Heterogeneous Graph-Based Intent Learning with Queries, Web Pages and Wikipedia Concepts

Xiang Ren[†*] Yujing Wang[‡] Xiao Yu[†] Jun Yan[‡] Zheng Chen[‡] Jiawei Han[†]
[†] University of Illinois at Urbana-Champaign, Urbana, IL, USA
[‡] Microsoft Research, Beijing, China
[†]{xren7, xiaoyu, hanj}@illinois.edu [‡]{yujwang, junyan, zhengc}@microsoft.com

ABSTRACT

The problem of learning user search intents has attracted intensive attention from both industry and academia. However, state-of-the-art intent learning algorithms suffer from different drawbacks when only using a single type of data source. For example, query text has difficulty in distinguishing ambiguous queries; search log is bias to the order of search results and users' noisy click behaviors. In this work, we for the first time leverage three types of objects, namely queries, web pages and Wikipedia concepts collaboratively for learning generic search intents and construct a heterogeneous graph to represent multiple types of relationships between them. A novel unsupervised method called *heterogeneous graph-based soft-clustering* is developed to derive an intent indicator for each object based on the constructed heterogeneous graph. With the proposed co-clustering method, one can enhance the quality of intent understanding by taking advantage of different types of data, which complement each other, and make the implicit intents easier to interpret with explicit knowledge from Wikipedia concepts. Experiments on two real-world datasets demonstrate the power of the proposed method where it achieves a 9.25% improvement in terms of NDCG on search ranking task and a 4.67% enhancement in terms of Rand index on object co-clustering task compared to the best state-of-the-art method.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*clustering, search process*

General Terms

Algorithms; Experimentation

Keywords

Search Intent; Wikipedia; Heterogeneous Graph Clustering

*This work was partially done when the first author was visiting Microsoft Research, Beijing, China.

WSDM'14, February 24–28, 2014, New York, New York, USA.

http://dx.doi.org/10.1145/2556195.2556222.

1. INTRODUCTION

Understanding user search intent is one of the most fundamental and critical problems faced by modern search engines. Research studies have been conducted extensively on identifying search intents in order to better satisfy users' information needs. Identifying the intents behind a query helps distinguish ambiguous or multi-faceted queries [16], suggest related queries [30], and guide users to corresponding vertical search engines [15]. Additionally, matching web pages and search intents together can benefit search result presentation through search result diversification.

There are mainly two categories of intent definitions, based on whether or not the intent is pre-defined. Intents such as search tasks [17] require intent labels defined in advance while others such as topics [1] do not rely on any existing ontology. In this work we focus on learning generic search intents which can fit into all domains without any kind of pre-defined intent category.

Traditional intent learning methods utilize only text information from queries or web pages [29]. Such methods cannot handle ambiguous or multi-faceted queries properly. Later studies leverage web search log to facilitate user interest interpretation when learning search intents [7]. One general assumption is that multiple web pages tend to share the same search intent if they are clicked under similar queries. Unfortunately, this assumption has some intrinsic limitation due to the nature of web search. User clicks might be biased by the order of search results, and are typically observed for only a small fraction of web pages. Users may click web pages which are irrelevant to their real search intents by mistake, bringing noise to the search log.

Recent studies attempt to tackle the aforementioned issues by either using search session data [24] and sponsored search data [27], or combining search log with different data sources such as query text [20, 16] and search session data [22]. In particular, semantic information such as named entities [15] and Wikipedia concepts [28, 4] have proven to be effective in terms of learning and interpreting search intents. However, existing methods can only learn intents within a specific domain such as cars or movies, and often can only handle concept-related queries. To our best knowledge, no previous works leverage query, web page and Wikipedia concepts collaboratively to learn generic search intents.

In this paper we propose a novel search intent definition by utilizing three types of objects, namely query, web page and semantic concept from Wikipedia. A heterogeneous graph-based framework is proposed to derive intents from query text data, web page content, click-through log and

Wikipedia concepts simultaneously. The unified optimization approach takes advantages of different types of data, which complement each other, and ameliorate the aforementioned drawbacks when using a single data source. For example, given the query "office", the proposed method is able to identify several search intents including software intent for Microsoft Office, entertainment intent for the NBC TV show "The Office" and daily supplies intent for office supplies as needed. However, using only query text, entity mentions in query or the search log we are not able to find all possible intents as stated above because (1) search queries are often short and ambiguous; and (2) user click is biased to more popular web pages related to Microsoft Office.

In our solution, we construct a heterogeneous graph using many types of data to encode query-page, page-page and page-concept relations where the relation strength indicates how similar two objects are, in terms of their search intents (see Figure 1). To learn search intents through the heterogeneous graph structure, we face three challenges: (1) how to model the relationships between objects to preserve the semantic information in the constructed heterogeneous graph; (2) how to incorporate different types of relations in a unified intent learning framework; (3) how to predict intents for newly emerged objects. We develop a graph-based soft-clustering method to learn how likely an object belongs to each search intent from the heterogeneous graph, yielding an intent indicator for each object. An approximate algorithm is further designed to first embed all types of objects into a unified feature space and then apply soft-clustering methods on the embedded intent features to obtain intent indicators. With the proposed method, we can simultaneously (1) identify information needs behind queries; (2) understand what intents a web page can satisfy; and (3) recognize Wikipedia concepts that can best interpret different intents.

Experiments on both search ranking, which evaluates the effectiveness of intent features, and object co-clustering of queries, web pages and concepts demonstrate the power of our method. For search ranking, the proposed method achieves a 9.35% improvement in terms of NDCG on the commercial data set and a 23.99% growth in terms of MRR on AOL data set. For object co-clustering, our method gets a 4.67% enhancement in terms of Rand index on the commercial data set compared to the best state-of-the-art method. Our case study on object co-clustering results shows that the learned intents can be well interpreted by the Wikipedia concepts within each intent cluster.

The rest of paper is organized as follows. Section 3 introduces the construction of heterogeneous graphs and provides a formal problem definition. Section 4 proposes the heterogeneous graph-based soft-clustering method, along with the computational complexity analysis. We provide experimental results and analysis in Section 5, and conclude our work together with ideas on future work in Section 6.

2. RELATED WORK

2.1 Query Intent Learning

The problem of learning query intent has been studied intensively and characterized along different dimensions, such as search subgoals [27], search tasks [17], search missions [1], topics [1], subtopics [16], patterns [6] and taxonomy [28]. Nevertheless, we categorize existing methods into query classification and query clustering.

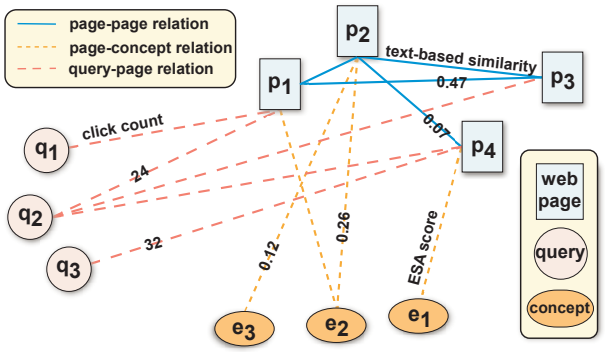


Figure 1: An illustration of the constructed heterogeneous graph which consists of three types of objects: queries, web pages and Wikipedia concepts, and three types of relations: query-page relation, page-page relation and page-concept relation.

Query classification aims to classify queries into pre-defined intents such as search goals or search tasks [17]. Instead of using only query keywords, quite a few studies turn to different resources, including search log [18], Wikipedia concepts [15] and search session data [5].

As an alternative, grouping queries into intents by means of query clustering has also been popularly studied [24, 28, 20, 1, 16, 27, 6]. Traditional document clustering methods cannot work well on queries due to their limited number of keywords. Search log which contains user interests is one kind of data widely used for query clustering [7], and is further combined with query content to achieve better performance [20]. Search session data [30] and sponsored search data [27] are also widely utilized for finding query relations and intents.

Our work is similar to [6] in terms of the motivation of seeking a better interpretation for learned intents. However, we achieve the interpretation by leveraging semantic concepts in Wikipedia. Compared to building taxonomy from entity-related queries [28], we collect external semantic concepts from Wikipedia, which is able to handle broader kinds of queries. Although there are studies such as [25], which learn user intents for queries and web pages simultaneously, and work such as [4], which explores heterogeneous relationships between queries and Wikipedia concepts, to our best knowledge, our work is the first for learning generic search intents behind queries, web pages and Wikipedia concepts in an unified graph-based framework.

2.2 Clustering on Graphs

Another group of related work is graph-based clustering which groups single or multiple types of objects with respect to their graph structure. Typical methods include spectral clustering on homogeneous graph and co-clustering on bipartite graph [8, 26].

