Understanding User's Query Intent with Wikipedia

Jian Hu[1], Gang Wang[1], Fred Lochovsky[2], Jian-Tao Sun[1], Zheng Chen[1]

[1]Microsoft Research Asia
No. 49 Zhichun Road, Haidian District
Beijing 100080, P.R. China
{jianh, gawa, jtsun, zhengc}@microsoft.com

[2] The Hong Kong University of Science & Technology
Clear Water Bay, Kowloon,
Hong Kong, P.R. China
fred@cse.ust.hk

ABSTRACT

Understanding the intent behind a user's query can help search engine to automatically route the query to some corresponding vertical search engines to obtain particularly relevant contents, thus, greatly improving user satisfaction. There are three major challenges to the query intent classification problem: (1) Intent representation; (2) Domain coverage and (3) Semantic interpretation. Current approaches to predict the user's intent mainly utilize machine learning techniques. However, it is difficult and often requires many human efforts to meet all these challenges by the statistical machine learning approaches. In this paper, we propose a general methodology to the problem of query intent classification. With very little human effort, our method can discover large quantities of intent concepts by leveraging Wikipedia, one of the best human knowledge base. The Wikipedia concepts are used as the intent representation space, thus, each intent domain is represented as a set of Wikipedia articles and categories. The intent of any input query is identified through mapping the query into the Wikipedia representation space. Compared with previous approaches, our proposed method can achieve much better coverage to classify queries in an intent domain even through the number of seed intent examples is very small. Moreover, the method is very general and can be easily applied to various intent domains. We demonstrate the effectiveness of this method in three different applications, i.e., travel, job, and person name. In each of the three cases, only a couple of seed intent queries are provided. We perform the quantitative evaluations in comparison with two baseline methods, and the experimental results show that our method significantly outperforms other approaches in each intent domain.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: *Information Search and Retrieval - Search process;* I.5.1 [**Pattern Recognition**]: *Models - Statistical*

General Terms

Algorithms, Experimentation

Keywords

Query intent, User intent, Wikipedia, Query Classification

1. INTRODUCTION

With the global search market approaching 10 billion queries per month and substantial incentives to increase market share, maximizing user satisfaction with search results continues to be a central goal for search engine companies. Recently, a number of vertical search engines, e.g. news search, image search, map search, job search product search, etc, have appeared that provide a specialized type of information service to satisfy a user's need according to a particular intent. Using a more specialized user interface, a vertical search engine can return more relevant and essential results than a general search engine for in-domain web queries. In practice, a user has to identify his intent in advance and decide which vertical search engine to choose to satisfy his intention. It would be convenient if a query intent identifier could be provided in a general search engine that could accurately predict whether a query should trigger a vertical search in a certain domain. Moreover, since a user's query may implicitly express more than one intent, it would be very helpful if a general search engine could detect all the query intents, distribute the query to appropriate vertical search engines and effectively organize the results from the different vertical search engines to satisfy a user's information need. Consequently, understanding a user's query intent is crucial for providing better search results and thus improving user satisfaction.

Typical queries submitted to web search engines contain very short keyword phrases [18], which are generally insufficient to fully describe a user's information need. Thus, it is a challenging problem to classify millions of queries into some predefined categories. A variety of related topical query classification problems have been investigated in the past [1] [3] [5] [6]. Most of them seek to use statistical machine learning methods to train a classifier to predict the category of an input query. From the statistical learning perspective, in order to obtain a classifier that has good generalization ability in predicting future unseen data, two conditions should be satisfied: discriminative feature representation and sufficient training samples. However, for the problem of query intent classification, even though there are huge volumes of web queries, both conditions are hardly to met due to the sparseness of query features coupled with the sparseness of labeled training data[1].

Overall, the query intent classification problem involves the following three challenges:

Intent representation challenge: how to define a semantic representation that can precisely understand and distinguish the intent of the input query. Traditional approaches use a number of seed queries as the representation of an intent. Thus, the intent is characterized by a set of queries. As such, in order to achieve better representation ability, a large number of labeled examples are required. Previous works [1] and [14] focused on solving this problem by augmenting the labeled queries through automatically labeling more user input queries using semi-supervised learning over search log data. However, such intent representation inevitably requires a huge amount of human effort.

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

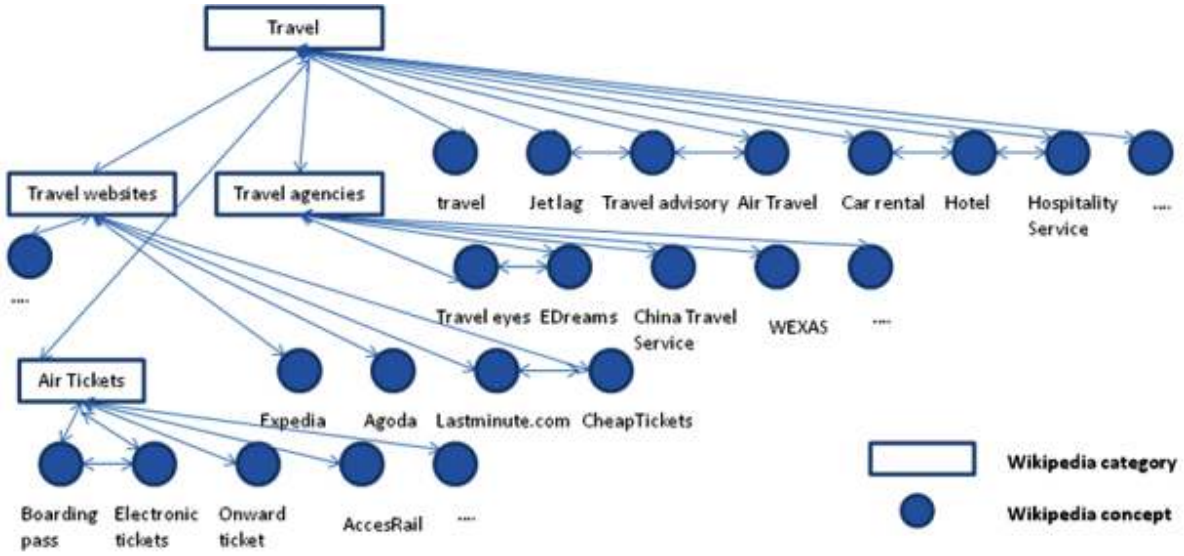


Figure 1. An example of travel intent concepts link graph

Domain coverage challenge: how to clarify the semantic boundary of the intent domain so that the intent classifier can correctly detect whether the input query falls under its domain. The coverage of an intent classifier using the previous learning approaches heavily relies on labeled samples. If the input samples only cover a subset of concepts in an intent domain, the learned classifier cannot make good predictions for those queries that are not covered by the training samples. This problem is an important issue in query intent classification. However, there are few public works addressing it.

