7-15-2012

Effectively Grouping Named Entities From Click-Through Data Into Clusters Of Generated Keywords1

Xuan Jiang
Key Labs of Data Engineering and Knowledge Engineering, MOE, China; School of Information, Renmin University of China, Beijing, China, jx@ruc.edu.cn

Hongyan Liu
Department of Management Science and Engineering, Tsinghua University, Beijing, China, hyliu@tsinghua.edu.cn

Jun He
Key Labs of Data Engineering and Knowledge Engineering, MOE, China; School of Information, Renmin University of China, Beijing, China, hejun@ruc.edu.cn

Rui Zhu
Key Labs of Data Engineering and Knowledge Engineering, MOE, China; School of Information, Renmin University of China, Beijing, China, ruizhu@ruc.edu.cn

Xiaoyong Du
Key Labs of Data Engineering and Knowledge Engineering, MOE, China; School of Information, Renmin University of China, Beijing, China, duyong@ruc.edu.cn

EFFECTIVELY GROUPING NAMED ENTITIES FROM CLICK-THROUGH DATA INTO CLUSTERS OF GENERATED KEYWORDS[1]

Xuan Jiang, Key Labs of Data Engineering and Knowledge Engineering, MOE, China
School of Information, Renmin University of China, Beijing, China,
jx@ruc.edu.cn

Hongyan Liu, Department of Management Science and Engineering, Tsinghua University, Beijing, China,
hyliu@tsinghua.edu.cn

Jun He †, Key Labs of Data Engineering and Knowledge Engineering, MOE, China
School of Information, Renmin University of China, Beijing, China,
hejun@ruc.edu.cn

Rui Zhu, Key Labs of Data Engineering and Knowledge Engineering, MOE, China
School of Information, Renmin University of China, Beijing, China,
ruizhu@ruc.edu.cn

Xiaoyong Du, Key Labs of Data Engineering and Knowledge Engineering, MOE, China
School of Information, Renmin University of China, Beijing, China,
duyong@ruc.edu.cn

Abstract

Many studies show that named entities are closely related to users' search behaviors, which brings increasing interest in studying named entities in search logs recently. This paper addresses the problem of forming fine grained semantic clusters of named entities within a broad domain such as "company", and generating keywords for each cluster, which help users to interpret the embedded semantic information in the cluster. By exploring contexts, URLs and session IDs as features of named entities, a three-phase approach proposed in this paper first disambiguates named entities according to the features. Then it properly weights the features with a novel measurement, calculates the semantic similarity between named entities with the weighted feature space, and clusters named entities accordingly. After that, keywords for the clusters are generated using a text-oriented graph ranking algorithm. Each phase of the proposed approach solves problems that are not addressed in existing works, and experimental results obtained from a real click through data demonstrate the effectiveness of the proposed approach.

Keywords: Named Entity Clustering, Click-through Data, Keyword Generation.

†Corresponding author.

[1] This work was supported by the National Natural Science Foundation of China under Grant No. 70871068, 70890083 and 71110107027, the 973 program of China under Grant No. 2012CB316205, and HGJ PROJECT 2010ZX01042-002-002-03.

1 INTRODUCTION

Over the course of the past 20 years, there has been increasing interest in managing the web data. As it is important for search engines to understand users' searching intents, click-through data, which records users' searching behaviors, has been intensively investigated recently. According to the analysis conducted by (Guo et al. 2009), about 71% of queries in click-through data contain named entities (i.e. atomic elements of predefined categories such as names of companies, people, locations, etc). Named entities contained in queries reflect users' searching intents. Therefore, analyzing named entities in click-through data can be essentially helpful in capturing users' searching intents, which is important for providing personalized service to users.

State-of-the-art studies analyzing named entities on click-through data fall into two fundamental categories: Named Entity Recognition and Classification (*NERC*) and Named Entity Relatedness Measurement (*NERM*). In this paper, we focus on solving the problem of *NERM* after named entities of certain broad categories have been extracted from click-through data using techniques of *NERC*. Traditional techniques addressing *NERM* on large corpus are mainly based on an assumption that there exist large knowledge repositories so that semantic relatedness between named entities can be induced by structure information and text information inside the knowledge repositories like Wikipedia (Gabrilovich & Markovitch 2007), Search Interface (Bollegala et al. 2007; Liu & Birnbaum 2007) and so on. However, such applications suffer from high human effort (based on Wikipedia) or inefficiency (based on Search Interface).

Prior work (Jain & Pennacchiotti 2010) has relaxed the assumption by exploring features of named entities and clustering named entities based on click-through data, which can be easily acquired by search engines. Their studies show that in contrast to web documents modelling the ***web space***, click-through data models the ***user space***. The web space contains general knowledge about concepts in an objective way, whereas the user space captures opinions of the crowd that directly express users' intents. For example, "Britney Spears" is similar to other singers like "Celine Dion" in the web space, but in the user space, she is more similar to people like "Paris Hilton" in terms of gossiped activities.

There are several disadvantages of Jain and Pennacchiotti's work (Jain & Pennacchiotti 2010). First, their approach doesn't consider the problem of named entity ambiguity. Based on their approach, one entity can only be in one cluster/category, which is inappropriate in many cases. For example, "Michael Jordan" can either refer to a basketball player or a professor. Second, the weights defined for the features in their approach are not effective in interpreting the semantic information of named entities, which we will discuss further. Third, their approach only uses the context feature (i.e. keywords left after named entity is removed from queries) and the click feature of named entity (i.e. URLs clicked corresponding to queries containing named entity), which is insufficient because the session feature (i.e. session IDs of queries containing named entity) is also an important source to improve the accuracy of *NERM*. Forth, clusters formed by their approach have no corresponding abstracts, which are desirable for users to understand the semantic information of the clusters in real world applications.