In particular, Guan *et al.* [12] exploit affinity and bipartite relationships between users, documents and social tags by graph embedding in recommendation. Recently, several studies exploit heterogeneous graphs which consist of heterogeneous types of objects and/or relationships by clustering [19]. For example, affinity relationships can be incorporated into co-clustering of bipartite graph [11]. Clustering of multiple types of objects in a heterogeneous graph which consists of multiple relationships are also studied [23].

The differences between our method and the above studies are (1) we study both affinity relationships and bipartite relationships in a co-clustering framework, and (2) we consider features of multiple types of objects simultaneously.

3. BACKGROUND

In this section, we provide the background of the problem which includes the details on construction of heterogeneous graph and the formal problem definition.

3.1 Heterogeneous Graph Construction

We use a heterogeneous graph, which consists of multiple types of objects and/or multiple types of relationships, to preserve different kinds of information from different data sources. The basic idea to construct the graph is that two objects are linked with larger relation strength if they are more likely to share similar search intent.

Specifically, we have three types of objects: queries $\mathcal{Q} = \{q_1, ..., q_{|\mathcal{Q}|}\}$ issued by different users in search log, web pages $\mathcal{P} = \{p_1, ..., p_{|\mathcal{P}|}\}$ clicked by different users in search log, and Wikipedia concepts $\mathcal{E} = \{e_1, ..., e_{|\mathcal{E}|}\}$ related to web pages[1]. There are three types of relations between these objects including query-page relation extracted from click-through log, page-page relation based on web page text, and page-concept relation provided by explicit semantic analysis (ESA) [9], leading to three subgraphs $H_{\mathcal{Q},\mathcal{P}}$, $G_{\mathcal{P}}$ and $J_{\mathcal{P},\mathcal{E}}$, respectively. In this way, information from these data sources can be learned to preserve the heterogeneous graph structure. We denote the constructed heterogeneous graph as G and show an illustration in Figure 1.

3.1.1 Query-page Subgraph

Click-through relations between queries and web pages form a bipartite query-page subgraph. We assume that a query and a web page are more likely to share the same intent if the number of users who click the web page after issuing the query is larger. For example, if most of users end at clicking web pages of Microsoft Office after they issue the query "office", "office" is more likely to carry software intent rather than intents relevant to entertainment.

The query-web page bipartite subgraph is also called *click graph* in the literature [7], where a query $q \in \mathcal{Q}$ is linked to a web page $p \in \mathcal{P}$ if and only if p has been clicked by at least one user after issuing query q in the search log. The edge weight is defined as click-through counts between q and p, reflecting the relation strength. We use $\mathbf{C} \in \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{P}|}$ to denote adjacency matrix of the click graph, where C_{ij} is equal to click counts from i-th query to j-th web page.

3.1.2 Web Page Subgraph

Text in a web page is a good source for identifying what search intents a web page can satisfy. For instance, if we find a web page containing keywords (*e.g.*, words with highest TF-IDF scores) such as "Microsoft" , "install" and "download", we are more confident to say this web page can accomplish software intent rather than entertainment intent. Therefore, we assume that two web pages might satisfy similar search intents if their textual content are similar.

Specifically, we extract keywords from title and body of all web pages based on TF-IDF scores which form an m-dimensional term space, and generate TF-IDF vectors for all the web pages as their textual features, denoted by $\mathbf{P} \in \mathbb{R}^{|\mathcal{P}|\times m}$. We further follow the construction of K-nearest neighbor (KNN) graph [2] to build a web page affinity subgraph whose adjacency matrix is denoted by $\mathbf{Y} \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{P}|}$. Specifically, each web page is linked to its K most similar web pages in terms of their textual features $\mathbf{P}$ as follows:

$$Y_{ij} = \begin{cases} \text{sim}(\mathbf{P}_i, \mathbf{P}_j), & \text{if } \mathbf{P}_i \in N_K(\mathbf{P}_j) \text{ or } \mathbf{P}_j \in N_K(\mathbf{P}_i); \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where $\text{sim}(\mathbf{P}_i, \mathbf{P}_j)$ is Cosine similarity score between $\mathbf{P}_i$ and $\mathbf{P}_j$, *i.e.*, $\text{sim}(\mathbf{P}_i, \mathbf{P}_j) = (\mathbf{P}_i \cdot \mathbf{P}_j^T)/(\|\mathbf{P}_i\|\|\mathbf{P}_j\|)$, and $N_K(\mathbf{a})$ denotes K nearest neighbors of $\mathbf{a}$.

3.1.3 Page-concept Subgraph

We extract web page-concept relations using explicit semantic analysis (ESA) [9], which can measure the semantic relevance between a Wikipedia concept and a given web page. For instance, given the web page about Microsoft Office, Wikipedia concept "microsoft_word" receives ESA score as 0.24 while "room_(architecture)" gets approximately 0 score, showing that "microsoft_word" is more likely to share the same intent with the web page. Similarly, two web pages are assumed more likely to share the same search intent if their related concepts are similar.

For simplicity, we only consider the top-300 most relevant concepts for each web page, and represent all web pages in the resulting $|\mathcal{E}|$-dimension concept space as ESA score vectors, denoted as $\mathbf{P}_{\mathcal{E}} \in \mathbb{R}^{(|\mathcal{P}|\times|\mathcal{E}|)}$. We further define the page-concept bipartite subgraph $J_{\mathcal{P},\mathcal{E}}$ by an adjacency matrix $\mathbf{P}_{\mathcal{E}} \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{E}|}$ as follows:

$$P_{\mathcal{E},ij} = \begin{cases} \text{sim}(p_i, e_j), & \text{if } p_i \in N_K(e_j) \text{ or } e_j \in N_K(p_i); \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where $\text{sim}(p_i, e_j)$ denotes the ESA scores between web page p_i and concept e_j; and $\sum_{j=1}^{|\mathcal{E}|} P_{\mathcal{E},ij} = 1$.

3.2 Problem Definition

In this work, we cast the search intent learning problem into heterogeneous graph-based soft-clustering of queries, web pages and concepts simultaneously where each cluster can be explained as one search intent.

Suppose there are k different search intents behind all three types of objects, we first define three indicator matrices $\mathbf{F}_{\mathcal{Q}} \in [0,1]^{|\mathcal{Q}|\times k}$, $\mathbf{F}_{\mathcal{P}} \in [0,1]^{|\mathcal{P}|\times k}$, and $\mathbf{F}_{\mathcal{E}} \in [0,1]^{|\mathcal{E}|\times k}$ which describe the confidence of queries, web pages and concepts belonging to different search intents, respectively. The definition of search intent learning problem is as follows:

DEFINITION 1 (PROBLEM DEFINITION). *Suppose there are k distinct search intents for queries, web pages and concepts. Given a constructed heterogeneous graph G, its adjacency matrices $\mathbf{C}$, $\mathbf{Y}$ and $\mathbf{P}_{\mathcal{E}}$, and web pages' textual features $\mathbf{P}$, learn $\mathbf{F}_{\mathcal{Q}}$, $\mathbf{F}_{\mathcal{P}}$ and $\mathbf{F}_{\mathcal{E}}$ as soft-clustering indicators for all three types of objects simultaneously.*

4. LEARNING INTENTS ON GRAPHS

In this section, we explain our method for learning unified search intents in details. An approximate method for efficient learning of concept intent is first developed. We then derive an optimization problem which preserve the heterogeneous graph structure and developed an efficient algorithm

[1] In the rest of the paper, we use the term "concept" to represent Wikipedia concept for simplicity.

based on graph embedding and fuzzy clustering for solving the optimization. Finally, we extend our method to predict intent for unseen queries and web pages.

4.1 Efficient Learning of Concept Intents

In practice, the concept number in knowledge base is huge (*e.g.*, Wikipedia contains 4,308,113 English concepts[2]), resulting in an even larger parameter space when learning concept intent indicator $\mathbf{F}_{\mathcal{E}}$, *i.e.*, $\mathcal{O}(|\mathcal{E}|d)$. Therefore, we propose to approximately compute concept intent features by first aggregating information from web page-concept subgraph into an concept-based web page affinity subgraph, and then deriving concept intent indicator based on learned web page intent indicator $\mathbf{F}_{\mathcal{P}}$ and web page-concept subgraph $\mathbf{P}_{\mathcal{E}}$. Through this way, the computational burden is largely reduced by getting rid of the parameter space of $\mathbf{F}_{\mathcal{E}}$ during the learning process.

