

Advertising Keyword Suggestion Based on Concept Hierarchy

Yifan Chen

Shanghai Jiao Tong University
Shanghai, 200240, China
ivan@apex.sjtu.edu.cn

Gui-Rong Xue

Shanghai Jiao Tong University
Shanghai, 200240, China
grxue@apex.sjtu.edu.cn

Yong Yu

Shanghai Jiao Tong University
Shanghai, 200240, China
yyu@apex.sjtu.edu.cn

ABSTRACT

The increasing growth of the World Wide Web constantly enlarges the revenue generated by search engine advertising. Advertisers bid on keywords associated with their products to display their ads on the search result pages. Keyword suggestion methods are proposed to fill the gap between the keywords chosen by advertisers and the popular queries, through finding new relevant keywords according to some statistical information (for example, the keyword co-occurrence). However, there is little effort taking semantic information, such as concept hierarchy, into account. In this paper, we propose a novel keyword suggestion method that fully exploits the semantic knowledge among concept hierarchy. Given a keyword, we first match it with some relevant concepts. Then the relevant concepts are used with their hierarchy to fertilize the meanings of the keywords. Finally new keywords are suggested according to the concept information rather than the statistical co-occurrence of the keyword itself. Experimental results show that our proposed method can successfully provide suggestion that meets the accuracy and coverage requirements.

Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval;

General Terms

Algorithms, Experimentation

Keywords

Keyword Suggestion, Concept Hierarchy, Advertising

1. INTRODUCTION

Search engine advertising is a successful business model in internet advertising based on keyword targeting technology. Advertisers create ads and bid on keywords that are related to their business. The ads are then displayed on the search result pages when users are searching for corresponding keywords. Since the total number of queries is extremely large, only a handful are bid on by advertisers. It is quite common that advertisers miss a lot of popular keywords [15], resulting in a gap

between the keywords chosen by the two most important parts of the business: advertisers and their potential customers.

To overcome this problem, *keyword suggestion* technology was employed to help advertisers to find more appropriate keywords. It involves discovering new words or phrases related to the existing keywords. Most keyword suggestion products in existence try to solve the problem by utilizing statistical information. For example, some keyword suggestion tools look for co-occurring phrases in search engine query logs [11].

Those methods have gained prevalence based on the assumption that the statistical co-occurrence indicates keyword relevance [22]. Actually, high co-occurrence between two keywords means they are semantically related. Therefore, these methods adopt indirect ways to discover the semantic similarity. However, these methods do not take semantic information, such as concept hierarchy, into account directly, which may improve the overall performance.

Two additional problems with traditional keyword suggestion methods are the low coverage and the lack of disambiguation ability. In some cases, two relevant keywords never occur with each other. They will not be found by the traditional means. In other cases, a keyword may have more than one meaning. A very famous example is that “apple” has at least two meanings: it can be either a computer company or a kind of fruit. The relevant phrases of apple for these two meanings are obviously different. The traditional method cannot distinguish between them and the suggested keywords may be a mixture of both meanings.

In this paper, we propose to suggest keywords with direct help of concept hierarchy, a kind of useful semantic information, to improve the accuracy and coverage of keyword suggestion. We try to suggest new keywords according to advertisers’ real objectives rather than the query keywords. Given a keyword, we first match it with some relevant concepts. Then the relevant concepts with their hierarchy are used to fertilize the meanings of the keywords. Finally new keywords are suggested according to the concept information rather than the statistical co-occurrence of the keyword itself.

The performance of our approach is directly affected by the following factors:

1. The quality and the coverage of the concept hierarchy.
2. The precision of associating keywords with their relevant concepts.
3. The accuracy of describing the different meanings of a keyword.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSDM’08, February 11–12, 2008, Palo Alto, California, USA.
Copyright 2008 ACM 978-1-59593-927-9/08/0002...\$5.00.