Semantic interpretation challenge: how to correctly understand the semantic meaning of the input query. The bag-of-word based model is based on a simplified assumption that often leads to the word sense ambiguity problem due to the short length of an input query. For example, the query "Steve Jobs" has the person name intent, but not the job intent, while the query "rotten stone" does not have the person name intent. Previous works [3] [5] [13] attempted to solve this problem through augmenting the query with more features using external knowledge, such as search engine results. However, their methods are still based on the bag-of-word model.

The motivation of this work is to meet these challenges in query intent classification. In this paper, we leverage the world's largest human knowledge base, Wikipedia, to help understand a user's query intent. Wikipedia is the largest online encyclopedia. It contains more than 2 million entries, referred to as *Wikipedia concepts* throughout this paper, and most of them are representative name entities and keywords of different domains. Each concept is summarized by an article and belongs to at least one category. Each Wikipedia concept article may be linked to other related concepts by hyperlinks. Categories of Wikipedia are organized hierarchically into an ontology. Wikipedia is a very dynamic and fast growing resource – articles about newsworthy events around the world are often added within a few days after their occurrence. Our proposed method attempts to address each challenge described above and can be summarized as follows:

- We use the Wikipedia concepts as the intent representation, and each intent domain is represented as a set of Wikipedia articles and categories. It is very easy to collect a well-organized and comprehensive concept set for an intent domain. For example, if we aim to identify the travel intention, we first search using the query "travel" in Wikipedia. Through browsing the category of the concept "travel," its siblings concepts and the concepts it links to, we can collect a large number of key concepts and subcategories that cover almost all aspects of the "travel" domain, for example, travel agency, travel tips, transportation services, such as airlines and taxis, and accommodation, such as hotels and entertainment venues. An illustration of this example is given in Figure 1. The seed examples for travel intent are discovered with little human effort.
- Wikipedia includes nearly every aspect of human knowledge and is organized hierarchically as an ontology. We use the Markov random walk algorithm to iteratively propagate intent from the seed examples into the Wikipedia ontology and assign an intent score to each Wikipedia concept. Hence, we obtain an intent probability for each concept in Wikipedia, which clearly identifies the semantic boundary of the intent domain.
- We do not use the bag-of-word model as the feature representation. Instead, we map each query into a Wikipedia representation. If the input query can be exactly mapped to a Wikipedia concept, we can easily predict its intent based on its associated intent probability. Otherwise, we map the query to the most related Wikipedia concepts using *explicit semantic analysis* (ESA) [23], and make the judgment based on the intent probabilities of mapped Wikipedia concepts, which overcomes the semantic disambiguation issue.

While we demonstrate the effectiveness of our method in three applications, personal name intent, job intent and travel intent, our approach is general enough to be easily applied to other tasks. The quantitative evaluations in comparison with two baseline methods show that our method significantly outperforms other methods in each intent domain. The rest of the paper is organized as follows. Section 2 introduces the link structure of Wikipedia. Section 3 presents our methodology to infer a query's intent based on the knowledge extracted from Wikipedia. Section 4 describes our evaluation methodology, experiments and results. Section 5 discusses related work followed by concluding remarks in Section 6.

2. WIKIPEDIA

Before introducing the core algorithms, we first introduce the structure of Wikipedia, especially the various kinds of links contained in Wikipedia. Launched in 2001, Wikipedia is a multilingual, web-based, free content encyclopedia written collaboratively by more than 75,000 regular editing contributors; its articles can be edited by anyone with access to its web site. Wikipedia is a very dynamic and fast growing resource – articles about newsworthy events are often added within a few days after their occurrence. Each article in Wikipedia describes a single topic; its title is a succinct, well-formed phrase that resembles a term in a conventional thesaurus [8]. Each article must belong to at least one category of Wikipedia. Hyperlinks between articles keep many of the same semantic relations such as equivalence relation, hierarchical relation and associative relation.

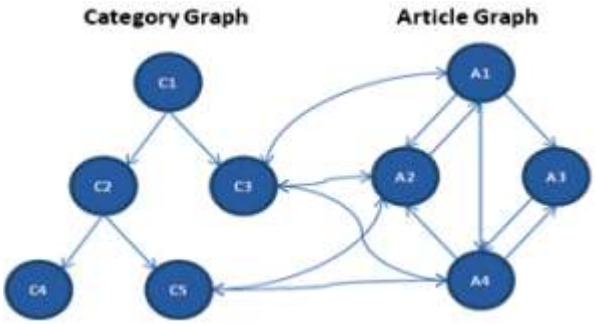


Figure 2. Relations beween Category Graph and Article Graph

2.1.1 Article Links

Wikipedia is structured as an interconnected network of articles. Each article can link to several Wikipedia entries. A contributor can insert a hyperlink between a word or phrase that occurs in the article and corresponding Wikipedia entry when editing an article. If we denote each Wikipedia article as a node, and each hyperlink between articles as an edge pointing from one node to another, then Wikipedia articles form a directed graph (see right side of Figure 2).

2.1.2 Category Links

In Wikipedia, both articles and categories can belong to more than one category, i.e. the article of "Puma" belongs to two categories: "Cat stubs" and "Felines". These categories can be further categorized by associating them with one or more parent categories. Consequently, the category structure of Wikipedia does not form a simple tree-structured taxonomy, but a directed acyclic graph in which multiple categorization schemes co-exist simultaneously [8], which makes Wikipedia categories not a taxonomy with a fully-fledged subsumption hierarchy, but only a thematically organized thesaurus.

In Wikipedia, each article can link to several categories, where each category is a kind of semantic tag for that article. A category backlinks to all articles in its category list. Thus, the article graph and the category graph are heavily interlinked (Figure 2). Links in the article and category graphs play different roles: article links are established due to associate relation between articles, while links between categories are typically established due to hyponymy or meronymy relations.

2.1.3 Redirect Links

Wikipedia guarantees that there is only one article for each concept by using "Redirect Pages" to link equivalent concepts to a preferred one. A redirect page exists for each alternative name that can be used to refer to a Wikipedia concept. We collect the titles of redirect pages for each Wikipedia concept and use them as synonyms for each concept. If a user's input query matches one of the synonyms of a concept, we will use the intent of that concept directly.

2.1.4 Disambiguation Pages

In Wikipedia, disambiguation pages are solely intended to allow users to choose among several Wikipedia concepts for an ambiguous term. Take "jaguar" as an example, which can be denoted as an animal, a car, a symbol and so on. There are 23 concepts in the disambiguation page of "jaguar". If a user's input query is ambiguous according to the Wikipedia disambiguation pages, we will not make any intent prediction for this query.