Specifically, web page-concept relationships in the bipartite subgraphs $J_{\mathcal{P},\mathcal{E}}$ are encoded into concept-based web page relationships by measuring web page-web page similarity in terms of their related concepts $\mathbf{P}_{\mathcal{E}i}$ and $\mathbf{P}_{\mathcal{E}j}$, respectively. We represent $J_{\mathcal{P}}$ by an adjacency matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{P}|}$ whose element is dot product of $\mathbf{P}_{\mathcal{E}i}$ and $\mathbf{P}_{\mathcal{E}j}$, *i.e.*, $E_{ij} = \mathbf{P}_{\mathcal{E}i}\mathbf{P}_{\mathcal{E}j}^T$. With this modification, the heterogenous graph G is now transformed into a simplified formation, consisting of a click graph, a text-based affinity graph of web pages, and an concept-based affinity graph for web pages. After learning the web page intent features, we can further derive the intent feature for j-th concept as follows:

$$\mathbf{F}_{\mathcal{E}j} = \mathbf{P}_{\mathcal{E}\cdot j}^T \cdot \mathbf{F}_{\mathcal{P}}, \quad \text{for} \quad j = 1, 2, \ldots, |\mathcal{E}|, \tag{3}$$

where $\mathbf{P}_{\mathcal{E}\cdot j}$ is the j-th column of matrix $\mathbf{P}_{\mathcal{E}}$.

In the following sections, we will introduce the detailed algorithm for learning intent features of queries and web pages based on the simplified heterogeneous graph G.

4.2 Problem Formulation

In this section, a heterogeneous graph-based soft-clustering optimization problem is derived for unified search intent learning and an approximate algorithm is developed to efficiently solve the optimization problem.

Mathematically, we model each type of relationship into a cost function based on locality preserving idea [2], which requires two nearby objects in G to have similar intent indicators, and further derive the optimization problem by using weighted summation of these cost functions as the objective function and imposing soft-clustering constraints on the intent indicators:

$$\begin{aligned}
&\min_{\mathbf{F}_{\mathcal{Q}},\mathbf{F}_{\mathcal{P}}} \mathcal{L}(\mathbf{F}_{\mathcal{Q}},\mathbf{F}_{\mathcal{P}}) = \lambda_{\mathcal{QP}} \cdot \sum_{i=1}^{|\mathcal{Q}|}\sum_{j=1}^{|\mathcal{P}|} C_{ij}\|\mathbf{F}_{\mathcal{Q}i} - \mathbf{F}_{\mathcal{P}j}\|_2^2 \\
&+\lambda_T \cdot \sum_{i,j=1}^{|\mathcal{P}|} Y_{ij}\|\mathbf{F}_{\mathcal{P}i} - \mathbf{F}_{\mathcal{P}j}\|_2^2 + \lambda_{\mathcal{E}} \cdot \sum_{i,j=1}^{|\mathcal{P}|} E_{ij}\|\mathbf{F}_{\mathcal{P}i} - \mathbf{F}_{\mathcal{P}j}\|_2^2 \\
&\text{s.t.} \quad \mathbf{F}_{\mathcal{Q}} \in [0,1]^{|\mathcal{Q}|\times k}, \quad \mathbf{F}_{\mathcal{P}} \in [0,1]^{|\mathcal{P}|\times k},
\end{aligned} \tag{4}$$

where $\|\cdot\|_2$ denotes the L_2 norm, *i.e.*, $\|\mathbf{a}\|_2^2 = \sum_i a_i^2$. $\lambda_{\mathcal{QP}}$, λ_T and $\lambda_{\mathcal{E}}$ are tuning parameters which control the trade-off between three types of relationships. Note that $0 \leq \lambda_{\mathcal{QP}}, \lambda_T, \lambda_{\mathcal{E}} \leq 1$.

[2]http://en.wikipedia.org/wiki/Wikipedia:Statistics

Query-web page relationship is modeled by the first term in Equation (4). Minimizing the term forces linked queries and web pages in click graph $\mathbf{C}$ to have similar intent indicator if click count between them is large. With the definition of two diagonal degree matrices $\mathbf{D}^{(\mathcal{P})}$ and $\mathbf{D}^{(\mathcal{Q})}$, where $D_{ii}^{(\mathcal{Q})} = \sum_{j=1}^{|\mathcal{P}|} C_{ij}$ and $D_{jj}^{(\mathcal{P})} = \sum_{i=1}^{|\mathcal{Q}|} C_{ij}$, we can rewrite L_2 norm of the first term into trace form and transform summation into matrix multiplication as follows:

$$\begin{aligned}
&\sum_{i=1}^{|\mathcal{Q}|}\sum_{j=1}^{|\mathcal{P}|} C_{ij} \cdot \|\mathbf{F}_{\mathcal{Q}i} - \mathbf{F}_{\mathcal{P}j}\|_2^2 \\
=\ & \mathrm{Tr}\Big(\sum_{i=1}^{|\mathcal{Q}|} D_{ii}^{(\mathcal{Q})} \cdot \mathbf{F}_{\mathcal{Q}i}\mathbf{F}_{\mathcal{Q}i}^T + \sum_{j=1}^{|\mathcal{P}|} D_{jj}^{(\mathcal{P})} \cdot \mathbf{F}_{\mathcal{P}j}\mathbf{F}_{\mathcal{P}j}^T \\
& -2 \cdot \sum_{i=1}^{|\mathcal{Q}|}\sum_{j=1}^{|\mathcal{P}|} C_{ij} \cdot \mathbf{F}_{\mathcal{Q}i}\mathbf{F}_{\mathcal{P}j}^T\Big) \\
=\ & \mathrm{Tr}\Big(\mathbf{F}_{\mathcal{Q}}^T\mathbf{D}^{(\mathcal{Q})}\mathbf{F}_{\mathcal{Q}} + \mathbf{F}_{\mathcal{P}}^T\mathbf{D}^{(\mathcal{P})}\mathbf{F}_{\mathcal{P}} - 2 \cdot \mathbf{F}_{\mathcal{Q}}^T\mathbf{C}\mathbf{F}_{\mathcal{P}}\Big).
\end{aligned} \tag{5}$$

Here we utilize two properties of trace: $\mathrm{Tr}(\mathbf{A}+\mathbf{B}) = \mathrm{Tr}(\mathbf{A}) + \mathrm{Tr}(\mathbf{B})$ and $\mathrm{Tr}(\mathbf{AB}) = \mathrm{Tr}(\mathbf{BA})$ [10].

The second term in Equation (4) attempts to model web page-web page relationships, which forces intent indicators of two web pages are similar if they are close to each other in terms of web page affinity subgraph $\mathbf{Y}$. Similarly, we can rewrite it into trace form as follows:

$$\begin{aligned}
&\sum_{i,j=1}^{|\mathcal{P}|} Y_{ij} \cdot \|\mathbf{F}_{\mathcal{P}i} - \mathbf{F}_{\mathcal{P}j}\|_2^2 \qquad (6)\\
=\ & 2\,\mathrm{Tr}\Big(\mathbf{F}_{\mathcal{P}}^T\mathbf{D}^{(T)}\mathbf{F}_{\mathcal{P}} - \mathbf{F}_{\mathcal{P}}^T\mathbf{Y}\mathbf{F}_{\mathcal{P}}\Big) = 2\,\mathrm{Tr}\Big(\mathbf{F}_{\mathcal{P}}^T\mathbf{L}^{(T)}\mathbf{F}_{\mathcal{P}}\Big),
\end{aligned}$$

where $\mathbf{D}^{(T)}$ is defined as $D_{ii}^{(T)} = \sum_{j=1}^{|\mathcal{P}|} Y_{ij}$ and $\mathbf{L}^{(T)} = \mathbf{D}^{(T)} - \mathbf{Y}$ is the graph Laplacian matrix of $\mathbf{Y}$ [2].

As described in Section 4.1, web page-concept relationships have been transformed into concept-based relationships between web pages. Minimizing the third term in Equation (4) forces two web pages to have similar intent indicators if concept-based relationship between them is strong. Similarly we can define degree matrix $\mathbf{D}^{(\mathcal{E})}$ where $D_{ii}^{(\mathcal{E})} = \sum_{j=1}^{|\mathcal{P}|} E_{ij}$ and $\mathbf{L}^{(\mathcal{E})} = \mathbf{D}^{(\mathcal{E})} - \mathbf{E}$, and rewrite the term as follows:

$$\sum_{i,j=1}^{|\mathcal{P}|} E_{ij} \cdot \|\mathbf{F}_{\mathcal{P}i} - \mathbf{F}_{\mathcal{P}j}\|_2^2 = 2 \cdot \mathrm{Tr}\Big(\mathbf{F}_{\mathcal{P}}^T\mathbf{L}^{(\mathcal{E})}\mathbf{F}_{\mathcal{P}}\Big). \tag{7}$$

