Identifying and Classifying Named Entities from Click-through Data Effectively

Xuan Jiang[1] **Hongyan Liu**[2] **Hong Cheng**[3] **Jun He**[1] **Xiaoyong Du**[1]

[1]School of Information, Renmin University of China
[2]Department of Management Science and Engineering, Tsinghua University
[3]Dept. of Syst. Eng. & Eng. Mgmnt, The Chinese University of Hong Kong

Abstract

This paper addresses the problem how to identify named entities from click-through data and classify them into predefined domains accurately. By proposing a novel measurement to measure the importance of contexts[1], the method identifies and ranks named entities effectively. The probabilistic ranking model combines multi-information from click-through data such as queries, clicks and sessions. To improve the identification recall, the identification and ranking steps iterate in a bootstrapping manner. Experiments on a real data set show that the method is effective.

Keywords: Named Entity Recognition, Log Mining

1. Introduction

Named Entity Recognition and Classification (NERC) is an important subtask of Information Extraction, which is to identify named entities in unstructured text and classify them into certain domains, such as movies, books, etc. Since search engines become popular, click-through data which records users' searching behaviors has been studying intensively recently. According to [5], about 71% of users' queries contain named entities. Thus NERC from click-through data can be useful for many applications, such as query suggestion, relevance search, etc. For example, in relevance search, if a user issues a ambiguous query "baker job opening", where "baker" can either be a job type or the name of a university, traditional keyword-matching techniques do not work well, which can be seen from Figure 1. If we identifies "baker" and classifies it as a "job", we can offer users better results.

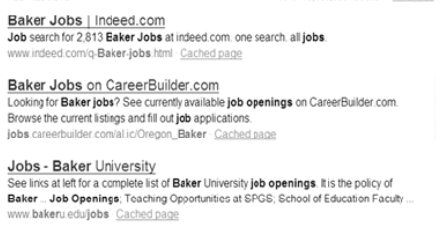

Fig. 1: Search results of query "baker job opening".

As queries are usually short and not in standard form, natural lan-

[1]the remaining terms of queries after removing the named entity

guage processing techniques based on grammar rules [10] or machine learning models [2] are not applicable to identify named entities from click-through data. However, by utilizing the observation of [3] that entities of one domain tend to share contexts, the goal can be accomplished. For example, in domain "job" , "# job opening"[2] can be the context of named entities like "driver", "teacher", etc. And the general process of the identification can be as follows. Given a certain domain and some seed entities, obtaining the associated contexts in the search log, using the contexts to cover candidate entities and ranking them by some measures.

The key point of the process is how to choose "important" contexts to identify candidate entities, because queries are quite ambiguous in expressing search intents [7]. Take the bipartite graph in Figure 2 for an example, each edge represents the query that contains the associated entity and context, and the weight of the edge represents the number of times that the query is issued. Suppose "apple" is the seed entity of domain "electronics", which has contexts "# fruit", "www # com" and "# laptop". If the three are equally weighted, we may rank "banana" high for "electronics" according to the topological structure. However, this result is not good.

[4, 5, 6] use the weight of the edges to weigh the contexts. However, this measurement may cause bias against contexts which cover the entities frequently searched by users. Thus unimportant contexts such as "www # com" corresponding to popular entities can mislead the training process.

We find that if a context covers a lot

[2] # is the placeholder for the named entity, and we only consider queries that contain one named entity

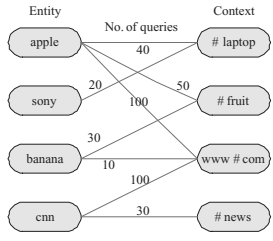

Fig. 2: An example of entity-context bipartite graph.

of named entities in domain d and few in other domains, it is important to d. For example, "www # com" may cover many named entities of different domains, thus it is not important in distinguishing named entities of a domain. However, "# headquarters" cover a lot of entities of "company" and few entities of other domains, thus it is important to "company". Based on the observation, we define a novel measurement to measure the importance of contexts.

Another difference between existing methods and ours is that we not only use query information [5, 6] and click information [4], but also exploit session information. Since queries within one session have a high probability to have the same query intent [8].

What's more, existing methods have high precision but low recall since they solve the problem in a single-step manner. To improve the identification recall, the identification and ranking steps iterate in a bootstrapping manner.