To ensure the accuracy of the concept hierarchy, we derive it from a high-quality manually-defined web directory, such as the Open Directory Project (ODP, [18]). This concept hierarchy involves the definition of concept relationships (i.e. the taxonomy structure of the hierarchy) and concept content (i.e. the meaning of each concept). The categories in the web directory can be treated as concepts. Therefore, the relationship between the categories indicates the concept relationships. Furthermore, phrases extracted from the web pages that are organized into categories can be used to describe the concept contents. The coverage of concept hierarchy is very important for commercial use to meet the needs of various customers. As a research topic, this issue is not as important and so we will not focus on it.

To guarantee the accuracy of associating keywords with their relevant concepts, we develop an algorithm to gather the relevant keywords for each of the concepts and propose some weight criteria to rank them.

To ensure the accuracy of describing the meaning of a keyword, we develop a probabilistic framework to rank the similarity of different concepts, which makes it possible to find the most relevant and representative ones.

Furthermore, with the concepts associated with the phrases, we find a way to estimate the similarity between the phrases via the related concepts. This way keyword suggestion could then be performed based on the similarity.

We conduct experiments to evaluate the effectiveness of the proposed keyword suggestion work and the accuracy of the categorization of relevant concepts. The experiment results prove that concept hierarchy based keyword suggestion can perform better than the co-occurrence based methods. It also shows that the categorized keyword suggestion result is acceptable.

The contributions of our work can be summarized as follows:

- We propose the novel approach of suggesting keywords according to advertisers' objectives mapped to concepts rather than the keyword itself.
- The idea of categorizing the distinguishing concepts is proposed for disambiguation and avoidance of interference.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 paints a general picture of the suggestion approach. Section 4 clarifies the construction of the concept hierarchy. Section 5 describes the keyword suggestion method. The evaluation of our keyword suggestion method is shown in Section 6. Finally, we conclude the effort and discuss further research possibilities in Section 7.

2. RELATED WORK

With the growth of web advertising, the demand for presenting advertisements via web elements has fueled research in related topics. There is a general emphasis on the importance of relevant association in advertising, and how the effect of advertising is constrained by the media contents. Recent work has paid more attention to keyword-related advertising from the publishers' perspective. For example, to match relevant ads to web pages, Ribeiro-Neto et al. [21] proposed several strategies and a term expansion method to fill up the gap between the ads and the web pages. Later, Yih et al. [31] tried to refine the keywords extracted from web pages directly. The keyword suggestion method, on the

other hand, aims at assisting the advertisers by ensuring the matching effectiveness.

Currently, the technologies used by keyword suggestion tools available online can be broadly divided into the following types: query-log and advertiser log mining, proximity searches, and meta-tag crawlers.

The keyword suggestion tools benefit a lot from search engines. Some query log based methods like [3] try to find new terms that co-occur with the seed query. Google's Adword Tools [11] and Overture's Keyword Selection Tool [19] recommend the popular queries that contain the initial keyword. Other tools that use proximity-based approaches send seed keywords to the search engine and expand new suggestions with words in its proximity. These technologies have the same limitation of failing to suggest keywords that do not contain the seed. To overcome this problem, some tools like Adword mine advertisers' logs. Such methods present new keywords which were searched for by other advertisers that concentrated on the seed keyword. However, this method cannot guarantee that its suggestion is really related to the seed query. Another method focuses on extracting new keywords from meta-tags. Many high ranked websites include relevant keywords in meta-tags. WordTracker [29] uses meta-tag spiders to search seed keywords and make suggestions for meta-tag words based on highly ranked web pages.

Recent work has raised attention about mining semantic relationships between terms for suggestions. Joshi and Motwani [15] present TermsNet, which leverages search engines to determine relevance between terms and capture their semantic relationships as a directed graph. Later, Abhishek [1] uses a web based kernel function to establish semantic similarity between terms. Jones et al. [14] modified the original query based on typical substitutions web searchers make to find semantically related ones.

A similar problem to keyword suggestion is query expansion ([5], [20], [30]), which has already been studied for many years. A similar system [6] was proposed to cluster the advertiser-keyword data into topics.