By substituting Equations (5), (6) and (7) into the objective function of Equation (4), the graph-based soft-clustering optimization problem is transformed equivalently as follows:

$$\begin{aligned}
&\min_{\mathbf{F}_{\mathcal{Q}},\mathbf{F}_{\mathcal{P}}} \mathcal{L}(\mathbf{F}_{\mathcal{Q}},\mathbf{F}_{\mathcal{P}}) \\
=\ & \lambda_{\mathcal{QP}} \cdot \mathrm{Tr}\Big(\mathbf{F}_{\mathcal{Q}}^T\mathbf{F}_{\mathcal{Q}} + \mathbf{F}_{\mathcal{P}}^T\mathbf{F}_{\mathcal{P}} - 2 \cdot \mathbf{F}_{\mathcal{Q}}^T\mathbf{C}\mathbf{F}_{\mathcal{P}}^T\Big) \\
& +2\lambda_T \cdot \mathrm{Tr}\Big(\mathbf{F}_{\mathcal{P}}^T\mathbf{L}^{(T)}\mathbf{F}_{\mathcal{P}}\Big) + 2\lambda_{\mathcal{E}} \cdot \mathrm{Tr}\Big(\mathbf{F}_{\mathcal{P}}^T\mathbf{L}^{(\mathcal{E})}\mathbf{F}_{\mathcal{P}}\Big) \\
&\text{s.t.} \quad \mathbf{F}_{\mathcal{Q}} \in [0,1]^{|\mathcal{Q}|\times k}, \quad \mathbf{F}_{\mathcal{P}} \in [0,1]^{|\mathcal{P}|\times k},
\end{aligned} \tag{8}$$

With the definition of augmented intent indicator $\mathbf{F} = [\mathbf{F}_{\mathcal{Q}}^T, \mathbf{F}_{\mathcal{P}}^T]^T$, Equation 8 can be further rewritten into a more concise form:

$$\min_{\mathbf{F}} \mathcal{L}(\mathbf{F}) = \mathrm{Tr}(\mathbf{F}^T\mathbf{L}\mathbf{F}) \ \text{ s.t. } \ \mathbf{F} \in [0,1]^{(|\mathcal{Q}|+|\mathcal{P}|)\times k}, \tag{9}$$

where $\mathbf{L}$ is the global graph Laplacian matrix:

$$\mathbf{L} = \begin{bmatrix} \lambda_{\mathcal{QP}} \cdot \mathbf{D}^{(\mathcal{Q})} & -\lambda_{\mathcal{QP}} \cdot \mathbf{C} \\ -\lambda_{\mathcal{QP}} \cdot \mathbf{C}^T & \lambda_{\mathcal{QP}}\mathbf{D}^{(\mathcal{P})} + 2\lambda_T\mathbf{L}^{(T)} + 2\lambda_{\mathcal{E}}\mathbf{L}^{(\mathcal{E})} \end{bmatrix}. \tag{10}$$

4.3 The Learning Algorithm

Directly solving the soft clustering problem in Equation (9) is not easy, since the objective function is non-convex and non-continuous. Similar to spectral clustering [8], we propose an efficient algorithm to approximately solve Equation (9) which first embeds each object into a k-dimensional intent space, and then clusters the objects based on the embedding intent features.

First, by relaxing $\mathbf{F}$ to $\mathbf{U} \in \mathbb{R}^{(|\mathcal{P}|+|\mathcal{Q}|)\times k}$ and imposing constraint on $\mathbf{U}$, we can learn an optimal graph embedding $\mathbf{U}$ similar to Laplacian eigenmaps [2], which encodes all types of relationships into a k-dimensional intent space as follows:

$$\min_{\mathbf{U}} \operatorname{Tr}(\mathbf{U}^T\mathbf{L}\mathbf{U}) \quad \text{s.t.} \quad \mathbf{U}^T\mathbf{M}\mathbf{U} = \mathbf{I}_k, \tag{11}$$

where $\mathbf{I}_k$ is identity matrix and $\mathbf{M}$ is global degree matrix defined as

$$\mathbf{M} = \begin{bmatrix} \lambda_{\mathcal{QP}} \cdot \mathbf{D}^{(\mathcal{Q})} & 0 \\ 0 & \lambda_{\mathcal{QP}}\mathbf{D}^{(\mathcal{P})} + 2\lambda_T\mathbf{D}^{(T)} + 2\lambda_{\mathcal{E}}\mathbf{D}^{(\mathcal{E})} \end{bmatrix}. \tag{12}$$

The constraint $\mathbf{U}^T\mathbf{M}\mathbf{U} = \mathbf{I}_k$ in Equation (11) is imposed to remove arbitrary scaling of $\mathbf{U}$ where the degree matrix $\mathbf{M}$ provides a global measure on variance of $\mathbf{U}$ [2]. From Equation (11) we can see $\mathbf{L}$ is a positive semi-definite matrix. Optimal intent features $\tilde{\mathbf{U}}$ can be computed by k generalized eigenvectors corresponding to k-smallest eigenvalues of the generalized eigenvalue problem $\mathbf{L}\mathbf{U} = \lambda\mathbf{M}\mathbf{U}$ (excluding the one with zero eigenvalue). In particular, we will show shortly (see Section 4.5) that Lanczos algorithm [10] can be used to iteratively compute the smallest k generalized eigenvectors, which largely increase the efficiency by avoiding solving all eigenvalues.

With intent features $\tilde{\mathbf{U}} = [\tilde{\mathbf{U}}_{\mathcal{Q}}^T, \tilde{\mathbf{U}}_{\mathcal{P}}^T]^T$ for both queries and web pages, we derive intent features for concepts based on Equation (3), *i.e.*, $\tilde{\mathbf{U}}_{\mathcal{E}k} = \mathbf{P}_{\mathcal{E}\cdot k}^T \cdot \tilde{\mathbf{U}}_{\mathcal{P}}$, for $k = 1, 2, \ldots, |\mathcal{E}|$. Therefore, various soft clustering methods can be adopted on $\tilde{\mathbf{U}}_{\mathcal{Q}}$, $\tilde{\mathbf{U}}_{\mathcal{P}}$ and $\tilde{\mathbf{U}}_{\mathcal{E}}$ to learn search intent indicators $\mathbf{F}_{\mathcal{Q}}$, $\mathbf{F}_{\mathcal{P}}$ and $\mathbf{F}_{\mathcal{E}}$ simultaneously. In this work, we adopt the widely known fuzzy c-means [3], which optimizes the intent features as follows:

$$\{\tilde{\mathbf{O}}, \tilde{\mathbf{\Theta}}\} = \operatorname*{argmin}_{\mathbf{O},\mathbf{\Theta}} \sum_{i=1}^{|\mathcal{Q}|+|\mathcal{P}|+|\mathcal{E}|} \sum_{j=1}^{k} O_{ij}^2 \cdot \|\tilde{\mathbf{\Pi}}_i - \mathbf{\Theta}_j\|_2^2 \tag{13}$$

where $\tilde{\mathbf{\Pi}} = [\tilde{\mathbf{U}}_{\mathcal{Q}}^T, \tilde{\mathbf{U}}_{\mathcal{P}}^T, \tilde{\mathbf{U}}_{\mathcal{E}}^T]^T$ is augmented intent features for all objects and $\mathbf{\Theta}_j$ is the center of j-th cluster. Details of the fuzzy c-means algorithm is in [3]. Finally, we can obtain the intent indicators for queries, web pages and concepts by $\tilde{\mathbf{O}} = [\tilde{\mathbf{F}}_{\mathcal{Q}}^T, \tilde{\mathbf{F}}_{\mathcal{P}}^T, \tilde{\mathbf{F}}_{\mathcal{E}}^T]^T \in [0,1]^{(|\mathcal{Q}|+|\mathcal{P}|+|\mathcal{E}|)\times k}$.

4.4 Generalization for Intent Prediction

In practice, users continuously issue new queries and look for new web pages to satisfy their search requirements. Thus, we desire an algorithm with generalization ability which can predict search intents for unseen queries and web pages. In this section, we propose a modified algorithm for newly emerged objects based on the linear model, which is closely related to locality preserving projection [14].