The contributions of our work include. (1) We define a novel measurement to measure the importance of contexts. (2) We combine multi-information in an effective probabilistic model to classify named entities. (3) We design the identification and ranking of named entities in a bootstrap-

ping manner.

The paper is organized as follows. In Section 2, we introduce related work. In Section 3, we describe our approach. In Section 4, we report our experimental results. Section 5 concludes our study.

2. Related work

There are three existing works [4, 5, 6] most relevant to our study. [6] is summarized as follows: Given domain d, they pick some seed entities of d, the contexts of which is extracted and weighed with the number of times the queries were issued. The reference vector of d is formed using the weights of the contexts. After using the contexts to cover candidate entities, they rank them according to the similarity between candidate entities' contexts and the reference vector. Not considering named entities' ambiguity, their work does not perform well under many circumstances.

[4, 5] both proposed a topic model called *WS-LDA*, which is a modification of the *LDA* model [9] to solve the problem that entities are ambiguous. They take each entity as a "document" and associated contexts as "words". Each document is labeled with predefined domains, and then the topic model is learned to get the probability that contexts belong to each domain. However, unimportant contexts may be considered important in one domain if the associated entities are very popular in some way.

3. Our approach

In this section, we formalize the mining task, and define a measurement called *domain-importance* to measure the importance of contexts. The we describe how we rank the candidate entities, and present how the method iterates in a bootstrapping manner.

3.1. Problem formalization

Given a set D of predefined domains such as "fruit", click-through data C and a set $E(d)$ of seed entities SE for each domain $d \in D$. Each click-through record r consists of a query q, an URL u, and a session ID s. The goal is to find a set of candidate named entities CE of D from C.

We scan C to obtain multi-information of $E(d)$, and create the *domain table* (Table 1). Note that for seed entities belonging to several domains such as "apple", we add all the domains to the Domain attribute.

3.2. Domain-importance

Domain-importance (DI) consists of two parts: *entity frequency (EF)* and *domain differentiation (DD)*.

3.2.1. Entity frequency

Given domain d, the EF of a context t is defined as follows.

$$EF(t,d) = \frac{\sum_{e \in E(d)} Cover(t,e)}{|E(d)|}, \quad (1)$$

where $|E(d)|$ denotes the number of seed entities in d. $Cover(t,e) \in \{0,1\}$ denotes if t covers entity e in a query of the click-through data.

3.2.2. Domain differentiation

Given a number of domains D, the DD of a context t is defined as follows.

$$DD(t) = \frac{|D|}{\sum_{d \in D} Exist(t,d)}, \quad (2)$$

where $|D|$ is the size of D. $Exist(t,d) \in \{0,1\}$ denotes if there exists a seed

Query ID	Domain	Entity	Context	URL	Session ID
1	Electronics,Fruit	Apple	# laptop	www.amazon.com	1
2	Electronics	Sony	# laptop	www.sony.com	1
3	Electronics,Fruit	Apple	# laptop	www.google.com	2
4	Electronics	Nokia	# phone	www.amazon.com	3
4	Electronics	Nokia	# phone	www.nokia.com	3

Table 1: Domain table.

entity e in domain d such that $Cover(t, e) = 1$.

Given $d \in D$, we say t is important, if t can cover many entities in d, and cover few entities in other domains. Thus we define the DI of t in d as follows.

$$DI(t, d) = EF(t, d) \times \log DD(t). \quad (3)$$

After calculating the DI of the contexts, we set a threshold α to eliminate those whose DI are less than α. The remaining contexts are used to extract candidate entities.

3.3. Extracting and ranking named entities

As mentioned above, entities are ambiguous. The candidate entities CE of domain d we extract using contexts may belong to multi-domains. If we calculate the probability that a candidate entity e belong to d, we can rank CE according to the probability and return the top k of CE as results. In the process, we exploit multi-information of named entities.

3.3.1. Click information

URLs that users click for a query represent their search intent in mind. If a user clicks a URL (e.g. www.imdb.com) that corresponds many named entities of a domain (e.g. "movie"), we can say URL is important in telling whether a corresponding named entity belongs to the domain.