Query expansion is employed when users want additional keywords to obtain relevant documents and filter the irrelevant ones. On the other hand, those who use advertising keyword suggestion systems want their advertisements to appear in more relevant situations. From this perspective, their goals are different from each other. Another difference is that query expansion only needs two to three new keywords to refine the results whereas keyword suggestion requires dozens to hundreds. Despite the fact that the two problems do not share the same goal, query expansion methods still inspire our work.

Much work has focused on the utility of using a concept hierarchy. Related work can be roughly divided into the following stages: hierarchy construction, content acquirement and application.

The construction of the concept hierarchy aims at finding the concepts and the relationships between them. Some work utilizes existing concept hierarchies or a hierarchically organized knowledge base, such as [7] and [26]. Others try to derive a concept hierarchy from document collections ([22], [17]). To acquire the content of concepts, many technologies have been proposed. Some people employed search engines to gather the content automatically like [12]. In some traditional ways such as

classification, the idea of feature selection ([16], [27]) was introduced to improve the performance. The major usage of concept hierarchy is to mine semantic similarities ([8]) or to distinguish meanings for classification ([10], [25]).

One of the most relevant applications of concept hierarchy is to categorize different word senses ([23]), which is also an important contribution in our work. The approach in our work has been taken similarly in Word Sense Disambiguation to address the data sparseness problem (such as in [4]).

Another relevant application is to compute the similarity of element sets in a hierarchy. Ganesan et al. [9] introduced a generalized vector space model to calculate the cosine similarity between two document sets with smoothing along the hierarchical domain structure.

3. FRAMEWORK

Our general approach, as described in the introduction, focuses on making suggestions based on the users' real need, which can be represented by concepts. A concept hierarchy is then constructed to represent the interrelationship between the concepts. With the help of such a concept hierarchy, we can pay more attention to the semantic information of the keyword when performing suggestions.

The keyword suggestion problem can be defined as following: Given a query phrase q , we wish to suggest a list of new phrases $\{t_i\}$ according to the similarity function $sim(q, t_i)$. Here $sim(q, t)$ measures the relevance between q and t . We then define the document set D_x to be the set of documents that a certain phrase x appears in. As a result, the similarity between the phrases $sim(q, t)$ could be acquired from the similarity between their document sets $sim(D_q, D_t)$. There are a number of methods proposed to measure the similarity of the sets. One of the most commonly used similarity measurement, the Jaccard coefficient, is defined as the size of the intersection divided by the size of the union of the sample sets:

$$sim(q, t) = \frac{|D_q \cap D_t|}{|D_q \cup D_t|} = \frac{\sum_d f_d(q)f_d(t)}{\sum_d (1 - (1 - f_d(q))(1 - f_d(t)))}$$

Where $f(x) = \begin{cases} 1, & \text{if phrase } x \text{ appears in document } d \\ 0, & \text{if phrase } x \text{ does not appear in document } d \end{cases}$

The similarity is then being represented as commonality divided by coverage. The commonality shows how much they share while the coverage shows how much they are different besides the common part.

When taking concept hierarchy into account, we generalize the original function into the following form:

$$sim(q, t) = \frac{\sum_c f_c(q)f_c(t)w_c(q)}{\sum_c (1 - (1 - f_c(q))(1 - f_c(t)))w_c(q)}$$

In this paper, we use the proportion of the phrase in the concept to represent the confidence, that is, $f_c(x) = \frac{|D_{c,x}|}{|D_c|}$. Here $D_{c,x}$ denotes the document set within the concept c that phrase x occurs, the size of the set is also known as document frequency (DF). D_c denotes all the documents in the concept c . The weight of the

concept $w_c(q)$ reflects the importance of a concept c to a query phrase q . The above similarity function does not take the difference of each concept into account. In our assumption, the following factors will also affect the importance.