Algorithm 1 Unified Search Intent Learning by HSoC

Input: Adjacency matrices $\{\mathbf{C}, \mathbf{Y}, \mathbf{P}_{\mathcal{E}}\}$, textual features $\{\mathbf{Q}, \mathbf{P}\}$, parameters $\{\lambda_{\mathcal{QP}}, \lambda_T, \lambda_{\mathcal{E}}\}$, number of intents k
Output: Intent features $\{\tilde{\mathbf{U}}_{\mathcal{Q}}, \tilde{\mathbf{U}}_{\mathcal{P}}, \tilde{\mathbf{U}}_{\mathcal{E}}\}$, intent indicators $\{\tilde{\mathbf{F}}_{\mathcal{Q}}, \tilde{\mathbf{F}}_{\mathcal{P}}, \tilde{\mathbf{F}}_{\mathcal{E}}\}$,

1: Derive adjacency matrix of concept-based affinity subgraph $\mathbf{E}$ based on $E_{ij} = \mathbf{P}_{\mathcal{E}i}\mathbf{P}_{\mathcal{E}j}^T$
2: Calculate matrices $\tilde{\mathbf{L}} = \mathbf{X}^T\mathbf{L}\mathbf{X}$ and $\tilde{\mathbf{M}} = \mathbf{X}^T\mathbf{M}\mathbf{X}$ using $\mathbf{X} = \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{P} \end{bmatrix}$, $\mathbf{L}$ defined in Equation (10), and $\mathbf{M}$ defined in Equation(12)
3: Use Lanczos algorithm to get k generalized eigenvectors $\tilde{\boldsymbol{\beta}}$ corresponding to k smallest eigenvalues (excluding 0) of eigen-problem in Equation (15)
4: Compute intent features $\tilde{\mathbf{U}}_{\mathcal{Q}}$, $\tilde{\mathbf{U}}_{\mathcal{P}}$ and $\tilde{\mathbf{U}}_{\mathcal{E}}$ based on Equations (14) and (3), respectively
5: Use fuzzy c-means in Equation (13) to learn intent indicators $\tilde{\mathbf{F}}_{\mathcal{Q}}$, $\tilde{\mathbf{F}}_{\mathcal{P}}$ and $\tilde{\mathbf{F}}_{\mathcal{E}}$

Specifically, we impose linear models on intent features $\mathbf{U} = [\mathbf{U}_{\mathcal{Q}}^T, \mathbf{U}_{\mathcal{P}}^T]^T$ as follows:

$$\mathbf{U}_{\mathcal{Q}} = \mathbf{Q}\mathbf{W} \quad \text{and} \quad \mathbf{U}_{\mathcal{P}} = \mathbf{P}\mathbf{V}, \tag{14}$$

where $\mathbf{W} \in \mathbb{R}^{|\mathcal{Q}|\times m}$ and $\mathbf{V} \in \mathbb{R}^{|\mathcal{P}|\times n}$ are parameters for the linear models. $\mathbf{Q} \in \mathbb{R}^{|\mathcal{Q}|\times n}$ is the term feature vector (weighted by TF-IDF) for queries and n is the dimensionality of query term space. Suppose we have $\boldsymbol{\beta} = [\mathbf{W}^T, \mathbf{V}^T]^T$ as augmented parameters, $\mathbf{X} = \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & \mathbf{P} \end{bmatrix}$ as augmented content feature, $\tilde{\mathbf{L}} = \mathbf{X}^T\mathbf{L}\mathbf{X}$ as global graph Laplacian and $\tilde{\mathbf{M}} = \mathbf{X}^T\mathbf{M}\mathbf{X}$ as global degree matrix, by substituting Equation (14) into Equation (11), a linear extension of originally intent learning problem can be derived as follows:

$$\min_{\boldsymbol{\beta}} \operatorname{Tr}(\boldsymbol{\beta}^T\tilde{\mathbf{L}}\boldsymbol{\beta}) \quad \text{s.t.} \quad \boldsymbol{\beta}^T\tilde{\mathbf{M}}\boldsymbol{\beta} = \mathbf{I}_{m+n}, \tag{15}$$

Similarly, the solution of above optimization problem can be computed by k generalized eigenvector corresponding to k-smallest eigenvalues of $\tilde{\mathbf{L}}\boldsymbol{\beta} = \lambda\tilde{\mathbf{M}}\boldsymbol{\beta}$. With estimated parameters $\tilde{\boldsymbol{\beta}} = [\tilde{\mathbf{W}}^T, \tilde{\mathbf{V}}^T]^T$, we can further calculate the intent features $\tilde{\mathbf{F}}_{\mathcal{Q}}$ and $\tilde{\mathbf{F}}_{\mathcal{P}}$ by Equation (14), $\tilde{\mathbf{F}}_{\mathcal{E}}$ by Equation (3), and learn intent indicator following the same process as that in Section 4.2. Given a new query or web page d and its content feature $\mathbf{p_d}$, we can predict its search intent by Equation (14) based on learned linear model parameters.

Finally, we summarize the entire procedure of the heterogeneous graph-based soft-clustering (HSoC) method for learning unified search intents in Algorithm 1.

4.5 Computational Complexity Analysis

In this section, we will analyze the computational complexity of proposed algorithm using the term *flam* [10].

For construction of heterogeneous graph, the cost for ESA is around $|\mathcal{P}|dt + |\mathcal{P}|t\log t$ flam given the built inverted index where t is average number of indexed concepts for a term ($t \ll |\mathcal{E}|$) and d is average number of terms in a web page ($d \ll n$) [9]. Constructing $\mathbf{Y}$ costs around $|\mathcal{P}|^2d + 4|\mathcal{P}|d + |\mathcal{P}|^2\log|\mathcal{P}|$ flam and similarly, constructing $\mathbf{E}$ costs around

Table 1: Statistics of two click graphs

Data sets	AOL	Commercial
# of unique queries	144,004	103,689
# of unique URLs	103,509	231,232
# of edges	266,954	363,753
# of query words	34,103	38,016
# of page words	50,810	60,177
Total click count	64,295k	20,740k

$|\mathcal{P}|^2e + 4|\mathcal{P}|e + |\mathcal{P}|^2 \log |\mathcal{P}|$ flam where e is average number of concepts related to a web page ($e \ll |\mathcal{E}|$).

For graph embedding, the calculation of sparse matrix $\tilde{\mathbf{L}} = \mathbf{X}^T\mathbf{L}\mathbf{X}$ requires around $|\mathcal{Q}|r + 2|\mathcal{Q}|rc + 2|\mathcal{P}|nd + |\mathcal{P}|d + 2|\mathcal{P}|Kd + 2|\mathcal{P}|md$ flam where c is average number of clicks from a query ($c \ll |\mathcal{P}|$). Similarly, calculation of sparse matrix $\tilde{\mathbf{M}} = \mathbf{X}^T\mathbf{M}\mathbf{X}$ requires around $|\mathcal{Q}|r + 3|\mathcal{P}|d$ flam where r is average number of terms in a query ($r \ll m$). Using Lanczos algorithm, eigen-problem in Equation (15) can be solved using $lk(m+n)(c+2K)$ flam where l is number of iterations for Lanczos ($l \approx 20$) [10].

For clustering, Equations (3) and (14) costs in total $|\mathcal{Q}|kr + |\mathcal{P}|kd + |\mathcal{E}|ke$ flam and fuzzy c-means costs around $(|\mathcal{Q}| + |\mathcal{P}| + |\mathcal{E}|)k^3v$ where v is number of iterations ($v \approx 30$). Overall, the total time complexity for Algorithm 1 is

$$\mathcal{O}\Big(|\mathcal{P}|^2(d + e + \log|\mathcal{P}|) + k^3v(|\mathcal{Q}| + |\mathcal{P}| + |\mathcal{E}|)\Big).$$

5. EXPERIMENTS

In this section, we evaluate the effectiveness of intent features using search ranking and test the performance of the unified search intent learning method on object co-clustering. Two different search datasets are used for evaluation: commercial search data set and AOL search data set.

5.1 Data Preparation

Two real click-through data sets are used for building click graphs in our experiments: AOL search data and one week click-through data collected from a commercial search engine. Statistics of two data sets are shown in Table 1.

5.1.1 Click Graphs

The AOL search data consists of about 20 million query entries collected from about 650k users over three months, which contains 4,811,651 unique queries (after converting to lower case and removing special characters such as punctuation), and 1,632,789 unique URLs (after normalization). We filtered out queries with objectionable content such as pornography and hate speech, and pruned potentially navigational queries (click ratio to a certain URL is more than 0.9). We only keep the links between queries and web pages if the click count is more than 4 since it is more popular and important to study [18], leading to 144,004 queries and 103,509 web pages (URLs) remaining.

For the commercial search log, it originally contained about 4 million entries spread over a week, including 2,111,378 unique queries and 1,897,130 unique URLs. A similar preprocessing procedure is applied to this data except that we set the minimum click frequency to be 20 since the average click count of each query in this data set is much larger than that of AOL data set. Finally we obtained 103,689 unique queries and 231,232 unique web pages (URLs).

5.1.2 Text-Based Web Page Subgraphs

We crawled the page content for each URL and dropped those URLs which have no useful content information, resulting in about 101k URLs remained for AOL data and about 228k URLs remained for the commercial data set. To obtain textual features, we first extracted words from queries and web pages' content, removed stop words and normalized white space. Next, we represented each query or web page as a bag-of-word vector using TF-IDF weighting. We finally obtained 34,103 unique words for queries and 50,810 unique words for web pages in the AOL data set. From the commercial data set, we collected 38,016 unique words for queries and 60,177 unique words for web pages. A text-based KNN graph $G_{\mathcal{P}}$ was then constructed based on the method mentioned in Section 3.

