparative algorithms. Experimental results show that the topic model based algorithms *WS-LDA* and *SS-PLSA* outperform the *Determine* approach both in offline training and online recognition, as the topic models can better resolve the entity ambiguities. Furthermore, *SS-PLSA* model outperforms the *WS-LDA* model due to the incorporation of large number of unlabeled entities, which can be used to model the distribution of the other unknown entity classes. For online recognition, incorporating search session

context of current query can further improve the performance as search session context can reduce the entity ambiguities.

6. CONCLUSION AND FUTURE WORK

This paper studied the problem of named entity recognition in user query. In offline training, we proposed a semi-supervised topic modeling approach SS-PLSA to disambiguate the entities across the existing classes and against the other unknown classes simultaneously. In online prediction, we incorporated the search session context to further reduce the entity ambiguities. Experimental results demonstrated the superiority of the SS-PLSA model over the baselines both in offline training and online prediction, and incorporating search session context further improved the online performance.

In the future, we plan to further explore the NERQ problem from three research directions. First, we plan to add more entity classes into the existing classes. We will add the classes that have big overlap with the existing ones and hence we can model these classes explicitly. This will help reduce the ambiguities between the existing classes and the other unknown ones. For example, in our offline training result, we found a lot of query contexts and entities in the "Software" class are mixed into the "Game" class. By explicitly modeling the "Software" class, the entities from "Software" or "Game" will be easily classified and hence help disambiguate the target classes from the other unknown ones. Second, we plan to model the similarities between entity classes. The degree of relatedness between entity classes is different. For example, the similarity between "Book" and "Movie" is higher than that between "Book" and "Game". Thus a "Book" entity is more likely also to be a "Movie" entity than to be a "Game" entity. Along this line, the Correlated Topic Model (CTM) [3] has already made some progress. Third, we plan to incorporate more user context information, such as user clicks, to further reduce the entity ambiguities.

7. REFERENCES

[1] D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. Nymble: a High-Performance Learning Name-finder. In Proc. Conference on Applied Natural Language Processing, 1997.

[2] D. Blei, A. Y. Ng and M. Jordan. Latent dirichlet allocation. In Journal of Machine Learning Research, 2003.

[3] D. Blei and J. Lafferty. Correlated Topic Models. In NIPS'06, 2006.

[4] S. Borman. The Expectation Maximization Algorithm: A Short Tutorial, 1997

[5] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. NYU: Description of the MENE Named Entity System as used in MUC-7. In Proc. Seventh Message Understanding Conference, 1998.

[6] H.Cao, D. H. Hu, D. Shen, D. Jiang, J. Sun, E. Chen, and Q. Yang. Context-aware query classification. In Proc. Of SIGIR'09, pp. 3-10, 2009.

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

[8] J. Chang and D. Blei. Relation topic models for document networks. Artificial Intelligence and Statistics, 2009.

[9] A. Cucchiarelli and P. Velardi. Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence. Computational Linguistics 27:1.123-131, 2001.

[10] J. Du, Z. Zhang, J. Yan, Y. Cui and Z. Chen. Using search session context for named entity recognition in user query. In Proc. Of SIGIR'10, pp. 765-766, 2010.

[11] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. Artificial Intelligence 165.91-134, 2005.

[12] J. Guo, G. Xu, and H. Li. Named entity recognition in user query. In Proc. Of SIGIR'09, pp. 267-274, 2009.

[13] T. Hofmann. Probabilistic latent semantic indexing. In Proc. Of SIGIR'1999, pp. 50-57, 1999.

[14] M. Komachi, S. Makimoto, K. Uchiumi, and M. Sassano. Learning semantic categories from click-through logs. In Proc. Of ACL'09, pp. 189-192, 2009.

[15] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields probabilistic models for segmenting and labeling sequence data. In ICML'01, pages 282-289, 2001.

[16] Y. Liu, A. Niculescu-Mizil, and W. Grys. Topic-Link LDA: joint models for topic and author community. In Proc. Of ICML'09, pp. 665-672, 2009.

[17] C. Lu, X. Hu, X. Chen, J. Park, T. He, and Z. Li. The topic-perspective model for social tagging systems. In Proc. Of KDD'10, pp. 683-692, 2010.

[18] A. McCallum and W. Li. Early Results for Named Entity Recognition with Conditional Random Fields, Features Induction and Web-Enhanced Lexicons. In Proc. Conference on Computational Natural Language Learning, 2003.

[19] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. Lingvisticae Investigationes, Vol. 30, No. 1. (January 2007), pp. 3-26.

[20] M. Pasca. Weakly-supervised discovery of named entities using web search queries. In Proc. Of CIKM'07, pp. 683-690, 2007.

[21] M. Pasca. Organizing and searching the world wide web of facts – step two: harnessing the wisdom of the crowds. In Proc. Of WWW '07, pages 101–110, 2007.

[22] M. Pasca and B. V. Durme. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In Proc. Of ACL'08, pp19-27, 2008.

[23] M. Pasca, E. Alfonseca, E. Robledo-Arnuncio, R. Martin-Brualla, and K. Hall. The role of query sessions in extracting instance attributes from web search queries. In Proc. Of ECIR'10, pp. 62-74, 2010.

[24] M. Pasca, D. Lin, J. Bigham, A. Lifchits, and A.Jain. Organizing and Searching the World Wide Web of Facts—Step One: The One-Million Fact Extraction Challenge. In Proc. Of AAAI'06, pp. 1400-1405, 2006.

[25] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina. Clustering the tagged web. In Proc. Of WSDM'09, pp. 54-63, 2009.

[26] S. Sekine and H. Suzuki. Acquiring ontological knowledge from query logs. In WWW '07, pages 1223–1224, 2007.

[27] B. Sergey. Extracting Patterns and Relations from the World Wide Web. In Proc. Workshop on the Web and Databases, pp. 172-183, 1998.

[28] Y. Shen, J. Yan, S. Yan, L. Ji, N. Liu, and Z. Chen. Sparse hidden-dynamics conditional random fields for user intent understanding. In Proc. Of WWW'11,pp. 7-16, 2011.

[29] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. In Proc. Of SIGIR'10, pp. 451-458, 2010.

[30] G. Xu, S. Yang and H. Li. Mining named entity using weakly supervised latent dirichlet allocation from click-through data. In Proc. Of KDD'09, pp. 1365-1374, 2009.

[31] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, and Z. Shi. D-LDA: a topic modeling approach without constraint generation for semi-defined classification. In Proc. Of ICDM'10, pp. 709-718, 2010.

[32] Y. Lu and C. Zhai. Opinion Integration Through Semi-supervised Topic Modeling, Proceedings of the World Wide Conference 2008 (WWW'08), pages 121-130.