1. Size. Given two concepts with the same proportion of a phrase, the greater size will result in greater importance.
2. Phrase distribution. The concept is important if the phrase is widely spread in the sub-concepts. From the information theory's perspective, such concept contains more information.

Moreover, we approximate the coverage with a function that only related to the candidate phrase. The similarity function will be simplified into:

$$\begin{aligned} sim(q, t) &= \frac{\sum_c f_c(q)f_c(t)w_c(q)}{coverage(t)} \\ &= \sum_c (f_c(q)w_c(q)) \left(\frac{f_c(t)}{coverage(t)} \right) \\ &= \sum_c P(q \rightarrow c)P(c \rightarrow t) \\ &= \sum_{c \in C_q} P(q \rightarrow c)P(c \rightarrow t) \end{aligned}$$

Where $P(q \rightarrow c)$ and $P(c \rightarrow t)$ are two weight functions only related to a single phrase and C_q denotes the concepts in which the phrase q occurs.

Our framework is divided into two parts: the construction of the concept hierarchy, and the keyword suggestion based on concept hierarchy. Figure 1 is an illustration about the framework.

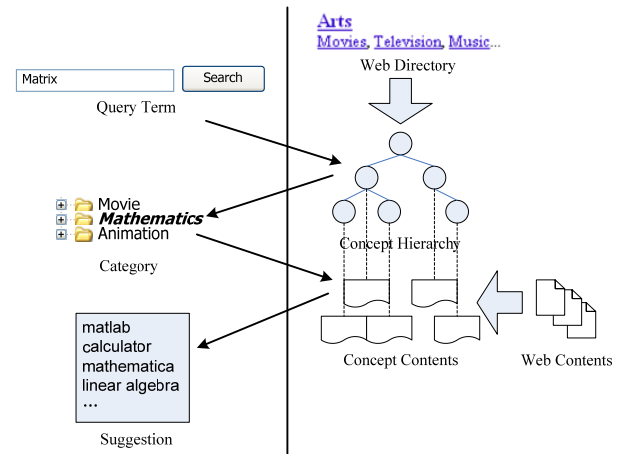


Figure 1. Illustration of keyword suggestion framework

We derive the concept hierarchy from a web directory for good coverage of the various aspects of real life. A web directory contains a large set of web pages that are well categorized. The categories from a web directory can be considered as concepts and the directory structure can be considered as concept relationships. To represent the meanings of the concepts, contents are then built in the form of bags of phrases.

Concept #43037		Top/Computers/Computer_Science	
1. computer science	0.12	15. distributed systems	0.02
2. department of computer science	0.05	20. software engineering	0.01
3. computer	0.04	21. computational complexity	0.01
4. computing	0.03	22. complexity	0.01
5. university	0.03	23. programming languages	0.01
6. computer science department	0.03	24. database systems	0.01
7. department	0.03	26. complexity theory	0.01
8. research	0.03	27. quantum	0.01
9. science	0.03	29. algorithms	<0.01
10. theoretical computer science	0.02	32. quantum information	<0.01

Concept #43056		Concept #84417		Concept #259003	
Top/Computers/Computer_Science/People		Top/Computers/Computer_Science/Academic_Departments		Top/Computers/Computer_Science/Theoretical	
1. computer science	0.08	1. computer science	0.25	1. complexity	0.09
2. programming languages	0.04	2. department of computer science	0.14	2. computational complexity	0.08
3. university	0.03	3. computer	0.10	3. quantum	0.08
4. university of edinburgh	0.03	4. department	0.09	4. complexity theory	0.08
5. software engineering	0.03	5. computer science department	0.08	5. computer science	0.08
6. database systems	0.03	6. science	0.07	6. quantum information	0.07
7. indian institute	0.02	7. computing	0.05	7. theoretical computer science	0.07
8. algorithms	0.02	8. university	0.04	8. university	0.05
9. computer	0.02	9. research	0.04	9. quantum computing	0.04
10. distributed systems	0.02	10. computer science department	0.13	10. theory	0.03

Figure 2. Sample of concept hierarchy and the top-10 keywords in the content. The bold phrases also appear in the top-10 keywords of the parent concept.