3. METHODOLOGY

In this section, we present a new methodology for understanding a user's query intent using Wikipedia knowledge. We begin with an overview of the overall solution (see Figure 3) and then describe each step in detail.

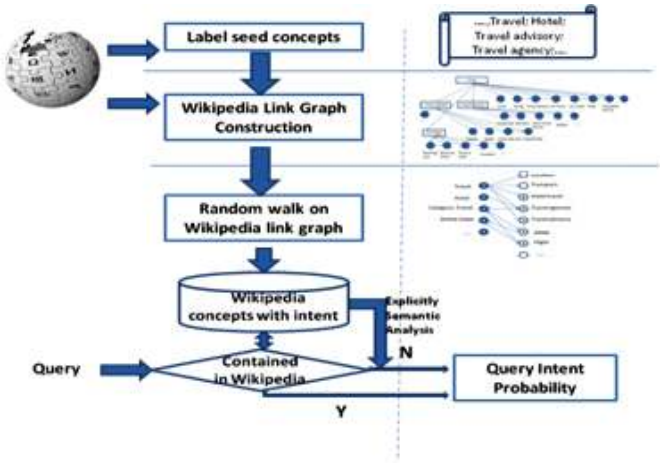


Figure 3. Overview of our method

3.1 Overview

When training a query intent predictor for a specific domain, we first select a few queries, which should be representative queries of the specific domain, and then we map these queries to Wikipedia concepts. Based on these mapped Wikipedia concepts, we can browse the categories to which they belong, their sibling concepts, and the concepts they link to in Wikipedia. Thus, we can easily collect a large amount of representative Wikipedia concepts and categories as a set of seed concepts.

As described in [26] [27], Wikipedia concepts, which link to each other through article or category links, often share similar topics. To complement the knowledge that is not covered by the seed concepts we collect, we first build a Wikipedia link graph in which each entry denotes a link relation either within concept articles and categories or between articles and categories. After that, through a random walk on the built Wikipedia link graph based on the intent label vector, we can get a vector of probabilities measuring how probable each concept in the link graph belonging to the defined intent.

For a user's input query, if it is already covered by Wikipedia concepts, we can easily judge whether the query has the defined intent or not based on the intent probability of the matched Wikipedia concept. For a query that is not covered by Wikipedia concepts, we employ a method similar to *explicit semantic*

analysis (ESA) [22] to map the query to its most related Wikipedia concepts and make the intent judgment based on the intent probabilities of the mapped Wikipedia concepts.

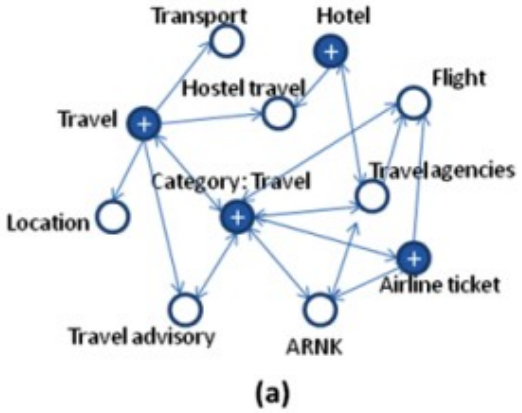
3.2 Wikipedia Link Graph Construction

Based on the article and category links of Wikipedia, we can construct a link graph $G = (X,E)$ where $X = A \cup C, A = \{a_i\}_{i=1}^{m}$ represents a set of Wikipedia concept articles and $C = \{c_k\}_{k=1}^{n}$ a set of Wikipedia categories. Each edge in E connects a vertex within concept articles, within categories, or between concept articles and categories. There are edges between two vertices in the same set and also between vertices in the different sets. As described in [26], there are some useless links between articles, and the existence of a link does not always imply that the two articles are related. In fact many phrases in a Wikipedia article link to other articles just because there are entries for the corresponding Wikipedia concepts. Therefore, to assure topic relatedness of linked entries, two entries have an edge only when they link to each other in Wikipedia.

Let $\boldsymbol{W}$ represent an $(m+n)\times(m+n)$ weight matrix, in which element w_{ij} equals the link count associating vertices between x_i and x_j in the matrix. Since the link between two vertices is undirected, the matrix W is symmetric due to $w_{ij} = w_{ji}$. Furthermore, we assume that there is a small set of seed concepts or categories, denoted as X_L, that are manually selected from Wikipedia as positive examples with respect to a specific intent. Given the constructed link graph and the labeled set $X_L = \{ X_{L1,\ldots,} X_{Lp} \}$, our goal is to automatically propagate labels to more concepts and categories in the set $X \setminus X_L$.

3.3 Random Walk on the Link Graph

Before introducing the algorithm, we first describe the intuition behind it using Markov random walk. Consider the example presented in Figure 4. Suppose we labeled four concepts *"Travel","Hotel","Category: Travel",* and*"Airline ticket*" as seeds for the travel intent. Since Wikipedia concepts, which link to each other through article or category links, often share similar topics, we assume that their immediate neighbors also have the same kind of intent to some extent. Iteratively, we propagate the intent of travel from its seed concepts to their neighborhood vertices. The algorithm proceeds until it converges to a stable state, and all the concepts in the graph have a probability that they belong to travel intent. In this example, *"Flight", "Travel advisory", "ARNK", "Travel agencies"* and *"Hostel travel"* have a high travel intent probability after propagation.



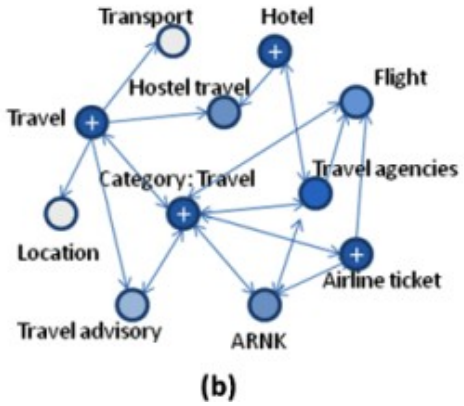


Figure 4. An example of propagating travel intent on link graph. (a) A link graph where blue nodes represent seed concepts labeled +; (b) Label information is propagated from the seed concepts to unlabeled concepts (the darkness of blue denotes the intent probability).