In this paper, we present a three-phase approach to solve the problems mentioned above. The first phase (named entity disambiguation) extracts the features of named entities in click-through data and disambiguates named entities according to the features. The second phase (named entity clustering) weights the features with a novel and effective measurement and clusters named entities according to the similarity calculated based on the weighted features. The third phase (keywords generation) generates keywords as abstracts for each cluster using a text-oriented graph ranking algorithm, which help users to understand the semantic information of the clusters. These clusters built based on the proposed approach can be essentially useful in many applications such as general web search, product searching in consumer marketing and so on, and we develop a demo system in which the clusters and corresponding keywords are used for query suggestion.

To illustrate the approach, suppose there are four named entities in a broad domain "company", which are "Microsoft", "IBM", "HP" and "Time Warner". They are different in terms of semantic information, because Microsoft, IBM and HP are technology companies, whereas Time Warner focuses on entertainment. What's more, among Microsoft, IBM and HP, Microsoft is considered as a software product manufacturer while IBM and HP are computer vendors. For those queries containing the companies, we extract the features of contexts, URLs and session IDs corresponding to the queries. Then we properly weight the features and calculate the semantic similarity between named entities according to the features. After that, these four companies are clustered and assigned with keywords as shown in Figure 1.

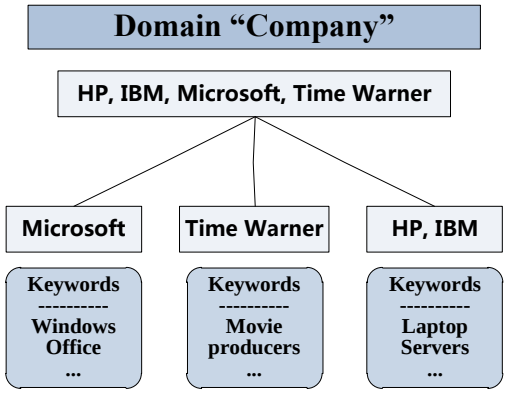

Figure 1. A toy example of the clusters in Domain "Company".

Our contributions are as follows. (1) We solve the problem of named entity ambiguity by clustering the corresponding features. (2) We propose a novel measurement that effectively captures the importance of features in expressing the semantic information of named entities. (3) We explore the session feature of named entities, which turns out to improve the accuracy of *NERM*. (4) We propose an approach for clustering named entities and assigning the clusters with keywords using a text-oriented graph ranking algorithm, which increases the interpretability of the clusters. (5) We extensively evaluate the algorithm over a large real life click-through dataset, and apply the clusters and corresponding keywords to the query suggestion module of a search engine system, in which the user feedbacks show the interestingness and usefulness of the module.

The paper is organized as follows. It first introduces the background of our problem, and then describes the three phases of our approach. After that, algorithms in the approach are reviewed and experimental results are shown to verify the algorithms. Finally the conclusion and future work are given.

2 RELATED WORK

There exist many studies which focus on organizing information by leveraging existing knowledge repositories like Wikipedia (Gabrilovich & Markovitch 2007), Search Interface (Bollegala et al. 2007; Liu & Birnbaum 2007) and so on. Different from these traditional studies, some studies try to solve the problem by modelling the user space, which is exploring information inside click-through data. As query is a fundamental component in web search, it is desirable to analyze them in order to provide better searching service. Chuang and Chien (2002) use hierarchical clustering to organize queries into hierarchical clusters based on the features of words, while Baeza-Yates and Tiberi (2007) study the semantic relation between queries by analyzing the graph generated according to the distance between queries.

By deeply studying the internal structure of queries, some studies (Guo et al. 2009) bridge the gap between named entities and user search intent behind queries by establishing some features of named

entities in click-through data, such as contexts and clicked URLs. After that, the problem of organizing named entities/intents has emerged as a hot research topic. Yin and Shah (2010) use three different methods to organize intents (i.e. contexts of named entities in queries) into taxonomy by using the features of named entities and clicked URLs, while Jain and Pennacchiotti (2010) cluster the named entities by proposing the framework of open information extraction on click-through data.

Comparing with existing works listed above, our work differs with them in following aspects. First, our work addresses the problem of named entity ambiguity in the process of organizing named entities, which is not considered before. Second, existing works weight the features of named entities using the number of records which contain them, which turns out to yield biased results in out experiments. Therefore, our work proposes a novel measurement to weight the features, which is effective in modelling the semantic information of features. Third, our work explores the session feature of named entities in click-through data, which enriches the feature space and improves the accuracy of final results. Fourth, each cluster of named entities generated by our algorithm has keywords which are assigned by a text-oriented graph ranking algorithm. These keywords are helpful in real world applications because they lower the dimensionality of the features space in each cluster and assist users to interpret the semantic information of the cluster.

Another category of studies related to our work is unsupervised keyword extraction from text corpus. Existing algorithms accomplish the task by assigning the saliency score to each candidate keyword, and consider those candidates with high scores as keywords (Muñoz 1996; Steier & Belew 1993). (Mihalcea & Tarau 2004; Wan et al. 2007) later propose graph-based ranking algorithms to generate keywords from text, which they prove to outperform existing algorithms by producing more promising results. In our approach, we apply a graph-based ranking algorithm to clusters of named entities by utilizing their features, and the results turn out to be interesting.

3 OUR APPROACH

3.1 Notations

Our problem is based on the fact that named entities of certain broad categories (domain) have been identified from queries in click-through data, as shown in Table 1, using methods that have been well studied (Pasca 2007; Guo et al. 2009). Let *d* denote a domain, and *e* denote a named entity that belongs to *d*. A record of click-through data consists of query *q*, URL *u* and session ID *s*. After extracting *e* from click-through data, we collect a set of tuples <*c*; *u*; *s*> for *e*, in which a context *c* of *e* is the remaining terms of *q* containing *e* after *e* is removed from *q*. For example, given a record of <Microsoft vista; http://www.microsoft.com; 2079326> shown in Table 1, "# vista" (# denotes a placeholder for named entity) is the context of named entity "Microsoft", which is obtained by removing "Microsoft" from query "Microsoft vista", and the corresponding tuple is <# vista; http://www.microsoft.com; 2079326>.