3.3.2. Session information

Studies [8] show that queries within one session tend to have similar search intents. As named entities reflect search intents, we can assume that entities in a session tend to belong to the same domain.

3.3.3. Probabilistic Model

Suppose we have a set of predefined M domains $D = \{d_1, \ldots, d_M\}$. For N records of click-through data whose queries containing a candidate entity e, URLs are denoted by $U = \{u_1, \ldots, u_N\}$, and session IDs are denoted by $S = s_1, \ldots, s_N$. Similarly, the set of contexts of e is denoted by $T = \{t_1, \ldots, t_N\}$. The probability that e belongs to d_i is inferred as follows:

$$\begin{aligned} p(d_i|e) &\overset{1}{=} \sum_{j=1}^{N} p(d_i, t_j, u_j, s_j|e) \\ &\overset{2}{=} \sum_{j=1}^{N} p(d_i|t_j, u_j, s_j, e) p(t_j, u_j, s_j|e) \\ &\overset{3}{=} \sum_{j=1}^{N} p(d_i|t_j, u_j, s_j) p(t_j, u_j, s_j|e) \\ &\overset{4}{=} \frac{\sum_{j=1}^{N} p(d_i|t_j, u_j, s_j)}{N} . \end{aligned} \quad (4)$$

In step 1, we expand $p(d_i|e)$ to the joint distribution with T, U and S. Step 2 rewrites the equation by Bayes' theorem. In step 3, because d_i is conditionally independent of e given t_j, u_j and s_j, we remove e. As there are N records, we assume $p(t_j, u_j, s_j|e) = \frac{1}{N}$ in step 4.

Based on Bayes' theorem, we have $p(d_i|t_j, u_j, s_j) = \frac{p(d_i)p(t_j,u_j,s_j|d_i)}{p(t_j,u_j,s_j)}$. Assuming that contexts, URLs and sessions are independent of each other and $p(d_i) = \frac{1}{M}$, we have

$$\begin{aligned} p(d_i|t_j, u_j, s_j) &\propto p(t_j, u_j, s_j|d_i) \\ &= p(t_j|d_i)p(u_j|d_i)p(s_j|d_i) \end{aligned}$$

$$\begin{aligned} p(t_j|d_i) &= EF(t_j, d_i) \\ p(u_j|d_i) &= \frac{\sum_{e \in E(d) Cover(u_j,e)}}{|E(d)|} \\ p(s_j|d_i) &= \frac{\sum_{e \in E(d) Cover(s_j,e)}}{|E(d)|} \end{aligned}$$

where $EF(t_j, d_i)$ and $E(d)$ are defined in Section 3.2. $Cover(u, e) \in \{0, 1\}$ denotes if there exists a record in click-through data that contains u and a query containing e. And $Cover(s, e) \in \{0, 1\}$ denotes if s contains a query containing e.

If the URLs and sessions of candidate entities do not exist in the domain table, we use Laplacian smoothing to smooth the calculation.

3.4. Bootstrapping

Although we could mine named entities by the above steps, the recall relies on the representativeness of the seed entities. Therefore, we need to address how to improve recall based on limited seed entities. After ranking the candidate entities for a domain, we can say that the candidate entities with high ranks have a high probability of belonging to the domain. Thus we can set a threshold to add the high ranking ones to the seed set and repeat the steps. As the information of original seed entities has been stored in the domain table, we only need to extract and store the information of newly added named entities.

4. Experiments

In this section, we introduce the data set we use, and demonstrate the effectiveness of our approach by comparing with several baselines and existing approaches.

4.1. Data set

The data set we use is from [1] with 12.25 million clicks. We do the data preprocessing as follows. (1) Only keeping characters and numbers in queries. (2) Only keeping the host name of each URL. (3) Segmenting sessions if the time interval between clicks of them exceeds 30 minutes, the strategy widely used in previous works [8]. And we define four domains including "Movie", "Book", "Music" and "Game", and choose 200 seed entities, the domain of which is labeled by ten human labelers.

4.2. Evaluation method

To show if DI and multi-information are useful in our model, besides *EI*, we implement three baselines, *NoDI*, *NoClick* and *NoSession*. In *NoDI*, we do not use DI to eliminate unimportant contexts, in *NoClick*, we only use context information in the probabilistic model, while in NoSession, we do not use session information. We also implement the algorithms proposed in

[6] and [4], and we call them NEQ and NEM respectively.