We define transition probabilities $P_{t+1|t}(x_k| x_j)$ from the vertex x_j to x_k ($x_j, x_k \in X$) by normalizing the score out of node x_j, so,

$$P_{t+1|t}(x_k| x_j) = w_{jk} / \sum_i w_{ji} \quad (1)$$

where i ranges over all vertices connecting to x_j. The notation $P_{t+1|t}(x_k \mid w_j)$ denotes the transition probability from node x_j at time t to node x_k at time $t+1$. While the score w_{jk} is symmetric, the transition probabilities $P_{t+1|t}(w_k|w_j)$ generally are not, because the normalization varies across nodes.We rewrite the one-step transition probabilities in a matrix form as $\boldsymbol{P} = [P_{t+1|t}(w_k|w_j)]_{jk}$ with size $(m+n)\times(m+n)$. The matrix $\boldsymbol{P}$ is row stochastic so that rows sum to 1. This transition matrix can also be calculated through

$$\boldsymbol{P} = \boldsymbol{D}^{-1/2}\boldsymbol{S}, \text{ where } \boldsymbol{D} = diag(\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}),$$

The function *diag(.)* results in a diagonal matrix in which all elements are zero except the diagonal elements whose values are from the input matrix of the same positions.

We consider the case where a small set of seed concepts or categories $X_L = \{ X_{L1,\ldots,} X_{Lp} \}$ is labeled as +1. The initial vector $v^0 = (p_0(x_j))_{j=1}^{m+n}$ is an $(m+n)$-dimensional vector with values

$$p_0(x_j) = \begin{cases} x_j / \sum_i x_{Li} & \text{if } j \in \{L_1, \ldots, L_p\} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $p_0(x_j)$ is the probability that a random walk starts from x_j. Given this definition, the random walk algorithm works as follows:

Algorithm 1:

Input: vector $\boldsymbol{v^0}$ and transition matrix $\boldsymbol{P}$

Output: $\boldsymbol{v^*}$

1: Initialize $\boldsymbol{v} = \boldsymbol{v^0}$;

2: Repeat

3: compute $\boldsymbol{v} = \alpha\, \boldsymbol{P}^T \boldsymbol{v^0} + (1-\alpha)\, \boldsymbol{v^0}$, where $\alpha \in [0, 1)$

4: Until $\boldsymbol{v}$ converges to $\boldsymbol{v^*}$

After t iterations, the transition probability from vertex x_j to vertex x_k, denoted by $P_{t|0}(w_k|w_j)$, is equal to $P_{t|0}(w_k|w_j) = [\boldsymbol{P}^t]_{jk}$. The random walk sums the probabilities of all paths of length t between two vertices and gives a measure of the volume of the paths from one vertex to another. If there are many paths, the transition probability will be higher. Since the matrix $\boldsymbol{P}$ is a stochastic matrix, the largest eigenvalue of $\boldsymbol{P}$ is 1 and all other eigenvalues are in [0, 1). Consequently, it is easy to see that the sequence of $\boldsymbol{v}$ asymptotically converges to

$$\boldsymbol{v}^* = (1-\alpha)(\boldsymbol{I}-\alpha\boldsymbol{P})^{-1}\boldsymbol{v}^0. \quad (3)$$

Since the score matrix $\boldsymbol{W}$ is a sparse matrix, most elements in $\boldsymbol{P}$ are zero. The computation complexity of each iteration in **Algorithm 1** is linear in the number of graph edges. Thus, it is very efficient to obtain the converged solution.

The value in the entry of the vector $\boldsymbol{v}^*$ is the posterior probability $P^*(x_j)$ $(j=1,\ldots,m+n)$ that the vertex x_j is associated with a specified intent. Therefore, each Wikipedia article or category is assigned a probability, i.e., $P^*(x_j)$, reflecting the degree of intent. The algorithm generalizes naturally to the case where multiple intents are specified. We treat the multiple intents classification as a set of binary intent classification. Consequently, **Algorithm 1** can be directly applied for each intent separately.

3.4 Query Intent Predictor

After random walk on the constructed Wikipedia link graph, each concept in the link graph will have a probability that it belongs to the intent we defined. For queries that are covered by Wikipedia concepts, we can get the probability that those queries belong to the defined intent using the intent probability of the mapped Wikipedia concepts.

Since there are hundreds of thousands of new queries generated by online users every day, it is highly likely that the Wikipedia concepts cannot cover all these search queries. Consequently, we need to address the problem of how to predict the intent of queries that are not covered by Wikipedia concepts. Wikipedia contains more than two million named entities, which cover almost all aspects of our daily life, and there is one article that provides a detailed summarization for each Wikipedia concept. Gabrilovich *et al.* [23] proposed explicit semantic analysis (*ESA*), which utilizes Wikipedia for feature generation for the task of computing semantic relatedness. Empirical evaluation indicates that using *ESA* leads to substantial improvements in computing relatedness between words and text, and the correlation of computed relatedness of *ESA* with human judgments is much better than previous state of the art [22]. Therefore, for queries not covered by Wikipedia, we try to map these queries to a few most similar Wikipedia concepts using a method similar to *ESA*, and then predict the intent of the queries using the intent of mapped Wikipedia concepts.

3.4.1 Explicit Semantic Analysis (ESA)

The intuition behind the use of *ESA* on Wikipedia can be described as: given a text fragment, *ESA* will provide a semantic interpreter that maps the fragment of text into some related concepts from Wikipedia ordered by their relevance to the input text fragment.

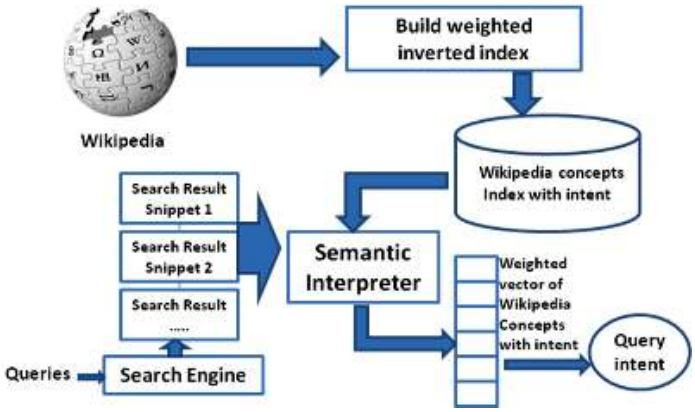


Figure 5. The process of intent predictor

Each Wikipedia concept is represented as a vector of words that are contained in the corresponding article. Entries of these vectors are weighted using the *TFIDF* scheme, which quantifies the strength of the relatedness between words and concepts. To speed up the semantic interpreter, we build an inverted index, which maps each word into a list of Wikipedia concepts in which it appears. The given text fragment is first represented as a vector of words weighted also by *TFIDF*. Then, the *ESA* semantic interpreter will go through each text word, retrieve corresponding entries in the inverted index, and merge them into a vector of concepts that is ordered by their relevance to the input text. The similarity measure used in the retrieval process in [22] is cosine similarity, while in our work we use the *OKAPI BM 25*[2]. *BM25* is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document. It has been proven to be quite effective especially in text document retrieval [2].