In the keyword suggestion stage, the advertiser will first input a keyword as a query. The keyword will then be matched to the concept hierarchy to search for relevant concepts. The relevant concepts will then be categorized if there is a distinct difference between their meanings. Finally, for each category, keyword suggestion work will be performed to find relevant keywords.

4. CONCEPT HIERARCHY

4.1 Deriving the Concept Hierarchy from A Web Directory

A concept hierarchy is a hierarchically organized concept set. The concepts in the hierarchy are organized into multiple levels such that concepts at higher levels have more general meanings than those at lower levels.

In order to construct such a concept hierarchy, two problems should be solved: how to assign the relationships among the concepts, and how to define the meanings of the concepts?

Assume that the hierarchy is constructed by “is-a” relationships only, which indicates the parent concept subsumes its children. Our approach is to build the hierarchy structure based on the web directory. In our work, ODP is employed to construct such a concept hierarchy. ODP runs the world’s largest manually defined web page directory, which contains hundreds of thousands of categories that are hierarchically organized. These categories cluster web pages which contain similar contents. For example, “Pop” and “Rock n Roll” are categories which arrange music of a specified style; and they are both children of the category named

“Music,” which is a kind of “Art.” Thus, the categories in the directory can be considered as the concepts in the hierarchy.

The meaning of each concept is represented by a bag of related phrases. Since the categories in the web directory organize web pages with similar topics, we can utilize these web pages to supply the meaning of the corresponding concept. Some keyword extraction techniques are employed to extract keyword phrases from web pages. As a result, these phrases will be gathered together to describe the meaning of the concept. However, these phrases alone are not sufficient to describe the complete meaning of the concepts.

As mentioned above, the concepts are related to each other and organized hierarchically according to their subsumption relationship. For a certain concept, its meaning is described not only by itself, but also by its sub-concepts. As described in Huang’s work ([12]), if we define c as a concept, whose immediate descendants are $d_i \in D_c$. The direct meaning of c is $M_D(c)$ while the complete meaning of concept c is $M_C(c)$. We say that $M_C(c)$ is the union of direct meaning, $M_D(c)$ and the complete meaning of all its descendants, $M_C(d_i)$:

$$M_C(c) = M_D(c) \cup \{M_C(d_i) | d_i \in D_c\}$$

The construction of the hierarchy can be performed in a bottom-up fashion. For the leaf nodes, only the directly extracted phrases are gathered. Then these contents will be accumulated and contributed to their ancestors. It is obvious that the content of concepts in the higher level will contain more phrases. However,

not all of the phrases accumulated from the sub-concepts are appropriate. Some methods should be applied to rank the phrases and find the most relevant ones, which are detailed in the next subsection.

4.2 Associating Concepts with Phrases

Each phrase in the content can be considered as a feature of the target concept. Intuitively, a phrase is a good feature if it is commonly used within the concept and seldom used by other concepts.

Remember in section 3, we proposed similarity measurement $P(c \rightarrow t) = \frac{f_c(t)}{\text{coverage}(t)}$. Here the support value, $f_c(t)$, indicates whether the phrase frequently occurs in the document collection associated with the concept and $\text{coverage}(t)$ indicates whether the phrase is widely used in the whole concept hierarchy.

The support of phrase t is determined by the co-occurring frequency:

$$f_c(t) = \frac{|D_{c,t}|}{|D_c|}$$

$f_c(t)$ can be regarded as the normalized DF, which will be referred to as DF for convenience.

To estimate the coverage, we propose the following criterion:

ICF: inverted category frequency

$$\frac{1}{\text{coverage}(t)} = ICF(c, t) = \log \left(\frac{N}{cf(t) - cf(c, t) + 1} + 1 \right)$$

Here $cf(t)$ means the number of categories that include phrase t and $cf(c, t)$ means the number of sub-categories of concept c that contain phrase t . N means the total number of categories. Since it is not easy to calculate the accurate value of $cf(t)$ directly, we can use the occurrence in the leaf categories to represent the approximate $cf(t)$ value.