Table 2: Statistics of two web page subgraphs

Data sets	AOL	Commercial
# of unique concepts	2,329,443	2,640,984
# of edges in $\mathbf{E}$	419,332	1,252,184
# of edges in $\mathbf{Y}$	513,326	1,239,856

5.1.3 Concept-Based Web Page Subgraphs

We implemented the ESA algorithm following [9] and extracted top-300 relevant Wikipedia concepts for each web page. Empirically we found the performance of our method does not noticeably change when the number of related concepts is larger than 300. The concept-based web page subgraph $\mathbf{E}$ is then constructed based on the method mentioned in Section 4.1. Statistics of the two kinds of web page subgraphs for the two datasets are shown in Table 2.

5.1.4 Evaluation Sets

To generate ground-truth data set, we manually judged a subset of query-page pairs into five different levels including "Perfect", "Excellent", "Good", "Fair" and "Bad". For AOL data set, we labeled 1,350 queries and 10.3 associated web pages on average. For the commercial data set, we labeled 2,260 queries and 13.2 associated web pages per query. Each ground-truth data set is split into three parts randomly and we used 1/3 as a validation set for tuning parameters, and 2/3 as a testing set for evaluations.

Similarly, we also labeled ground-truth data for query-concept pairs, including 1,080 queries with 8,652 concepts for AOL data set, and 2,120 queries with 10,241 concepts for the commercial data set.

In the rest of this section, we will present our experimental results on both data sets under two different tasks, search ranking and object co-clustering.

5.2 Search Ranking

In this section, we evaluate the effectiveness of the intent features, which are derived from the proposed heterogeneous graph embedding (HGE) method, in terms of search ranking. Given a query, the search ranking problem aims to find pages that best meet the search intents behind the query.

5.2.1 Experimental Settings

To study the usefulness of textual (bag-of-word) and conceptual (Wikipedia concepts) information, we considered four variations to the proposed method: 1) Only click and textual (bag-of-word) information is used; 2) Only click and concept information is used; 3) all types of information are used and 4) use the linear model along with all types of information. Details are listed below.

HGE: We applied heterogeneous graph embedding in the proposed algorithm to learn intent features for queries and

Table 3: Search ranking performance comparisons on AOL dataset and the commercial dataset in terms of MRR (larger is better) and NDCG@1, 3, 5. We report the performance under two different feature dimensionality ($k = 200$ and 300) for the four variations of the proposed method.

Method	AOL data				Commercial data			
	NDCG@1	NDCG@3	NDCG@5	MRR	NDCG@1	NDCG@3	NDCG@5	MRR
BM25	0.6545	0.7477	0.8108	0.3423	0.5929	0.6963	0.7743	0.4508
RW	0.7148	0.7892	0.8308	0.5980	0.6577	0.7905	0.8480	0.6072
MCoC	0.6964	0.7724	0.8354	0.3655	0.6348	0.7461	0.8073	0.4756
M-PLS	0.7667	0.8264	0.8468	0.5732	0.7711	0.8061	0.8490	0.5012
HGE-200	0.7799	0.8306	0.8706	**0.7415**	0.8401	0.8459	**0.8823**	0.7197
HGE-300	0.7748	0.8314	0.8715	0.7268	**0.8423**	**0.8451**	0.8820	**0.7389**
HGE_{CG}-200	0.7956	0.8188	0.8594	0.6531	0.8030	0.8262	0.8671	0.6595
HGE_{CG}-300	0.7891	0.8151	0.8532	0.6764	0.7987	0.8209	0.8624	0.6697
HGE_{EG}-200	0.7933	0.8285	0.8651	0.6641	0.7962	0.8281	0.8700	0.6739
HGE_{EG}-300	**0.8014**	0.8345	0.8698	0.6806	0.7931	0.8241	0.8684	0.7067
HGE_{Fea}-200	0.7703	0.8522	0.8857	0.6126	0.8294	0.8398	0.8712	0.6606
HGE_{Fea}-300	0.7992	**0.8540**	**0.8865**	0.6050	0.7965	0.8218	0.8707	0.6871

web pages. Query-web page similarity is then calculated by Euclidean distance between intent features. For parameters, we set $\lambda_{qp} = 0.1$, $\lambda_t = 0.05$ and $\lambda_e = 0.05$ based on our parameter study (see Section 5.4).

HGE_{CG} : By setting $\lambda_e = 0$ in HGE, we only consider click graph and text-based web page subgraph $\mathbf{Y}$. Other parameters are set to be the same as in HGE.

HGE_{EG} : By setting $\lambda_t = 0$ in HGE, we only consider click graph and concept-based web page subgraph. Other parameters are set to be the same as in HGE.

HGE_{Fea} : This method integrates linear models with HGE as that in Section 4.4. The same parameters are applied here.

For comparison, we also considered four state-of-the-art methods, which are described below.

BM25: BM25 is a purely content-based method. It computes similarity scores between textual features of queries and those of web pages. We use the default settings, *i.e.*, $k_1 = 2$, $b = 0.75$.

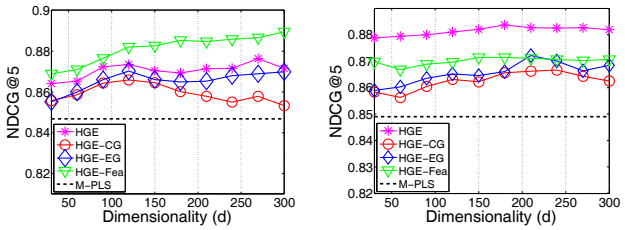
RW: Given a query, RW performs backward random walk on the click graph [7] and provides probabilistic ranking of web pages. The method contains two parameters: the self-transition probability s and number of transition steps t. As suggestion by [7], we fixed $s = 0.9$ and tuned t in the range of $\{1, ..., 20\}$. It is observed that when $t > 5$ the method achieved stable performance. Thus, we set $t = 5$.

MCoC: Multi-class co-clustering (MCoC) [22] aims to learn latent features from a bipartite graph through an embedding procedure. We adopted this method on our click graph and calculated the query-web page similarity by Euclidean distance between corresponding feature vectors. Feature dimension is empirically set to $d = 100$ as suggested in [22].

M-PLS: Multi-view partial least square (M-PLS) [25] uses both click graph and different types of features of queries and web pages to learn query-web page similarity. We consider only textual features $\mathbf{Q}$ and $\mathbf{P}$ in our experiments. Feature dimension is set to $d = 100$ to balance the computational cost and performance [25].

5.2.2 Evaluation Metrics

To evaluate the performance of different methods in the search ranking task, we employ NDCG@1, 3, 5 as our evaluation metrics. The human-judged labels are mapped to ratings 0-4 accordingly.



(a) NDCG@5 on AOL (b) NDCG@5 on Commercial

Figure 2: Search ranking performance versus feature dimensionality ($k = \{30, 60, \ldots, 300\}$) of four alternative methods on AOL and the commercial datasets

In order to evaluate different methods globally, we also calculated Mean Reciprocal Rank (MRR) over the entire data set, which is defined in Equation (16). Since MRR is only influenced by positive records, we treated "Perfect", "Excellent" and "Good" examples as positive.

$$\text{MRR} = \frac{1}{|\mathcal{Q}_T|} \sum_{q_i \in \mathcal{Q}_T} \frac{1}{\text{rank}(q_i)}, \tag{16}$$

where $\mathcal{Q}_T$ is the query set and $\text{rank}(q_i)$ denotes the rank of first positive web page from the entire web page set.

5.2.3 Results

Table 3 summarizes the comparison results for search ranking on both data sets. All four variations of the heterogeneous graph embedding method outperform state-of-the-art algorithms. In particular, the HGE method obtains 24% improvement in MRR and 9.23% improvement in NDCG@1 compared to the best baseline, demonstrating the power of integrating multiple kinds of information.

There are also several interesting observations about the four variations of the proposed method. HGE consistently performs better than HGE_{FEA} on the commercial data set while on AOL data set HGE_{FEA} outperforms HGE in terms of NDCG@1, 3, 5. The difference may come from the quality difference between the textual features of the two data sets. AOL data set may contain richer textual information than that in the commercial data set, yielding less sparse textual features. We can also tell the usefulness of textual and concept information by comparing results of HGE_{EG} with those of HGE_{CG}. It can be seen that HGE_{EG} has better results than HGE_{CG} in most cases, which leads to the conclusion that concepts bring richer information than textual content.

Moreover, HGE achieves significantly better MRR on both of the data sets, showing that its performance is stable in entire training set.

Figure 2 presents NDCG@5 of the four variations with respect to different intent space dimensionalities on both data sets. It can be seen that performances of these methods are not very sensitive to the dimensionality d. On the commercial data set, HGE always outperforms HGE_{FEA} because of the sparsity issue of textual features. Also, HGE and HGE_{FEA} always outperform HGE_{CG} and HGE_{EG} on all dimensionalities, showing that integrating information of the commercial data is helpful. However, we find NDCG@1 on AOL data is sensitive to the change of dimensionality, and HGE_{CG} and HGE_{EG} achieve better results when dimensionality is larger than 200. We believe this is due to the information conflict when merging textual and concept information.