Session ID	Query	URL
2079326	fujitsu	http://www.Fujitsu.com
2079326	ibm thinkpad	http://pc.ibm.com
2079326	Microsoft vista	http://www.microsoft.com
8653507	ibm	no-clicking
8653507	ibm headquarters	http://www.whiting-turner.com

Table 1. *A slice of Click-through Data.*

Many studies (Cao et al. 2008) show that co-occurring in sessions can statistically prove the correlation between queries. By analogy, session feature also shows the correlation of named entities in terms of semantic information. As context *c* and URL *u* directly show the user searching behavior

for entity *e*, we can say that by utilizing the three features, we can capture users' searching intents, which is closely related to the semantic information of named entities in queries.

3.2 Named Entity Disambiguation

Named entity disambiguation is one of the main challenges for research about named entity. Named entities are ambiguous, because one name can refer to different named entities and one named entity can have different names, which can lead to inferior performances of applications. For example, in the domain of "people", "Michael Jordan" can either refer to a famous basketball player or a well known professor in statistics. What's more, people may sometimes refer to them as "MJ", "Jordan" and so on. Without named entity disambiguation, the relatedness measured between "Michael Jordan" and other basketball players or statistic professors can be misleading, because the information about "Michael Jordan" collected from click-through data are mix of basketball player and statistical community.

For the problem of one named entity having various names, we differentiate them in the second phase. If named entities have similar semantic information, they will be grouped together in the second phase of our method. In the first phase of our approach, we focus on solving the problem of one name referring to different named entities.

It's a common sense that one click record bears single search intent in user's mind. Therefore, we assume that one tuple refers to only one named entity. As mentioned above, features reflect the semantic information of named entities, so tuples of one named entity should be similar in features. Thus based on the assumption, for tuples generated according to an ambiguous named entity, we group the similar tuples and regard each cluster as one named entities without ambiguity.

For two tuples t_i and t_j of a named entity *e*, $sim(t_i,t_j)$ between them is defined as follows:

$$sim(t_i,t_j)=\frac{\theta(c_i,c_j)+\theta(u_i+u_j)+\theta(s_i+s_j)}{3} \quad (3.1)$$

in which

$$\theta(x_i,x_j)=\begin{cases}1, & if\ x_i=x_j\\ 0, & if\ x_i\neq x_j\end{cases}$$

As the number of real named entities to which each ambiguous named entity refers is unknown, here we apply Hierarchical Agglomerative Clustering (HAC) method (Mirkin 1996) to find the hidden real named entities. We use average-linkage to measure the similarity between two clusters.

After applying HAC, we group the tuples related to one named entity into several clusters. We regard tuples within one cluster are about one named entity without ambiguity, and different clusters represent different named entities although they may have the same named entity names. Therefore, named entities with the same name can belong to different clusters. Experiments show that after the process of named entity disambiguation, the results become more satisfactory.

3.3 Named Entity Clustering

3.3.1 Feature weighting

So far, given a domain *d* and a set of named entities of this domain, we get a set of clusters from disambiguation process based on click-through records, each of which contains a set of tuples, and represents a named entity without ambiguity. Then, in order to measure similarities between named entities, we describe each named entity using three features, which are contexts, URLs and session IDs.

Intuitively, two named entities are similar if they share a lot of features in common. When comparing the features of named entities, it is necessary to study the importance of the features, which weights the extent to which a feature represents the semantic information of the associated named entity. Existing feature weighting methods don't work well here, because they are misled by users' complex

searching intents. According to Lee et al. (2005), the goals of users using search engine can be divided into "navigational queries" and "informational queries". Navigational queries don't represent the semantic information of entities, because people issue such queries just to navigate to certain websites. Some informational queries are related to only one named entity, and these queries can mislead the calculation of semantic similarity between named entities.

For example, given a domain "company", context "# vista" is closely related to Microsoft because it's a specific product belonging to Microsoft. What's more, URL "www.microsoft.com" is frequently clicked by users who want to navigate to Microsoft, which is not helpful in the similarity calculation process either. Based on the weighting method used in existing works, these features may be weighted high because they have a high correlation with Microsoft and have a fairly small *IDF* (inverse document (query) frequency). However, both of them are not important in terms of measuring similarity of two named entities in fact, because "# vista" and "www.microsoft.com" are only related to "Microsoft", and have weak relationship with other companies, so they are not important in interpreting the semantic information of domain "company". For features like "www.wikipedia.org", they are related with named entities in all kinds of domains, so they are not important in interpreting the semantic information of domain "company" either.

Therefore, we define a novel measurement called *Semantic Importance* (*SI*) that can effectively weight the features in capturing the semantic information of named entities. Note that our work is based on the fact that named entities of certain broad categories have been extracted from click-through data using techniques of *NERC*. Thus *SI* we define and clusters we build is for certain category/domain.

First, we consider the semantic importance of context, URL and Session ID. In the follow discussions, feature *f* refers to a context *c* or a URL *u* or a Session ID *s*.

For a domain *d*, there are two requirements for *f* to be of high semantic importance to named entities of *d*: *f* should have strong relationship with named entities belonging to *d*, and *f* should have weak relationship with named entities belonging to other domains. To this end, we define two measures, Entity Frequency (*EF*) and Domain Differentiation (*DD*) for each feature *f* which are defined as follows.

$$EF_{f,d} = \frac{n_{f,d}}{|d|} \qquad DD_{f,d} = \frac{EF_{f,d}}{\sum_{i=1}^{|D|} EF_{f,d}} \tag{3.2}$$

in which $|d|$ denotes the number of named entities in domain *d*, $n_{c,d}$ denotes the number of named entities in *d* that have *f* as their features, and $|D|$ denotes the number of domains.