3.4.2 Intent Predictor

As shown in Figure 5, for queries that are not contained in Wikipedia, we first enrich the features of the query with the top K search result snippets from Live Search[1]. We only use the title and description part of the search result snippets. The words appearing in snippet titles are additionally scaled by a constant factor ε ($\varepsilon=3$ in our experiment). Then, we segment each title and description of the top K search result snippets into sentences. For each sentence, we can get a list of relevant Wikipedia concepts by *ESA*. After summarizing the max-min normalized [21] rank scores of retrieved concepts of all the sentences, we get a desired number of highest-scoring concepts. Thus, the query intent can be predicted according to the sum of the intent probabilities for the top M highest-scoring concepts. The detailed algorithm is shown:

Intent Predictor:

Input: query $\boldsymbol{q}$, the number of snippets $\boldsymbol{K}$, the number of selected concepts $\boldsymbol{M}$, intent probability vector of concepts $\boldsymbol{V}$, and the threshold of the specific intent θ

Output: *Has intent or not* (***True*** *or* ***False***)

1: *SnippetVector* ← Query Live Search (***q***, ***K***)
2: **for** each *snippet* in *SnippetVector*
3: *SentencesVector* ← sentences(*snippet*)
4: **for** each *sentence* in *SentencesVector*
5: *TextVector* ← *tfidf*(sentence)

[1] http://search.live.com

```
6:        for each c in Wikipedia concepts
7:            Score(c) ← BM25(TextVector,Vector(c))
8:        endfor
9:        Let GeneratedConcepts be a set of top 10 concepts
          with the highest max-min normalized Score(c)
10:       for each c in GeneratedConcepts
11:           ConceptsScoreVector(c) += normalized Score(c)
12:       endfor
13:   endfor
14: endfor
15: Let SelectedConcepts be a set of concepts containing top M
    highest ConceptsScoreVector(c) and sumProb = 0
16:   for each s in SelectedConcepts do
17:       sumProb += V(s)
18:   endfor
19:   If sumProb > θ return True; Else return False.
```

4. Experiments

In this section, we first define three kinds of query intent applications and then we show the results of several experiments that demonstrate the effectiveness of our proposed method on these three intent applications.

4.1 Intent Applications

The query intent identification discussed in this work is to make a judgment regarding whether a query contains a specific intent we defined. In this work, we evaluate our algorithm on three applications, travel intent, job intent, and personal name intent identification, our approach is general enough to be applied to other applications as well.

Travel Intent Identification. According to the definition from Wikipedia, "Travel is the change in location of people on a trip, or the process of time involved in a person or object moving from one location to another" (http://en.wikipedia.org/wiki/travel). Travel is a complex social activity that interacts with various services including agency services, such as travel agency and visa application, transportation services, such as airlines and taxis, accommodation services, such as hotels and entertainment venues, and other hospitality industry services, such as resorts. Therefore, we denote a query to have travel intent if it is directly or indirectly related to the services mentioned above. Query log analysis shows that about 3%-4% of distinct web search queries have travel intent.

Personal Name Intent Identification. Our definition of personal name intent identification is a little different from personal name classification defined in [15], which classifies a query as a personal name query only when the query as a whole is for a personal name. We denote a query to have personal name intent if the query is a personal name or contains a personal name. For example, for the query "john smith pictures", we consider this query has personal name intent. Personal names are an important kind of Web query in Web search. As reported in [15], about 30% of daily web queries have personal name intent.

Job Intent Identification. We denote a query to have job intent if the user is interested in finding job-related information including employment compensation, employment laws, occupations, job web sites, etc. For example, "trucking job", "part-time job", and "monster.com" are queries that have job intent whereas "Steve Jobs" does not. We also treat queries, such as "resume" and "sample cover letters," as having job intent since they are indirectly related to jobs. As reported in [14], about 0.2%-0.4% of distinct web search queries have job intent.

4.2 Data preparation

4.2.1 Wikipedia data

As an open source project, Wikipedia content is readily downloadable from http://download.wikipedia.org. The data is available in the form of database dumps that are released periodically, from several days to several weeks apart. The version used in our experiments was released on March 12, 2008. We identified over four million distinct entities that constitute the vocabulary of Wikipedia concepts. These were organized into 337,960 categories with an average of two subcategories and 26 articles in each category. The articles themselves are highly inter-linked; each links to an average of 25 other articles.

4.2.2 Query log data

A one-month search query log from June 2007 of Live Search was used. After cleaning the data by only keeping the frequent[2], well-formatted, English queries, we had a total of 2,614,382 distinct queries.

Table 1. Labeled seed queries and the number of selected seed concepts

Intent Type	Seed queries	The number of seed queries
Travel Intent	Travel; Hotel; Tourism; airline tickets; Expedia	2389
Personal Name Intent	Britney Spears; David Beckham; George W. Bush	10K
Job Intent	Employment; monster; career	2543

4.2.3 Seed datasets

As mentioned in the methodology section, we need to manually label a small set of seed queries in order to find some Wikipedia concepts and categories strongly related to the query intent we want to define. Since we only need a few seed queries for each query intent classifier, we can label some representative queries according to common knowledge and then map the labeled queries to Wikipedia concepts. By browsing the categories to which the seed queries belong, their sibling concepts, and the concepts they link to, we can easily collect a large number of representative Wikipedia concepts and categories as a set of seed concepts. For example, if we want to collect seed concepts for the personal name intent, we can first use the names of some well-known people, such as "Britney Spears", as search terms in Wikipedia. After we get the mapped Wikipedia concept, through browsing the parent category, e.g., "Living People", we can easily collect a large number of people's names as seed concepts. (In this case, we can get more than 5,000 people's names.) The labeled seed queries and the number of selected seed concepts used in our experiments are shown in Table 1.

4.2.4 Evaluation datasets

To measure classification performance, we separately prepared evaluation sets for human annotators to label and ensured that queries in these evaluation sets do not overlap with any seed queries. For each query intent identification application, we randomly selected queries in the query log. After filtering the

[2] In our experiments the frequency of a query is greater than 20.

ambiguous queries based on Wikipedia disambiguation pages, we asked two annotators with good English language skills to judge whether the selected query has the evaluated intent. We only kept the queries for which two annotators had the same judgment. In each application, a user interface was created for human annotators where each query was presented along with retrieval results from two major search engines. Human annotators looked at both results to make a binary judgment. The number of labeled evaluation sets is shown in Table 2.