5.3 Object Co-Clustering for Intent Learning

In this section, we evaluate the proposed heterogeneous graph-based soft-clustering method for search intent learning on AOL and the commercial datasets.

5.3.1 Experiment Setup

We compared the proposed method (HSoC) and its variation with the linear model (HSoC-Fea). We also conducted comparisons with several baselines including content-based methods and graph-based methods. Detailed introduction is shown below.

TFIDF: Uses unigrams with TF-IDF weighting to represent queries and web pages; calculates bag-of-word features for concepts based on Equation (3); employs fuzzy c-means clustering based on the Cosine similarity measure.

PLSA: Adopts probabilistic latent semantic analysis to co-cluster queries, web pages and concepts, by representing all objects with bag-of-word features. The number of topics are set to be the same as the number of clusters (k).

BSGP: Adopts bipartite spectral graph partition (BSGP) method [8] on the click graph and the web page-concept subgraph, respectively; We set the dimensionality $d = \lceil \log_2 k \rceil$ as suggested by [8].

MCoC: Uses Multi-class Co-clustering (MCoC) method [26] on the click graph and the web page-concept subgraph.

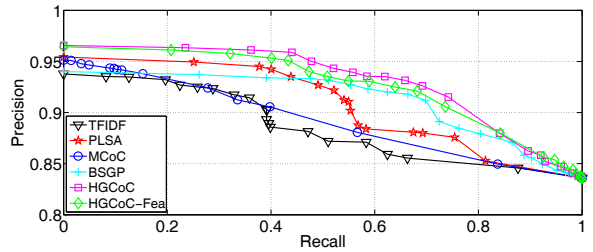
HSoC: Our proposed method for object co-clustering of queries, web pages and concepts. 1) Intent features are first learned; 2) Fuzzy c-means clustering method is adopted on the intent features to learn intent indicators.

HSoC-Fea: Adopt HSoC with the linear model.

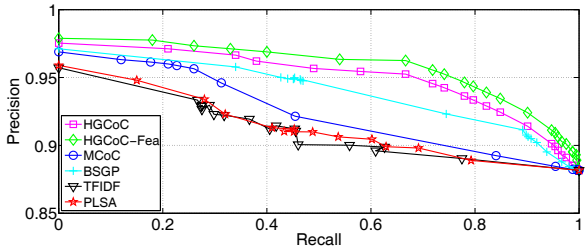
5.3.2 Evaluation Metrics

Evaluation for the proposed method is different from traditional clustering evaluation methods because: 1) we cluster three types of objects collaboratively instead of a single type of objects; 2) we soft-cluster instead of assigning each object to single cluster. Thus, we evaluated the heterogeneous graph-based soft-clustering results by precision, recall, and F_1 score over pairs of objects including query-page pairs and query-pairs [13].

Specifically, we first generated binary judgements by treating pairs with "Perfect", "Excellent" and "Good" labels as positive examples, and those with "Bad" labels as negative ones. We define TP (true positive) as the number of positive objects pairs that are correctly assigned to same cluster, FP (false positive) as the number of negative object pairs that are assigned to the same cluster, TN (true negative) as the number of negative object pairs that are assigned to different clusters and FN (false negative) as the number of positive object pairs that are assigned to different clusters.



(a) AOL dataset



(b) The commercial dataset

Figure 3: Precision-recall curves (plot by varying the number of intents, *i.e.*, dimensionality k) of HSoC, HSoC_{Fea} and baselines for (a) the AOL and (b) the commercial datasets

Precision (P) is calculated as the fraction of pairs correctly put in the same cluster (*i.e.*, P=TP/(TP + FP)), recall (R) is the fraction of similar pairs that were identified (*i.e.*, R=TP/(TP + FN)), F_1 score is the harmonic mean of precision and recall (*i.e.*, $\text{F}_1 = 2\text{PR}/(\text{P} + \text{R})$). We also adopt the widely used clustering measure Rand index [21], defined as RI=(TP + TN)/(TP + FP + FN + TN).

5.3.3 Performance Comparisons

Since the ground-truth data set consists of binary-labeled object pairs, we propose to evaluate the soft-clustering methods (PLSA, MCoC and the proposed methods) by assigning each object to its 3 most confident intents in the intent indicator. For hard clustering methods (TFIDF and BSGP), each object is only assigned to single intent.

Table 4(a) and Table 4(b) show comparison results for 20 intents and 50 intents on both data sets. Each method is repeated for 10 trials for accurate clustering performance evaluation. We use bold number to highlight the best results on different metrics. The proposed HSoC and HSoC-Fea methods achieve significantly better results on Precision, Recall, F_1 score and Rand Index (RI), demonstrating the integration of different information in a unified framework does help the effectiveness of intent learning. In particular, we find content-based methods such as TFIDF outperform our methods and graph-based methods on Precision, but perform very poor on Recall and other metrics. This shows that content-based methods tend to generate purer but less complete clusters as intents while our methods generate intent clusters with better coverage. The reason may be because the unified optimization on heterogeneous graphs helps identify objects which share intents in a more comprehensive

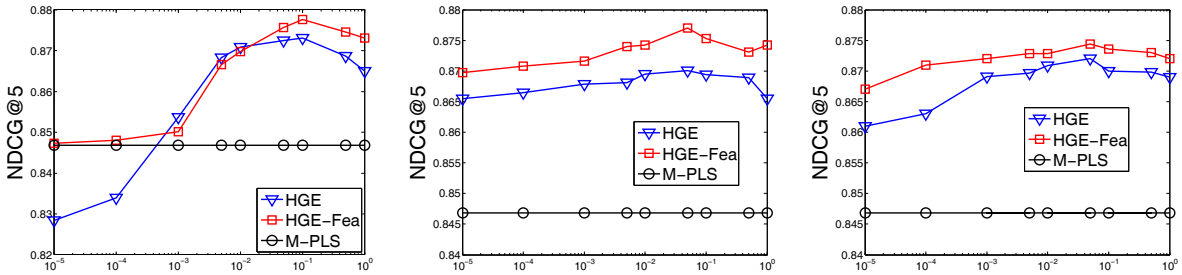
Table 4: Object co-clustering performance comparisons on (a) AOL and (b) the commercial datasets in terms of Precision, Recall, F_1 and Rand index (RI). We set the number of intents to be 20 and 50 ($k = 20, 50$).

(a) The AOL dataset

	performance on 20 clusters				performance on 50 clusters			
Method	Precision	Recall	F_1	RI	Precision	Recall	F_1	RI
TFIDF	0.8841±.01	0.4569±.07	0.5992±.06	0.4497±.07	0.8986±.01	0.3996±.07	0.5496±.07	0.4519±.04
PLSA	**0.8978±.01**	0.5756±.02	0.7013±.01	0.5904±.01	**0.9085±.01**	0.5551±.02	0.6889±.01	0.5812±.01
BSGP	0.8550±.00	0.8755±.00	0.8842±.00	0.7996±.00	0.8549±.00	0.8635±.00	0.8541±.00	0.7781±.00
MCoC	0.8870±.01	0.5267±.07	0.6582±.05	0.5476±.05	0.8823±.01	0.3169±.02	0.4054±.02	0.3658±.01
HSoC	0.8465±.00	0.9534±.01	0.8967±.00	0.8163±.00	0.8604±.01	0.8857±.01	0.8728±.00	0.7842±00
$HSoC_{Fea}$	0.8417±.00	**0.9776±.01**	**0.9045±.00**	**0.8275±.01**	0.8614±.00	**0.8880±.02**	**0.8743±.01**	**0.7868±.01**

(b) The commercial dataset

	performance on 20 clusters				performance on 50 clusters			
Method	Precision	Recall	F_1	RI	Precision	Recall	F_1	RI
TFIDF	**0.9083±.00**	0.4496±.04	0.6012±.03	0.4752±.02	0.9150±.00	0.4011±.04	0.5565±.03	0.4390±.03
PLSA	0.9052±.01	0.4881±.02	0.6352±.01	0.5060±.01	0.9105±.01	0.4527±.01	0.6047±.01	0.4728±.01
BSGP	0.9018±.00	0.9198±.00	0.9107±.00	0.8410±.00	0.9079±.00	0.8985±.00	0.9032±.00	0.8301±.00
MCoC	0.8822±.00	0.9258±.00	0.9036±.00	0.8365 ±.00	0.8926±.01	0.8311±.08	0.8589±.04	0.7632±.06
HSoC	0.8928±.00	**0.9667±.01**	0.9283±.00	**0.8683±.00**	0.9012±.00	**0.9472±.00**	0.9236±.00	0.8619±.00
$HSoC_{Fea}$	0.9028±.01	**0.9667±.00**	**0.9337±.00**	**0.8683±.00**	**0.9182±.01**	**0.9472±.00**	**0.9324±.00**	**0.8689±.00**



(a) Varing $\lambda_{\mathcal{QP}}$ (b) Varing λ_T (c) Varing $\lambda_{\mathcal{E}}$

Figure 4: Parameter study for $\lambda_{\mathcal{QP}}$, λ_T and $\lambda_{\mathcal{E}}$ in terms of NDCG@5 on search ranking

way. Also, we see that the $HSoC_{Fea}$ method outperforms HSoC in general, demonstrating the effectiveness of the linear model for the clustering problem.