As mentioned above, if feature *f* is important in capturing the semantic information of domain *d*, $EF_{f,d}$ and $DD_{f,d}$ should be both high, which leads to the definition of measurement *SI* for the semantic importance of *f* in *d* as follows:

$$SI_{f,d} = \frac{2EF_{f,d} \cdot DD_{f,d}}{EF_{f,d} + DD_{f,d}} \tag{3.3}$$

Note that when *f* refers to context *c*, $SI_{f,d}$ becomes $SI_{c,d,}$, for URL *u*, it becomes $SI_{u,d,}$, and for Session ID *s*, it becomes $SI_{s,d}$.. By defining SI, we can make sure that the features listed below are considered low in interpreting the semantic information of a concerned domain such as "company".

- Features that do not reflect the semantic information of "company" such as "www # com" or "www.wikipedia.org".
- Features that are only related to specific named entities such as "# vista".

3.3.2 Named Entity Clustering

In this section, we use three vectors to model the three types of features of named entities (context, URL and session ID), and calculate the semantic similarity between named entities by combining the

cosine similarities according to the three vectors. After that, we apply *HAC* to cluster the named entities.

Take feature context as an example. Given two named entities e_1 and e_2 of a domain d, their context vectors are $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ respectively. Each dimension of the vector corresponds to a distinct context that belongs to named entities of domain d. And for each dimension, if the corresponding context belongs to e, the weight of the dimension is set to be $SI_{c,d}$, otherwise the weight is 0. For example, suppose domain d has two named entities e_1 and e_2, and e_1 has contexts c_1 and c_3 while e_2 has contexts c_1 and c_2. Then we have $\boldsymbol{v_1} = <SI_{c_1,d}, 0, SI_{c_3,d}>$ and $\boldsymbol{v_2} = < SI_{c_1,d}, SI_{c_2,d}, 0 >$. We use cosine similarity to estimate the similarity between $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$, $\boldsymbol{s_c(e_1,e_2) = \vec{v_1} \cdot \vec{v_2} / \|\vec{v_1}\| \cdot \|\vec{v_2}\|}$, and $s_c(e_1, e_2)$ ranges from 0 to 1. The similar strategy can be applied in calculating $s_u(e_1,e_2)$ and $s_s(e_1,e_2)$.

Semantic similarity. As described above, we have three similarities calculated based on the three features. The next step is to combine them into a single similarity that represents the semantic similarity between named entities. We define semantic similarity $s(e_1, e_2)$ as follows:

$$\boldsymbol{s(e_1,e_2) = \sqrt[3]{s_c(e_1,e_2) s_u(e_1,e_2) s_s(e_1,e_2)}} \qquad (3.4)$$

Based on the semantic similarity, the named entities under one domain are clustered using *HAC*.

3.4 Keyword Generation

In order to improve the interpretability of the named entity clusters, we propose an iterative graph ranking algorithm to automatically extract keywords out of the features of clusters.

Overview. We consider contexts and URLs as ideal candidate abstracts for the clusters of named entities, because they represent users' search intent in mind. Therefore, the objective here is to select salient contexts and URLs of each cluster, which can reduce to the keyword generation problem.

According to empirical study, we make the following assumption.

Assumption. A context or URL is salient if it belongs to many salient named entities, and a named entity that is similar to many salient named entities is salient itself.

Based on the assumption above, we consider named entities as "hubs", which influence their corresponding contexts and URLs. What's more, the saliencies of named entities influence each other according to the relationship between them.

Given a cluster of named entities and corresponding features, we model them in a relationship graph as shown in Figure 2.

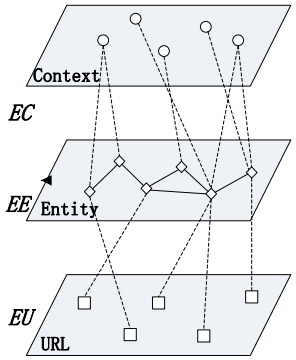

Figure 2. *Relationship graph of named entities in one cluster.*

In the relationship graph, there are three layers of concepts that indicate context, named entity and URL respectively, in which named entity serves as "hub". *EC* denotes the containing relationship between named entities and corresponding contexts, *EU* represents the click relationship between

named entities and URLS respectively; and *EE* indicates the semantic similarity relationship between named entities.

Given the graph, our approach first assigns an initial saliency score to each node in three layers. Then, based on the structure of *EC*, *EU* and *EE*, our approach iteratively updates the saliency score of each node. When the saliency distribution in the graph converges, our approach outputs the saliency scores of each node, and selects those contexts and URL with promising saliency scores as keywords of the cluster.

Formulation. Given a cluster of named entities $E = \{e_i | 1 \leqslant i \leqslant n\}$ and corresponding contexts $C = \{c_j | 1 \leqslant j \leqslant m\}$ and URLs $U = \{u_k | 1 \leqslant k \leqslant p\}$, we treat them as nodes in the relationship graph. We use undirected graph G_{EE}, G_{EC} and G_{EU} to model the relationship *EE*, *EC* and *EU* respectively.

In G_{EE}, the adjacency matrix is denoted as $\boldsymbol{T} = [T_{ij}]_{n \times n}$, and each entry in the matrix indicates the weight of a corresponding edge in the graph. Formally, $\boldsymbol{T}=[T_{ij}]_{n \times n}$ is defined as follows:

$$\boldsymbol{T}_{ij} = \begin{cases} \boldsymbol{s(e_i, e_j),} & \boldsymbol{if\ i \neq j} \\ \boldsymbol{0,} & \boldsymbol{if\ i = j} \end{cases} \tag{3.5}$$

where e_i and e_j denotes the corresponding named entity of node i and j, and the weight of the link between them is defined as the semantic similarity between e_i and e_j.