Table 2. Number of labeled evaluation sets

Intent Type	Number of sets
Travel Intent	2466 (27% positive)
Personal Name Intent	2260 (12% positive)
Job Intent	2361 (22% positive)

4.3 Evaluation Metrics

We utilize the standard measures to evaluate the performance of the three intent classifiers, i.e., precision, recall and F_1-measure [16]. Precision (*P*) is the proportion of actual positive class members returned by our method among all predicted positive class members returned by our method. Recall (*R*) is the proportion of predicted positive members among all actual positive class members in the data. F_1 is the harmonic average of precision and recall which is defined as $F_1 = 2PR / (P + R)$.

4.4 Algorithms Compared and Results

4.4.1 Comparison algorithms

To demonstrate the effectiveness of our algorithm, we compared the four methods described below in our experiment.

1. **Supervised learning with seed concepts**. We trained a classifier using logistic regression based on seed concepts selected from Wikipedia. We denote this method as "*LR*".
2. **Supervised learning with concepts expanded with Wikipedia**. We used a method similar to [14] except that we expanded the seed concepts with the Wikipedia link graph, not the query-URL graph. We set $K = 5$, $M = 10$ in the intent predictor part and the random walk runs for 100 iterations. After the random walk, we selected those concepts with high intent probability (in the experiment we select concepts of which the intent probability is bigger than 0.00004). Then, we trained a classifier using logistic regression on selected concepts. We denote this method as "*LRE*".
3. **Our method**. $K = 5$, $M = 10$ in the intent predictor part and the random walk runs for 100 iterations. We denote this method as "*WIKI*".
4. **Our method without the random walk step.** To demonstrate the importance of the random walk step in our method, we remove the random walk step in this algorithm. The parameter setting is the same as "*WIKI*" and we denote this method as "*WIKI-R*".

The features used in the first and second algorithms are *n-grams* of the training query content. In all three query intent applications, we used the *n-grams* feature with n = 1,2. We implement logistic regression using limited memory optimization (L-BFGS) [28]. As shown in recent studies, training using L-BFGS gives good performance in terms of speed and classification accuracy.

As the supervised methods *LR* and *LRE* need negative samples to train a binary classifier, we asked the two annotators to label negative samples, which are queries that do not have the defined intent in the query log for each intent application. The number of labeled negative data samples is comparable to that of positive data samples.

4.4.2 Experiment Results

In this section, we present the experimental results of the four compared algorithms. For *WIKI* and *WIKI-R*, we need to tune the threshold θ which is used to judge whether a query has the specific intent or not to get the best performance on a certain data set. For the two algorithms based on supervised learning, in order to make a trade-off between precision and recall, and reach the best F_1, we also need a threshold of probability. Therefore, we randomly split the evaluation dataset into validation set and tuning set according to a 4:1 ratio. Based on the tuning set, we can select the intent score threshold θ where our method reaches the best performance in terms of F_1. Then we report the performance of our method on the validation set with the same threshold. Similarly, for the two supervised baseline methods, we can also obtain a threshold of probability based on the tuning set, and then get the performance on the validation set with the obtained threshold.

In Table 3, Table 4, and Table 5, we report the precision, recall and F_1 value of the four methods on the three kinds of query intent identification tasks we studied. We also report the precision and recall value for both positive and negative samples in the validation data. As shown in these tables, *WIKI and WIKI-R* significantly outperformed the *LR* and *LRE* methods in all four intent applications based on the F_1 measure and *WIKI* achieved more than 90% F_1 on all three applications. After expanding the positive samples with a random walk on the Wikipedia link graph, the performance of the intent classifier *LRE* is significantly improved compared with the *LR* classifier, which is only trained on seed concepts.

Table 3. Comparison of the four methods on travel intent identification

	Positive		Negative		Overall		
	Pre	Rec	Pre	Rec	Pre	Rec	F1
LR	0.802	0.377	0.94	0.277	0.888	0.305	0.454
LRE	0.733	0.513	0.925	0.541	0.835	0.527	0.6462
WIKI	**0.939**	**0.909**	**0.961**	**0.941**	**0.954**	**0.933**	**0.943**
WIKI-R	0.85	0.618	0.944	0.848	0.922	0.785	0.848

Table 4. Comparison of the four methods on personal name intent identification

	Positive		Negative		Overall		
	Pre	Rec	Pre	Rec	Pre	Rec	F1
LR	0.909	0.622	0.924	0.645	0.921	0.639	0.7545
LRE	0.874	0.866	0.826	0.804	0.838	0.819	0.8284
WIKI	**0.938**	**0.876**	**0.935**	**0.899**	**0.935**	**0.897**	**0.9156**
WIKI-R	0.924	0.69	0.939	0.783	0.937	0.772	0.8465

Table 5. Comparison of the four methods on job intent identification

	Positive		Negative		Overall		
	Pre	Rec	Pre	Rec	Pre	Rec	F1
LR	0.609	0.488	0.901	0.256	0.77	0.309	0.441
LRE	0.635	0.557	0.917	0.358	0.806	0.403	0.5373
WIKI	**0.923**	**0.888**	**0.961**	**0.932**	**0.953**	**0.922**	**0.9372**
WIKI-R	0.887	0.672	0.954	0.856	0.941	0.815	0.8735

Comparing *WIKI* with *WIKI-R*, we can see that *WIKI* is significantly better than *WIKI-R*, which indicates that the random walk step is quite important in our method. Through random walk on the constructed Wikipedia link graph, we can propagate the intent of labeled seed concepts to other Wikipedia concepts and categories in the graph iteratively, which helps us identify more concepts and categories with that intent. Specifically, if we compare the precision and recall of *WIKI* and *WIKI-R* on positive and negative samples, we find that the performance of *WIKI-R* on negative data is relatively better than that on positive data. This is largely due to the fact that the positive seed concept set was relatively small with respect to the large variety of queries, resulting in a poor intent predictor based on *ESA* and hence an unreliable prior, while the set of negative concepts (Wikipedia concepts minus seed concepts) was quite large, but noisy (it contains a great many unlabeled concepts that have the intent), which also decreases the intent predictor performance on negative data to some extent.

Another observation from these tables is that *LR* and *LRE* did not work well for the query intent identification tasks. *LR* and *LRE* are supervised methods that require a large number of training data to ensure good generalization ability. However, a query is often quite short (most queries have less than three words) and the features based on n-grams of query words obviously suffer seriously from the feature sparseness problem, which affects their performance. Moreover, queries with a specific intent only consist of a small part of online queries. Since we cannot collect a full set of data samples, this causes the classifier to be unpredictable when faced with queries whose constructed features do not exist in the training feature set. This observation further confirms the three challenges we addressed in the introduction part.