To compare the performance with respect to the number of intents, we plot precision-recall curves in Figure 3 for both data sets by varying the number of cluster, k, from 1 to the number of objects in the ground-truth data set. First, the proposed HSoC and $HSoC_{Fea}$ methods clearly outperform baselines significantly, showing that unified optimization with three types of information helps. Also, we find that BSGP achieves similar performance as our methods when k is small. This is probably because adding concept information does not help much in coarse level clustering.

5.3.4 *Case Study*

To provide a clearer look at the learned intents, we randomly selected 4 intent clusters to give as examples. For each intent cluster, we randomly picked 4 queries, 4 URLs and 4 concepts to show in Table 5.

First, we can see that the 4 randomly selected clusters cover 4 generic search intents from different domains. For example, cluster 19 is about software while cluster 43 is about music. This verifies that the proposed method is able to learn generic search intents which cover different domains. The queries "enterprise" (in cluster 8) and "office" (in cluster 19) indicate that the proposed method can handle ambiguous queries well. Also, we find the proposed method is able to handle queries with typos or shortenings such as "xboxx" in cluster 5 and "video2map3" in cluster 19, indicating that click-through and concept information are good complements for raw text. Finally, it can be observed that concepts from each intent cluster assist in intent interpretation, especially when the queries and URLs are hard to understand directly (see cluster 5 and 43 in Table 5).

5.4 Parameter Study

In this section we study the impact of three parameters (*i.e.*, $\lambda_{\mathcal{QP}}$, λ_T and $\lambda_{\mathcal{E}}$) in the proposed method. They control the information trade-off between query-page, text-based web page and concept-based web page relations, respectively. We studied the importance of these three relations on validation data sets by varying one of the parameter and fixing the other two parameters.

Figure 4 shows the search ranking performance in terms of NDCG@5 with respect to different parameter values on the AOL data set. We find the proposed method is more sensitive to $\lambda_{\mathcal{QP}}$ than the other two parameters, showing that query-web page relationship is more important to the search ranking performance. Also, increasing $\lambda_{\mathcal{E}}$ leads to larger performance enhancement than increasing λ_T, which may be due to richer information from concepts.

6. CONCLUSION AND FUTURE WORK

In this paper, we study the problem of learning unified search intents for queries, web pages and Wikipedia concepts. The proposed method aims to leverage three types of objects together to tackle drawbacks from using only a single type of data source, yielding better understanding of intents. Also, bridging intents with concepts helps interpret intents

Table 5: Examples of 4 randomly selected search intents (from 50 intent clusters) learned by $HSoC_{Fea}$ method. We randomly selected 4 queries, 4 URLs and 4 Wikipedia concepts from each of the selected intent cluster.

Queries	URLs	Concepts
cluster ID: 5 (video games)		
pokemon	http://gamehouse.com/mahjong-games	crazy_taxi
daily jigsaw	http://gamestop.com/xbox360	plants_vs._zombies
nickmom	http://freefishinggames.biz	call_of_duty_(video_game)
xboxx	http://freegamepick.com	the_sims
cluster ID: 8 (car rentals)		
car rental	http://rentals.com	the_hertz_corporation
carmax	http://carmax.com/enus/car-search/new-cars.html	enterprise_tent-a-car
hertz car rental	http://enterprise.com/car_rental/contactus.do	budget_rent_a_car
enterprise	http://carmax.com	alamo_rent_a_car
cluster ID: 19 (softwares)		
adobe flash player	http://adobe.com/products/reader.html	microsoft_office
video2map3	http://freemake.com/free_youtube_converter	nuance_pdf_reader
office	http://support.microsoft.com/ph/14019	windows_defender
google chrome	http://download.cnet.com/windows/activex	quest_software
cluster ID: 43 (music)		
nicki minaj	http://youtubemusic.net	flo_rida
pitbull	http://en.wikipedia.org/wiki/john_mayer	linkin_park
adele	http://buzzworld.in/songs-pk	die_another_day_(song)
karaoke	http://pitbullmusic.com/us	pitbull_(rapper)

when it is hard to understand queries and URLs. Technically, we cast this problem into a heterogeneous graph-based soft-clustering problem and develop an effective and efficient algorithm for solving it. Experimental results from search ranking demonstrate the effectiveness of the proposed intent features and the co-clustering results show significant improvement compared with state-of-the-art methods.

Interesting future work includes: 1) extending our method to automatically learn the importance of different types of relations; and 2) enabling our method to update the intent indicator in an online manner so that newly emerged search intents can be efficiently included.

7. ACKNOWLEDGEMENTS

The work was supported in part by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA) and W911NF-11-2-0086 (Cyber-Security), the U.S. Army Research Office under Cooperative Agreement No. W911NF-13-1-0193, DTRA, MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC, and U.S. National Science Foundation grants CNS-0931975, IIS-1017362, IIS-1320617, IIS-1354329.

8. REFERENCES

[1] L. M. Aiello, D. Donato, U. Ozertem, and F. Menczer. Behavior-driven clustering of queries into topics. In *CIKM*, 2011.

[2] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, 2001.

[3] J. C. Bezdek, R. Ehrlich, and W. Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203, 1984.

[4] I. Bordino, G. De Francisci Morales, I. Weber, and F. Bonchi. From machu_picchu to rafting the urubamba river: anticipating information needs via the entity-query graph. In *WSDM*, 2013.

[5] H. Cao, D. H. Hu, D. Shen, D. Jiang, J.-T. Sun, E. Chen, and Q. Yang. Context-aware query classification. In *SIGIR*, 2009.

[6] J. C. K. Cheung and X. Li. Sequence clustering and labeling for unsupervised query intent discovery. In *WSDM*, 2012.

[7] N. Craswell and M. Szummer. Random walks on the click graph. In *SIGIR*, 2007.

[8] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *SIGKDD*, 2001.

[9] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, 2007.

[10] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[11] Q. Gu and J. Zhou. Co-clustering on manifolds. In *SIGKDD*, 2009.

[12] Z. Guan, C. Wang, J. Bu, C. Chen, K. Yang, D. Cai, and X. He. Document recommendation in social tagging services. In *WWW*, 2010.

[13] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[14] X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003.

[15] J. Hu, G. Wang, F. Lochovsky, J. Sun, and Z. Chen. Understanding user's query intent with wikipedia. In *WWW*, 2009.

[16] Y. Hu, Y. Qian, H. Li, D. Jiang, J. Pei, and Q. Zheng. Mining query subtopics from search log data. In *SIGIR*, 2012.

[17] M. Ji, J. Yan, S. Gu, J. Han, X. He, W. V. Zhang, and Z. Chen. Learning search tasks in queries and web pages via graph regularization. In *SIGIR*, 2011.

[18] X. Li, Y. Wang, and A. Acero. Learning query intent from regularized click graphs. In *SIGIR*, 2008.

[19] J. Liu, C. Wang, J. Gao, and J. Han. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*, 2013.

[20] F. Radlinski, M. Szummer, and N. Craswell. Inferring query intent from reformulations and clicks. In *WWW*, 2010.

[21] W. M. Rand. Objective criteria for the evaluation of clustering methods. *JSTOR*, 66(336):846–850, 1971.

[22] E. Sadikov, J. Madhavan, L. Wang, and A. Halevy. Clustering query refinements by user intent. In *WWW*, 2010.

[23] Y. Sun, Y. Yu, and J. Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *SIGKDD*, 2009.

[24] X. Wang, D. Chakrabarti, and K. Punera. Mining broad latent query aspects from search sessions. In *SIGKDD*, 2009.

[25] W. Wu, H. Li, and J. Xu. Learning query and document similarities from click-through bipartite graph with metadata. In *WSDM*, 2013.

[26] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *WWW*, 2012.

[27] T. Yamamoto, T. Sakai, M. Iwata, C. Yu, J. Wen, and K. Tanaka. The wisdom of advertisers: mining subgoals via query clustering. In *CIKM*, 2012.

[28] X. Yin and S. Shah. Building taxonomy of web search intents for name entity queries. In *WWW*, 2010.

[29] H. Zeng, Q. He, Z. Chen, W. Ma, and J. Ma. Learning to cluster web search results. In *SIGIR*, 2004.

[30] X. Zhu, J. Guo, X. Cheng, and Y. Lan. More than relevance: high utility query recommendation by mining users' search behaviors. In *CIKM*, 2012.