Similarly, in G_{EC} and G_{EU}, the adjacency matrices, denoted as $\boldsymbol{V} = [V_{ij}]_{n \times m}$ and $\boldsymbol{W} = [W_{ij}]_{n \times p}$, are defined as follows:

$$\boldsymbol{V}_{ij} = \begin{cases} 1, & \boldsymbol{if\ c_j\ belongs\ to\ e_i} \\ 0, & \boldsymbol{otherwise} \end{cases}, \quad \boldsymbol{W}_{ik} = \begin{cases} 1, & \boldsymbol{if\ u_k\ belongs\ to\ e_i} \\ 0, & \boldsymbol{otherwise} \end{cases} \tag{3.6}$$

where V_{ij} equals to 1 if c_j is one of the contexts of e_i and W_{ik} equals to 1 if u_k is clicked for one of the queries containing e_i.

Iteration. We use three vectors $\boldsymbol{t} = [t(e_i)]_{n \times 1}$, $\boldsymbol{v} = [v(c_j)]_{m \times 1}$ and $\boldsymbol{w} = [w(u_k)]_{p \times 1}$ to indicate saliency distributions of named entities, contexts and URLs respectively. Based on the assumptions mentioned above, we propose following iteration steps.

1. The entries in $\boldsymbol{t}$ are all set to 1, and the entries in $\boldsymbol{v}$ and $\boldsymbol{w}$ are set to be their corresponding *SI* value. The reason for this step is that SI value of each feature represents its salient semantic importance to the cluster before the convergence of saliency distribution.

2. Calculate and normalize the saliency scores of $\boldsymbol{v}$, $\boldsymbol{w}$ and $\boldsymbol{t}$ as follows.

$$\boldsymbol{v}^{(n)} = \boldsymbol{Normalize}(\boldsymbol{V}^T)\boldsymbol{t}^{(n-1)}, \quad \boldsymbol{v}^{(n)} = \frac{\boldsymbol{v}^{(n)}}{\left\|\boldsymbol{v}^{(n)}\right\|_1} \tag{3.7}$$

$$\boldsymbol{w}^{(n)} = \boldsymbol{Normalize}(\boldsymbol{W}^T)\boldsymbol{t}^{(n-1)}, \quad \boldsymbol{w}^{(n)} = \frac{\boldsymbol{w}^{(n)}}{\left\|\boldsymbol{w}^{(n)}\right\|_1} \tag{3.8}$$

$$\boldsymbol{t}^{(n)} = \boldsymbol{Normalize}(\boldsymbol{V})\boldsymbol{v}^{(n)} + \boldsymbol{Normalize}(\boldsymbol{W})\boldsymbol{w}^{(n)} + \boldsymbol{Normalize}(\boldsymbol{T})\boldsymbol{t}^{(n-1)}, \quad \boldsymbol{t}^{(n)} = \frac{\boldsymbol{t}^{(n)}}{\left\|\boldsymbol{t}^{(n)}\right\|_1} \tag{3.9}$$

where n denotes the n-th iteration, and the "Normalize" function normalizes each matrix to make sure that the sum of each row equals to 1.

3. The iteration steps stop if the convergence of the saliency distributions is met. In other words, the steps stop if the max average difference of $\boldsymbol{v}$, $\boldsymbol{w}$ and $\boldsymbol{t}$ between two successive iterations is lower than a certain threshold.

Given the iteration steps, we can induce following equations, which fulfill the assumption proposed above.

$$t(e_i) \propto \sum_j \frac{V_{ij} v(c_j)}{\sum_i V_{ij}} + \sum_k \frac{W_{ik} w(u_k)}{\sum_i W_{ik}} \qquad v(c_j) \propto \sum_i \frac{V_{ij} t(e_i)}{\sum_j V_{ij}}, \qquad w(u_k) \propto \sum_i \frac{W_{ik} t(e_i)}{\sum_i W_{ik}} \tag{3.10}$$

After the iteration converges, we assign the saliency scores to contexts and URLs in the cluster, and output those with high saliency scores as keywords.

4 APPROACH REVIEW AND SYSTEM DEMONSTRATION

This section gives a brief review of our approach, *NECK* (Named Entity Clustering and Keywords Generation), building clusters of named entities with keywords from click-through data. The flow chart is described in Figure 3.

At the beginning, given several domains, for each domain we obtain a list of named entities from click-through data. This can easily be done since there are many well studied algorithms. After that, we scan the records of click-through data, retrieve the features of the lists of named entities, and calculate the semantic importance *SI* of the features of named entities in each domain.

For named entities of each domain, we calculate the semantic similarity between them, and cluster them based on the similarity. After that, each cluster is assigned with keywords using the graph ranking algorithm we propose.