4.4.3 Performance on unseen queries

To validate the performance of our intent predictor when faced with new queries not covered by Wikipedia concepts, we separated the queries into two parts: those that are covered by Wikipedia concepts and those that are not covered by Wikipedia concepts, and compared the performance of the two parts. As shown in Table 6, our method works almost as well for queries no matter whether they are covered by Wikipedia concepts or not, which means that our method has a good generalization ability for new queries that are not covered by Wikipedia concepts. This is a quite important characteristic for a query intent classifier.

Table 6. Comparison of the performance of our method for queries that are/are not covered by Wikipedia concepts on the three intent identification tasks

	Covered by Wikipedia				Not Covered by Wikipedia			
Intent Type	**Num**	**Pre**	**Rec**	**F1**	**Num**	**Pre**	**Rec**	**F1**
Travel	993	0.968	0.954	0.9609	980	0.941	0.912	0.92627
Personal Name	849	0.942	0.899	0.92	959	0.93	0.896	0.91268
Job	953	0.984	0.956	0.9698	936	0.921	0.889	0.90472

4.4.4 Case Study

To illustrate the effectiveness of Wikipedia representation for new queries that are not covered by Wikipedia concepts, we show the generated Wikipedia concepts for some sample queries. In Table 7, we present three query examples and their top 6 generated concepts with intent probability for the job, travel and personal name intent applications. From these tables, we can see that the generated concepts based on *ESA* are quite related to the queries and the intent probability associated with these concepts can accurately reflect their intent tendency. This explains why our intent predictor can judge new queries with good accuracy.

Table 7. New query examples and their generated concepts with intent of the three intent applications

employment guide	
Employment website:1.034e-4	Job hunting:0.871e-4
Job search engine:0.901e-4	Eluta.ca:0.969e-4
CareerLink:0.64e-4	Types of unemployment:0.788e-4
ei canada	
Unemployment benefit:0.609e-4	Workers' compensation:0.588e-4
Corporate-owned life insurance:0.021e-4	Social programs in Canada:0.556e-4
Paid Family Leave:0.719e-4	Taxation in Canada:0.108e-4
job builder	
Job search engine:0.907e-4	CareerBuilder:0.729e-4
JobServe:0.643e-4	Eluta.ca:0.969e-4
Falcon's Eye (BBS door):0.000e-4	Monster (website):0.609e-4

(a) Job intent example queries

vacation reviews	
TripAdvisor:0.869e-4	SideStep:0.987e-4
Golf resort:0.844e-4	Hotwire.com:0.983e-4
Conde Nast Traveler:0.587e-4	Victoria Clipper:0.055e-4
cheapfares	
Advanced Booking Charter:0.403e-4	Travel agency:0.931e-4
Young Persons Railcard:0.021e-4	Priceline.com:0.532e-4
Hotwire.com:0.982e-4	kayak.com:1.05e-4
bellagio casino	
Bellagio (hotel and casino):0.735e-4	Steve Wynn (developer):0.161e-4
Dunes (hotel and casino):0.905e-4	Boardwalk Hotel and Casino:0.783e-4
Bellagio:0.246e-4	Mirage Resorts:0.396e-4

(b) Travel intent example queries

harry shum	
Lydia Shum:0.825e-4	Under the Canopy of Love:0.421e-4
Mina Shum:0.995e-4	Itzhak Shum:0.854e-4
Shamash-shum-ukin:1.04e-4	Forest of Death (film):0.319e-4
toby walker	
Fear (film):0.461e-4	Annie Oakley :0.803e-4
Brian Froud:0.955e-4	Khe Sanh (song) :0.865e-4
Clearlake (band):0.151e-4	Bob Fass:0.757e-4
lee-feng chien	
Brave Archer:1.12e-4	List of Taiwanese people:0.946e-4
Chien-Ming Wang:1.1e-4	Lists of Chinese people:1.01e-4
List of Taiwanese footballers:0.991e-4	Qi (section Feng shui):0.827e-4

(c) Personal name intent example queries

5. RELATED WORK

5.1 Query Classification

The works on query classification can be categorized into two main research directions: one is enriching the feature representation of queries and the other is augmenting labeled training samples.

The 2005 KDD Cup competition on web query classification inspired research works focusing on enriching queries using Web search engines and directories [5] [6] [9] [13]. Shen *et al.* [5] [6] proposed to utilize search result snippets as query features and built

classifiers using an intermediate document taxonomy based on ODP. Classifications in the document taxonomy of the enriched query were mapped to those in the target taxonomy. Broder *et al.* [3] also used search result snippets as query features and directly classified the query features with the target taxonomy. Dai *et al.* [13] proposed a method to detect a query's online commercial intent by training a binary classifier on a large amount of labeled queries enriched with search result snippets. Compared with these approaches, we also use search result snippets for enriching query features, but we do not need to train a document classification model based on the target taxonomy.

To augment labeled training samples for query classification, Beitzel *et al.* [1] proposed a method for automatic web query classification by leveraging unlabeled data within a semi-supervised learning framework. Li *et al.* [14] described a semi-supervised learning approach to query intent classification with the use of search click graphs. They infer the classes of unlabeled queries from those of labeled ones based on their proximities in a click graph. However, our method does not utilize a query log or search-click-through log as unlabeled data for training the classifier.

5.2 Wikipedia Mining

Previously, Wikipedia has been exploited for lexicon acquisition [23] [24], information extraction [10] [12], taxonomy building [10], entity ranking [19] [20] and text categorization [22] [26]. Gabrilovich *et al.* [22] tried to apply feature generation techniques on Wikipedia to create new features that augment a bag of words. Experiments on various datasets indicate that the newly generated features can significantly improve the text classification performance. Later, Gabrilovich *et al.* [23] proposed *explicit semantic analysis (ESA)*, which utilizes Wikipedia for feature generation for the task of computing semantic relatedness of words. The resulting relatedness measures are much better than WordNet and Latent Semantic Analysis based on human judgments. Pu *et al.* [26] and Hu *et al.* [27] proposed to extract concept relations from Wikipedia and utilized the extracted relations to improve text classification and clustering. Their reported results confirm that relations in Wikipedia can enhance text classification and clustering performance. Ruiz *et al.* [24] further used Wikipedia to extract lexical relationships to extend WordNet.

Bunescu *et al.* [10] and Cucerzan *et al.* [12] explored Wikipedia's articles and rich link structures to disambiguate named entities of different documents. This is an important task of information extraction since different entities often share the same name in reality. Strube *et al.* [10] proposed some methods to derive a large-scale taxonomy from Wikipedia. Zaragoza *et al.* proposed to utilize Wikipedia for the task of entity type ranking [19] and later proposed a method to predict the more important entity type relevant to a query in an informational search task [20]. However, the entity type in their works is restricted to only Location, Date time, and Organization, which are heavily dependent on external name entity tagging tool. Toral *et al.* [3] extracted named entity lexicons, such as People, Location, Organization, etc. from Wikipedia.