We use $Precision@N = \frac{N_{correct}}{N}$ to measure the accuracy of the methods, where $N_{correct}$ denotes the number of correctly identified and classified named entities in the top N candidate entities.

4.3. Performance and comparison

We list the top 5 most important contexts for four domains in Table 2, where contexts with high DI for each domain represent users' intent to search for named entities.

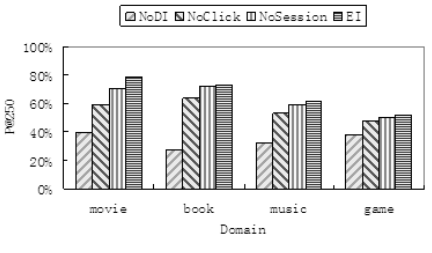

Fig. 3: Comparisons with baselines.

For about 2.4 million candidate named entities identified, we manually judge the top 250 entities ranked for each domain. Figure 3 shows the comparison between baselines and ours.Figure 4 indicates that our model outperforms *NEQ* and *NEM*. By analyzing their results, we find that most errors are brought by unimportant contexts such as "www # com" and "free #", etc. However, by utilizing DI, our method eliminates those unimportant contexts.

To test if the bootstrapping strategy works effectively, We split the 200 original seed entities into two sets. The first set including 120 entities is used as new seed entities, and those click-through records that contain any of the

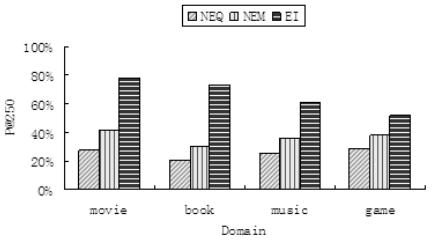

Fig. 4: Comparisons with *NEQ* and *NEM*.

200 entities are input as click-through data. The second set including 80 entities is used as a test set. Figure 5 shows that as the iteration goes on, recall increases.

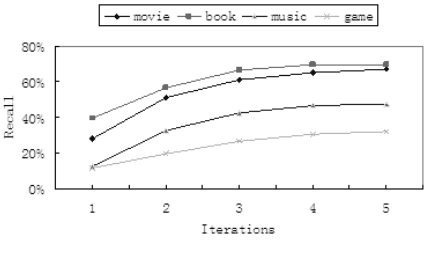

Fig. 5: Recall at each step of iterations.

5. Conclusion

In this paper, we address the problem of identifying and classifying named entities by proposing a novel measurement to measure the importance of contexts and developing a probabilistic model to rank the named entities using multi-information. The identification and ranking steps of our method are designed in a bootstrapping manner to improve recall.

References

[1] Microsoft research. microsoft live labs: Accelerating search in aca-

Movie	# film, # imdb, dvd #, quotes from #, movies #
Music	lyrics to #, lyrics #, lyrics for #, # the song, # lyric
Game	# cheat, # for xbox, download #, # online, # cdkey
Book	book #, # lesson plans, summary of #, # spark notes, # best seller

Table 2: Contexts with high DI for each domain.

demic research 2006 rfp awards. research grant, 2006.

[2] D. 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 (ANLP 97)*, 1997.

[3] Ganesh Agarwal, Govind Kabra, and Kevin Chen-Chuan Chang, Towards rich query interpretation: walking back and forth for mining query templates. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 1–10, New York, NY, USA, 2010.

[4] Gu Xu, Shuang-Hong Yang, and Hang Li, Named entity mining from click-through data using weakly supervised latent dirichlet allocation. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1365–1374, New York, NY, USA, 2009.

[5] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li, Named entity recognition in query. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274, New York, NY, USA, 2009.

[6] Marius Paşca, Weakly-supervised discovery of named entities using web search queries. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 683–690, New York, NY, USA, 2007.

[7] Uichin Lee, Zhenyu Liu, and Junghoo Cho, Automatic identification of user goals in web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 391–400, New York, NY, USA, 2005.

[8] Huanhuan Cao, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li, Context-aware query suggestion by mining click-through and session data. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 875–883, New York, NY, USA, 2008.

[9] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

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