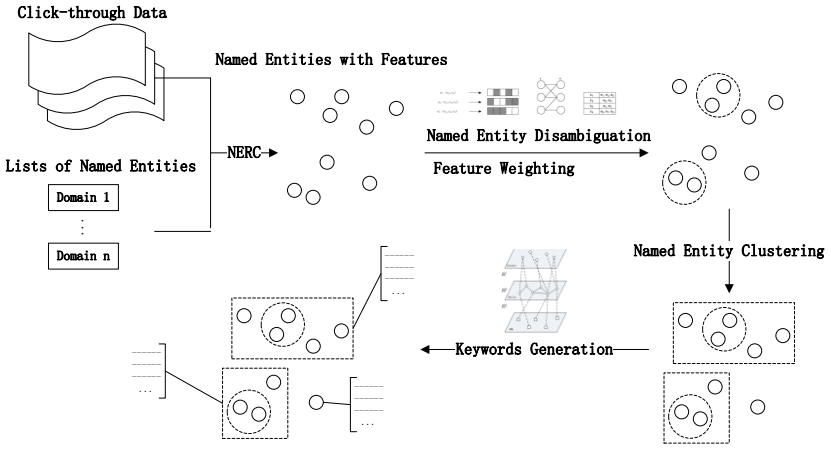

Figure 3. *Flow chart of NECK.*

Query suggestion based on *NECK*. Different from traditional query suggestion algorithms, methods we propose are practical in applying to real world applications in a different view of named entity. For example, for query suggestion, if a user queries "IBM stock price", other methods may suggest "IBM laptop" and "IBM server" to user because they are related in the *query* level. However, *NECK* can first identify the similar named entities of "IBM" as "HP" and "Microsoft", and then suggest "HP stock price" and "Microsoft stock price" to the user because they are related in the *named entity* level.

We implement a system called "EntityCenter" (As the content of our system would reveal the authors' information, the website is not included here), in which *NECK* is applied to the module of query suggestions. As shown in Figure 4, given a query issued by a user, the system first identifies and classifies the named entity contained by the user query based on *NERC* methods. Then, query suggestions based on named entity are given.

Under a privacy policy, the searching behaviors of users are recorded by our logging system. By analyzing the logging system, we find that named entities recommended to users are frequently clicked, which shows the interestingness and usefulness of our proposed methods.

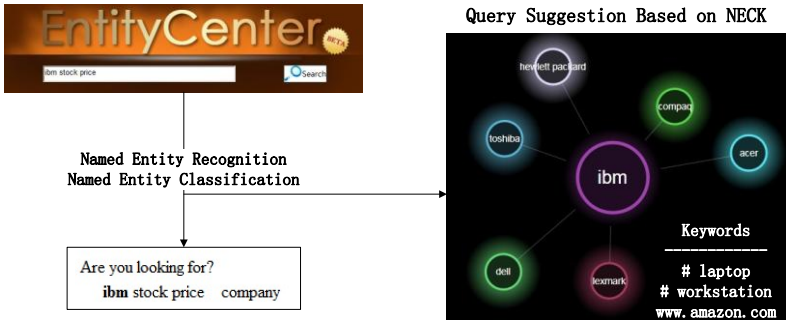

Figure 4. *Demonstration system* "*EntityCenter*".

5 EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our approach. Experiments are conducted in a windows PC with Intel Core 2 Duo CPU and 2GB main memory.

5.1 Experimental Settings

5.1.1 Dataset

The click through data we use is an accessible real life dataset collected by AOL (Pass et al. 2006) from March 2006 to May 2006. Each entry of the dataset consists of a user ID, a query, a clicked URL (or null) and an issued time stamp. We clean queries by only keeping characters and numbers, and other symbols are replaced by spaces. As there is no session attribute in the dataset, we group the entries into sessions such that entries in a session have the same user ID and the intervals of the time stamps are within 30 minutes. This strategy captures the users' search intents at certain time, and is widely used in user behavior research (Cao et al. 2008).

After data preprocessing, we have 9.97 million unique queries, 1.62 million unique URLs, 17.13 million unique query-URL pairs (clicks) and 10.58 million sessions. To avoid being affected by the accuracy of current methods of *NERC*, we use existing lists of named entities and corresponding domains retrieved from Freebase (http://www.freebase.com). The domains and related information are shown in Table 2. With these named entities, we scan the dataset to acquire the tuples of them. To remove noise, we only consider named entities which appear in at least ten unique queries.

Domains	Quantity of named entities	Freebase category
Company	87236	Business/company
City	318131	Location/city&town
Car model	4609	Automotive/model
Politician	51725	Government/politician
Organization	45725	Organization/organization
Basketball player	8334	Basketball/basketball player
School	94615	Education/school

Table 2. *Information of Named Entities.*

5.1.2 Evaluation Metric

We employ twenty undergraduate students in Computer science to take part in the evaluation of our approach. In the evaluation, we use Fleiss' kappa (Shrout & Fleiss 1979) to test the inter-rater liability of the labelers, which measures the degree of the consensus between them.

To evaluate the performance of named entity clustering, we randomly choose 1% of named entities of each domain, and ask labelers to label the similarity between them. For each pair of named entities in

each domain, each labeler grades the semantic similarity with an integer between 0 and 10 by referring to the existing knowledge sources such as Wikipedia. After that, we take an average of the grades by the labelers and divide it by 10 to regard it as *true* similarity. In the process of building this test set, the Fleiss' kappa of labelers is 0.72, which represents a substantial agreement between them.

In the labeled test set, we consider the semantic similarity between named entities as S_{true}. If $S_{true} \geqslant 0.5$, we consider the two associated named entity should be in the same cluster. Then we adopt F-measure, $F = 2PR/(P+R)$, where P and R denote precision and recall respectively.

Let Y be the set of pairs of named entities which should be in the same clusters and N be the set of pairs of named entities which should **not** be in the same clusters, we define P and R as follows:

$$P = 1 - \frac{\sum_{i \in N} InNode(i)}{|N|}, \quad R = \frac{\sum_{i \in Y} InNode(i)}{|Y|} \tag{5.1}$$

where $InNode(i) = 1$ denotes that the named entities of pair i are grouped into one cluster according to *NECK*, otherwise $InNode(i) = 0$. If *NECK* has a high P, it means that most of the dissimilar named entities are put into different clusters. While if *NECK* has a high R, it means that most of the similar named entities are grouped into the same clusters.