To the best of our knowledge, this paper is the first work to utilize knowledge from Wikipedia to infer a user's query intent.

6. CONCLUSION AND FUTURE WORK

In this work, we utilize the world's largest knowledge resource, Wikipedia, to help understand a user's query intent. Our work differs from previous works on query classification in that we aim to use a human knowledge base to identify the query intent, and we do not collect large quantities of examples to train an intent classifier. This approach allows us to minimize the human effort required to investigate the features of a specified domain and understand the users' intent behind their input queries. We achieve this goal by mining the structure of Wikipedia and propagating a small number of intent seeds through the Wikipedia structure. As a result, each article and each category in Wikipedia is assigned an intent probability. These intent concepts are directly used to identify the intent of the input query and the experimental results demonstrate that our algorithm achieves much better classification accuracy than other approaches.

In the future, we would like to explore more intent classifiers for other applications and integrate them to provide a better search service that combines the results from multiple vertical search engines. Furthermore, it would be interesting to apply our approach to the intent identification problem in multi-lingual environments so that the algorithm can understand intents for other languages.

7. REFERENCES

[1] S. Beitzel, E. Jensen, O. Frieder, D. Lewis, A. Chowdhury, and A. Kolcz. Improving automatic query classification via semi-supervised learning. In *Proc. of the 5th IEEE International Conference on Data Mining(ICDM-05)*, 2005.

[2] Robertson, S., Zaragoza, H. and Taylor, M., Simple BM25 extension to multiple weighted fields. In *Proc. of the Thirteenth ACM Conference on Information and Knowledge Management (CIKM-04),* 2004.

[3] A. Broder, M. Fontoura, E. Gabrilovich, A. Joshi, V. Josifovski, and T. Zhang. Robust classification of rare queries using web knowledge. In *Proc. of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-07)*, July 2007.

[4] Toral, A. and Munoz, R., A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. In *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics(EACL-06),*2006.

[5] D. Shen, J. Sun, Q. Yang, and Z. Chen. Building bridges for web query classification. In *Proc. of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-06)*, 2006.

[6] D. Shen, R. Pan, J. Sun, J. Pan, K. Wu, J. Yin, and Q. Yang. Q2C@UST: Our winning solution to query classification in KDDCUP 2005. In *SIGKDD Explorations, volume 7*, pages 100-110. ACM, 2005.

[7] Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C. and Nevill-Manning, C.G., Domain-specific keyphrase extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99),* 1999.

[8] Schonhofen, P., Identifying document topics using the Wikipedia category network. In *Proc. of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI-06),* 2006.

[9] D. Vogel, S. Bickel, P. Haider, R. Schimpfky, P. Siemen, S. Bridges, and T. Scheffer. Classifying search engine queries using the web as background knowledge. In *SIGKDD Explorations, volume 7.* ACM, 2005

[10] S Strube, M. and Ponzetto, S.P., Deriving a large scale taxonomy from Wikipedia. In *Proc. of the Twenty-Second National Conference on Artificial Intelligence* (*AAAI-2007),* 2007.

[11] Bunescu, R. and Pasca, M., Using encyclopedic knowledge for named entity disambiguation. In *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics(EACL-06),*2006

[12] Cucerzan, S., Large-scale named entity disambiguation based on Wikipedia data. in *Proc. of the 2007 Conference on Empirical Methods in Natural Language Processing (EMNLP-07)*.

[13] Honghua (Kathy) Dai, Lingzhi Zhao, Zaiqing Nie, Ji-Rong Wen, Lee Wang, Ying Li: Detecting online commercial intention (OCI). *In Proc. of the 15th World Wide Web Conference (WWW-06)*, 2006.

[14] Xiao Li, Ye-Yi Wang, Alex Acero: Learning query intent from regularized click graphs. In *Proc. of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-08)*, 2008.

[15] Dou Shen, Toby Walkery, Zijian Zheng, Qiang Yang, Ying Li: Personal name classification in web queries. In *Proc of the First ACM International Conference on Web Search and Data Mining (WSDM-08)*, 2008.

[16] C. J. van Rijsbergen. Information Retrieval. *Butterworths, London, second edition,* 1979.

[17] Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C. and Nevill-Manning, C.G. "KEA: Practical automatic keyphrase extraction". In *Proc. of The Fourth ACM Conference on Digital Libraries*, 1999.

[18] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2):207–227, 2000.

[19] Hugo Zaragoza, Henning Rode, Peter Mika, Jordi Atserias, Massimiliano Ciaramita, Giuseppe Attardi: Ranking very many typed entities on Wikipedia. In *Proc. of the Sixteenth ACM Conference on Information and Knowledge Management (CIKM-07)*, 2007.

[20] David Vallet, Hugo Zaragoza: Inferring the most important types of a query: A semantic approach. In *Proc. of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-08)*, 2008.

[21] J.H. Lee: Combining multiple evidence from different properties of weighting schemes. In *Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-95)*, 1995.

[22] Gabrilovich, E. and Markovitch, S., Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proc. of the Twenty-First National Conference on Artificial Intelligence* (*AAAI-2006),* 2006.

[23] Gabrilovich, E. and Markovitch, S., Computing semantic relatedness using Wikipedia based explicit semantic analysis. In *Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07),* 2007.

[24] Ruiz-Casado, M., Alfonseca, E., and Castells, P., Automatic extraction of semantic relationships for WordNet by means of pattern learning from Wikipedia. In *Proc of the 11th International Conference on Applications of Natural Language to Information Systems (NLDB2006)*, 2006.

[25] Strube, M. and Ponzetto, S.P., WikiRelate! Computing semantic relatedness using Wikipedia. In *Proc. of the Twenty-First National Conference on Artificial Intelligence* (*AAAI-2006),* 2006.

[26] Pu, W., Jian, H., Hua-Jun, Z., Zheng, C., Improving text classification by using encyclopedia knowledge. In *Proc. of the 7th IEEE International Conference on Data Mining(ICDM-07)*, 2007.

[27] Jian Hu, Lujun Fang, Yang Cao, Hua-Jun Zeng, Hua Li, Qiang Yang, Zheng Chen: Enhancing text clustering by leveraging Wikipedia semantics. In *Proc. of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-08)*, 2008.

[28] D. Liu and J. Nocedal. On the limited memory BFGS method for large-scale optimization. *Mathematical Programming*, 45:503–528, 1989.