The ICF factor enforces the terms that are peculiar to a certain category and reduces the weight of common ones. As a result, the weighting scheme can be written as DF-ICF, which is a variation of tf-idf.

Assigning weights to the phrases is also a recursive procedure which can be performed along with the construction of the hierarchy. The process starts from the leaf categories. Phrases in the leaf categories will be ranked simply by their support value (for leaf categories there is no more sub-categories, so the coverage is unavailable here). And for their super-categories, all the phrases from the subs are accumulated and ranked by $P(c \rightarrow t)$. This process continues until the root concept has been handled. Figure 2 is a sample of a concept contents.

5. KEYWORD SUGGESTION

After the concept hierarchy is built, keyword suggestion work can be performed on it. The major intuition of keyword suggestion is to find the relevant concepts for the given keyword and look for phrases related to these concepts. Because the given keyword may have multiple meanings, keyword suggestion cannot make out which one meets the user's demand. However, since it is still possible for us to figure out all the distinguishing meanings of a certain phrase, keyword suggestion work can be performed

according to each of the meanings and then deliver the choice to the user.

The procedure of keyword suggestion consists of three steps. 1. Matching. Given a certain phrase, find out the most relevant concepts. 2. Categorization. Categorize the related concepts with distinct meanings to avoid the interference of the suggestion result from each other. 3. Suggestion. Suggest new phrases with only the categorized concepts.

Figure 3 illustrates the idea of keyword suggestion. The box on the left denotes the seed query phrase, from which new keywords should be suggested. The tree in the middle denotes the concept hierarchy. The boxes on the right denote the candidate phrases to be suggested. By mapping the given keyword to the hierarchy we can get a list of relevant concepts. The width of the dotted line between the box (keyword) and the circle (concept) reflects the degree of similarity. The related concepts are then categorized into 2 clusters: the grey circles denote the first cluster and the shaded circles denote the second one. With the relevant concepts, we can find relevant phrases. Since the concepts in the different clusters concentrate on different topics, the expanded phrases will be significantly different.

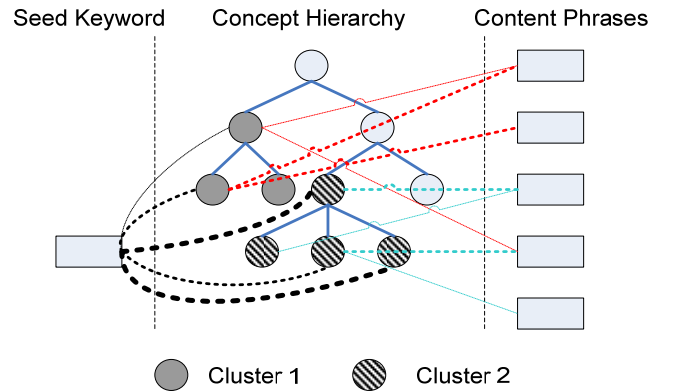


Figure 3. Illustration of keyword suggestion based on concept hierarchy

5.1 Matching

We use a probabilistic model to explain the weights of phrases in a concept, $P(q \rightarrow c) = f_c(q)w_c(q)$. For a certain concept, the weight of relevant phrase, $f_c(q)$, indicates the conditional probability of query q given the concept c , which can be defined as $P(q|c)$.

Due to the variation of the categories, for a certain phrase q , $P(q|c)$ alone cannot tell which category is more important for it. Instead, $P(c|q)$, which indicates the conditional probability of concept c given query q , should be used. To calculate $P(c|q)$ we can use the Bayesian theorem:

$$P(c|q) = \frac{P(q|c)P(c)}{P(q)}$$

Assume that $P(c)$ reflects the importance of the concept, and the importance of a concept can be estimated by the number of documents in its category, which means $\frac{|D_c|}{\sum_{c \in \mathcal{C}} |D_c|} = \frac{1}{N} |doc(c)|$.