We use precision to evaluate the third phase of our approach. For the generated keywords of each cluster, we let labelers decide whether each keyword is suitable in interpreting the semantic information of the cluster. We take a majority vote of labelers, and the precision P_k is defined as follows.

$$P_k = \frac{\sum_i isKey(k_i)}{N_k} \tag{5.2}$$

Where N_k stands for the number of generated keywords, and $isKey(k_i)$=1 if the majority of labelers consider keyword k_i suitable in interpreting the semantic information of the cluster, otherwise $isKey(k_i)$=0 . In this process, the Fleiss' kappa of labelers is 0.67.

5.2 Performance of *NECK*

We implement following algorithms to compare with our approach *NECK*.

- *OIE*: methods used in (Jain & Pennacchiotti 2010). It weights contexts using *CPMI* and eliminates URL using *IDF*. The clustering algorithm it utilizes is Clustering by Committee (*CBC*) (Pantel & Lin 2002).
- *NECK*-A: baseline 1. This is our approach without the process of named entity disambiguation.
- *NECK*-W: baseline 2. This is our approach without using *SI* to weight features.

To address the problem of named entity ambiguity, for each named entity, we disambiguate it by utilizing *HAC* algorithm to group the tuples of the named entity. We determine the threshold about when to stop iteration by sensitivity analysis, which turns out that 0.3 gives fairly satisfying results.
We then calculate the semantic importance measurement SI for each context and URL of each domain.

Domain	Contexts
Company	# headquarters, # promotional codes, # store locater, # nutritional information
City	# festival, best western hotel #, cabins in #, chamber of commerce #, motels in # ca
Car model	rims for #, body kits for #, # curb weight, 2006 # reviews, 2000 # recalls
Politician	project vote smart #, # mail house gov, state senator #, us congressman #, us senator #
Organization	journal of the #, # summer employment, tax exempt #, 2004 # slogan, founder of the #
Basketball player	# nba, # high school stats, # shoe size, # championship rings, # stats
School	# class reunion, # year book, # for the deaf, # of irish dance, # class of 1972

Table 3. *Top Five Contexts with Highest SI in Seven Domains*.

Take context for example. We rank contexts of each domain according to SI in descending order. Table 3 shows the top five contexts with the highest SI of each domain, and we can see that these contexts are meaningful in telling the semantic information of named entities in each domain

To compare with *OIE*, we extract the features of an ambiguous named entity "Apple" of domain "**company**". We weight the features by the method we use in the first phase of our approach and *OIE*, respectively. Table 4 shows the top 5 contexts and URLs with highest weight of Apple.

	Contexts	URLs
NECK	# store locator , # store locations # coupon codes, # promo codes # discount code	abusaki.com, stereo411.com plemix.com, pcauthority.com.au affordablecomputers.com
OIE	# vacations, # ipod # pie, # mac # store	apple.com, applevacations.com applebottoms.com, cooks.com southernfood.about.com

Table 4. *Top Five Contexts and URLs Found by OIE and NECK*

We can see that contexts and URLs got by our approach are meaningful in telling that Apple is a company in the company domain. However, without named entity disambiguation and SI, contexts and URLs found by *OIE* are mostly navigational such as "apple.com", ambiguous such as "# pie" and "southernfood.about.com". These contexts and URLs can severely mislead the process of clustering.

We use metric *F* and P_k to evaluate the performances of *OIE*, *NECK* and two baselines in the process of named entity clustering and keyword generation respectively. We take the average of *F* and P_k for the clusters of all domains, and illustrate them in Figure 5.

As shown in Figure 5, *NECK* outperforms the other methods in *P*, *R*, *F* and P_k. For *F*, the big gap between *NECK*-A and *NECK*-W shows that named entity ambiguity affects the accuracy of results significantly. Comparing with *NECK*-A, *NECK* significantly improves *F* from 0.47% to 0.85 % in domain Politician and from 0.43% to 0.89% in domain Basketball player. The reason for the improvement is that human names are very ambiguous, and the irrelevant features of ambiguous human names lead to inferior clustering results. As the performance of clustering directly affects the performance of keyword generation phase, *NECK* maintains a substantial lead in the comparison with other algorithms in P_k.

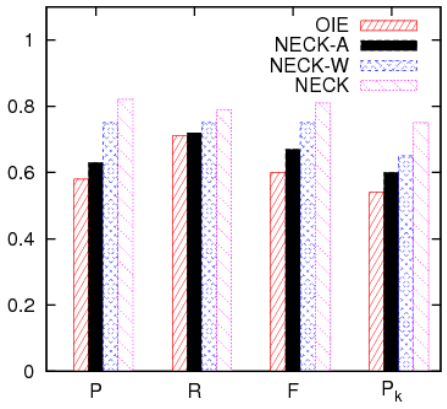

Figure 5. *Comparisons between NECK and other algorithms.*

Due to the limited space, we list the generated keywords of three clusters of named entities in domain "company" in Table 5. We can see from Table 5 that the keywords do reflect the semantic information of the named entities in each cluster. For example, cluster 1 is mainly about investment or finance companies, and the keywords are also related to the investment or financial business.

	Cluster 1	Cluster 2	Cluster 3
Keywords	# closed end funds, # financial services, # funds, # stock. http:// finance.yahoo.com	# cameras, # printer,# scanners, # copier, # driver ink. http://www.amazon.com http://www.ritzcamera.com	# inn, # hotels, # over georgia, # motels, # plaza resorts, # Orlando. http://travela.priceline.com
Named entities	citigroup, centerpoint, morgan stanley, north fork bank, smith barney, blackrock, t rowe price.	canon, olympus, casio, jvc, compaq, lexmark, eos, minolta, epson, ibm, pavilion, garmin, philips, zoom, toshiba.	best western, days inn, holiday inn, hotels com, hyatt, sea world, six flags, south beach, ramada, regency, hampton inn.

Table 5. *Three Sampled clusters of Domain* "*Company*"

5.3 Implementation and Efficiency

The proposed algorithm can be run very fast even with our limited hardware condition.