For a certain phrase q , $P(q)$ can be considered as a constant, so the formula can be rewritten as below:

$$P(c|q) = \frac{P(q|c)P(c)}{P(q)}$$

$$= P(q|c) \frac{|D_c|}{constant}$$

Note that here $\frac{|D_c|}{constant}$ reflects the size of concept c , which is one of the factors about the concept importance. To measure the other factor, the distribution of the phrases, we propose new weighting functions:

SCF: sub-category frequency

$$SCF(c, t) = \log(cf(c, t) + 1)$$

Here $cf(c, t)$ denotes the number of sub-categories of concept c that contain phrase t as described above, $cf(c)$ denotes the total number of sub-categories of concept c .

To simplify the calculation, a more straight forward method is proposed.

LCF: local category frequency

$$LCF(t) = \log(lcf(c, t) + 1)$$

Here $lcf(c, t)$ denotes the number of immediate sub-categories of concept c with which phrase t occurs, and $lcf(c)$ denotes the total number of immediate sub-categories of concept c . LCF can be considered as a variation of SCF.

5.2 Categorization

After the matching stage, the relevant concepts have been obtained. Since it cannot guarantee that all the relevant concepts share a similar meaning, we should determine which concepts hold different meanings to avoid them interfering with each other in the suggestion stage.

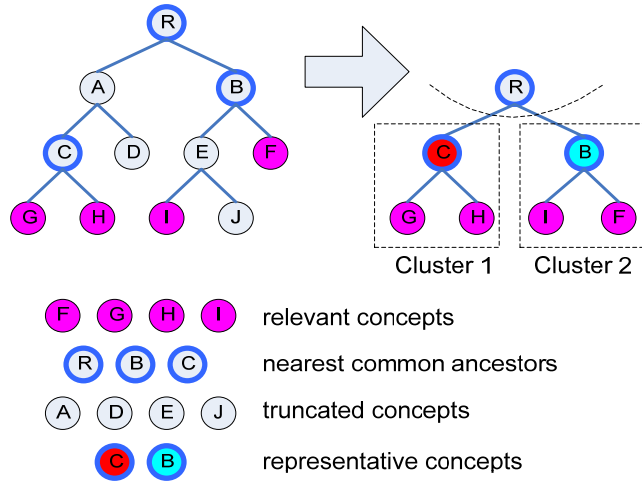


Figure 4. Illustration of hierarchy truncating and concept categorizing

A procedure is carried out to categorize the relevant concepts into *clusters*. To make the users understand the meaning of the clusters more easily, for each of the clusters, a *representative concept* will be used to describe this cluster of concepts.

Since the concepts are hierarchically organized, we can make an assumption that for any concept the intra-similarity is greater than the inter-similarity. This assumption indicates that the clustering

of the concepts in the hierarchy will result in the optimized solution. We can perform the clustering work in an agglomerative way: the fusion of the clusters starts from the lowest concepts in the hierarchy and the similar clusters are then merged and represented by their *LCS (least common subsume)*. Such work continues until the remaining clusters are sufficiently distinct. The clustering work can be considered as the truncating of the original concept hierarchy with only relevant concepts and their closest common ancestors left. A sample clustering process is illustrated in Figure 4. The original hierarchy will be truncated into a smaller tree. After the truncation, all the relevant concepts are hierarchically clustered and represented by their LCS. For example, concepts I and F are categorized in one cluster and represented by concept B. Figure 5 presents the detailed truncating algorithm.

A simple but effective approach to categorizing the concepts is just to categorize the clusters according to the uppermost level concepts (i.e. the immediate sub-concepts of the root). This approach guarantees that the ultimate number of clusters will not exceed that of the immediate children of the root concept. In Figure 4, for example, after cutting the truncated hierarchy from the uppermost level, the concepts will be categorized into 2 clusters (represented by concept C and concept B in the figure respectively).