1, In the process of string matching between named entities and queries, we apply Aho-Corasick Algorithm (Aho & Corasick 1975). The time complexity of Aho-Corasick is linear with the character length of the lexical patterns plus the character length of the matched query plus the number of matched named entities.

2, In the calculation of the semantic similarity, we apply hash join to finding the intersection of the sets of contexts, URLs and sessions, in which we set the number of hash barrels as 1000.

3, The time complexity of the *HAC* algorithm and the graph ranking algorithm in the third phase of our approach are both $O(n^2)$. As click-through data is sparse, the complexity can reduce to $O(mn)$, in which n denotes the number of named entities and m denotes the maximum number of pairs of named entities between which the semantic similarity is not zero.

6 CONCLUSION AND FUTURE WORK

In this paper, we address the problem of clustering named entities from click-through data and generating keywords to interpret the semantic information of the clusters. As click-through data models the user space, which directly expresses users' searching behaviors, our approach can be essentially helpful in many web applications, such as general web search, product searching in consumer marketing and so on.

Comparing with existing works, the three-phase approach we propose solve the problem of named entity ambiguity, which leads to a substantial improvement in the accuracy of named entity clustering. The exploration of session ID enriches the feature space of named entities and the novel measurement we propose to weight each feature captures the importance of the feature in expressing the semantic information of named entities. What's more, the keyword generation phase is helpful for users to interpret the semantic information of corresponding clusters. Extensive experiments conducted shows the effectiveness of our approach.

The future work of our approach lies in two orientations. First, we plan to explore more features in click-through data, such as click model, bounce rate (Sculley et al. 2009) and so on. These features are effective in helping us in understanding users' behavior. Second, we plan to build a bridge between the web space and the user space. Although keywords generated relying on features do reflect the semantic information of named entity clusters, they need to be further summarized into a more brief and accurate manner in order to be more easily understood. For example, many existing knowledge base such as Wikipedia have a human-edited semantic label for each node, such as "singers" and so on. The way we choose to accomplish this goal in the future is to leverage existing knowledge bases without compromising the user space.

References

Aho, A. V. and Corasick, M. J. 1975. “Efficient String Matching: an Aid to Bibliographic Search,” Commun. ACM 18, 6, pp. 333-340.

Baeza-Yates, R. and Tiberi, A. 2007. “Extracting Semantic Relations from Query Logs,” in Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, pp. 76-85.

Bollegala, D., Matsuo, Y. and Ishizuka, M. 2007. “Measuring Semantic Similarity between Words using Web Search Engines,” in Proceedings of the 16th international conference on World Wide Web, ACM, New York, NY, USA, pp. 757-766.

Cao, H., Jiang, D., Pei, J., He, Q., Liao, Z., Chen,E. and Li, H. 2008. “Context-aware Query Suggestion by Mining Click-through and Session Data,” in Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, pp. 875-883.

Chuang, S. L. and Chien, L. F. 2002. “Towards Automatic Generation of Query Taxonomy: A Hierarchical Query Clustering Approach,” in Proceedings of the 2002 IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, USA, pp. 75-82.

Gabrilovich, E. and Markovitch, S. 2007. “Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis”, in Proceedings of the 20th international joint conference on Artifical intelligence, Rajeev Sangal, Harish Mehta, and R. K. Bagga (eds.), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1606-1611.

Guo, J., Xu G., Cheng, X. and Li, H. 2009. “Named Entity Recognition in Query,” in Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, pp. 267-274.

Jain, A. and Pennacchiotti, M. 2010. “Open Entity Extraction from Web Search Query Logs,” in Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics, Beijing, China, pp. 510-518.

Lee, U., Liu, Z. and Cho, J. 2005. “Automatic Identification of User Goals in Web Search,” in Proceedings of the 14th international conference on World Wide Web, ACM, New York, NY, USA, pp. 391-400.

Liu, J. and Birnbaum, L. 2007. “Measuring Semantic Similarity between Named Entities by Searching the Web Directory,” in Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society, Washington, DC, USA, pp. 461-465.

Mihalcea, R. and Tarau, P. (2004) “TextRank: Bringing Order into Texts,” in Proc. Of Conf. Empirical Methods in Natural Language Processing, D. Lin and D. Wu, eds., pp. 404-411, 2004.

Muñoz, A. 1996. Compound keyword generation from document databases using a hierarchical clustering ART model. Intelligent Data Analysis, 1(1).

Pantel, P. and Lin, D. 2002. “Discovering Word Senses from Text,” in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, pp. 613-619.

Pasca, Marius. 2007. “Weakly-supervised Discovery of Named Entities using Web Search Queries,” in Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, ACM, New York, NY, USA, pp. 683-690.

Pass, G., Chowdhury, A. and Torgeson, C. 2006. “A Picture of Search,” in Proceedings of the 1st international conference on Scalable information systems, ACM, New York, NY, USA, Article 1 .

Sculley, D., Malkin, R. G., Basu, S. and Bayardo, R. J. (2009) “Predicting bounce rates in sponsored search advertisements.” Knowledge Discovery and Data Mining, pp. 1325-1334.

Shrout, P. and Fleiss, J. L. (1979) "Intraclass Correlation: Uses in Assessing Rater Reliability," in Psychological Bulletin. Vol. 86, No. 2, pp. 420–428.

Steier, A. M. and Belew, R. K, 1993. “Exporting phrases: A statistical analysis of topical language,” in Proceedings of Second Symposium on Document Analysis and Information Retrieval, pp. 179-190.

Wan, X. and Yang, J. and Xiao, J. (2007) “Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction,” in Meeting of the Association for Computational Linguistics. 2007.

Yin, X. and Shah, S. 2010. “Building Taxonomy of Web Search Intents for Name Entity Queries,” in Proceedings of the 19th international conference on World Wide Web, ACM, New York, NY, USA, pp. 1001-1010.