Algorithm Hierarchy_Truncating

Input: The related concept set S

The sub-hierarchy whose root is concept c

Output: root of truncated sub-hierarchy in c

1. candidate_list = {}
2. **for all** sub-concept c_i of c **do**
3. $c_i' \leftarrow \text{Hierarchy_Truncating}(S, c_i)$
4. **if** c_i' not null **then**
5. candidate_list \leftarrow candidate_list \cup c_i'
6. **if** $c \in S$ or $|\text{candidate_list}| > 1$ **then** // c is the related concept, or the LCS
9. **for all** $c_i' \in \text{candidate_list}$ **do**
10. $c.\text{children} \leftarrow c.\text{children} \cup c_i'$
12. **return** c
14. **if** $|\text{candidate_list}| == 1$ **then** // c is not the nearest common ancestor
15. **return** the only concept in candidate_list
16. **else** // c is totally irrelevant
17. **return** null
18. **end if**

Figure 5. Pseudo code of hierarchy truncating procedure

5.3 Suggestion

As a result, keyword suggestion is performed via the concepts. Given a query q , all the phrases in the related concepts will be examined according to the similarity function defined in section 3,

$$\begin{aligned} Sim(q, t) &= \sum_c P(q \rightarrow c)P(c \rightarrow t) \\ &= \sum_{c \in C_q} P(q \rightarrow c)P(c \rightarrow t) \end{aligned}$$

The phrases with the highest rank will be provided in the end.

6. EVALUATION

We propose a series of experiments to examine the effect of our algorithm for keyword suggestion. The experiments are designed to answer the following questions:

1. How do the weight functions of phrases affect the meaning of concepts? Different weight functions measure the fitness of phrases in different way, resulting in different weight values and different ranked order correspondingly. Experiments are conducted to determine how much of the meaning changes when cooperating with different weight functions.
2. How do the weight functions of phrases affect the accuracy of concept matching? Our system attempts to determine the meaning of a phrase by matching it to the concepts. Experiments will be made to examine the accuracy of matching when taking different weight functions into account.
3. How does keyword suggestion work compared with the other suggested approaches? We compare it with some traditional suggestion approaches.
4. Could the categorization of concepts reduce the interference from distinct meanings in suggestion? We performed a case study to see that categorizing the concepts can discover distinguishing meanings and provide tight relevant suggestions.

6.1 Dataset and Implementation

The taxonomy used in this research to build the concept hierarchy is ODP. About 1,306,586 web pages are crawled from the 150,446 ODP categories. All the documents and sub-categories within “top/regional” are excluded because they are duplicated. The documents and sub-categories in “top/world” are also excluded for they are composed by languages other than English. In the preprocessing stage, a keyword extraction tool [11] is employed to convert the web pages into bags of phrases. To achieve better suggestion performance, some common phrases are filtered. To reduce the calculation burden in generation of the concept content, only the phrases that rank within the top 200 for each concept are eliminated. This process also affects the results in the matching stage because for those ranked among the top 200, the root node will never be the most relevant concept when utilizing DF or DFICF weighting functions. To reduce the cluster found in the categorization stage, all the concepts whose similarity to the given keyword is less than 1% of the similarity of the most relevant concepts are eliminated.

To measure the quality of the concept hierarchy, we will evaluate the accuracy of both concept-term matching and term-concept matching. For the first part, five level-2 concepts and up to three immediate sub-concepts of each of them are randomly taken. Three labelers are asked to label the representative phrases. There are four degrees of relevance: from three (all three of the labelers think the concept is highly related) to zero (no labeler thinks it is relevant). The test data of the second part consists of 100 phrases which are randomly sampled from those whose document

frequency ranges from 100 to 1000. Labelers are asked to label the most related concepts for each of the phrases. Similarly, there are also four degrees of relevance.

To evaluate the performance of keyword suggestion, 30 queries (including 10 ambiguous queries) are employed. Some queries are extracted randomly from the phrase list while others are commonly used ambiguous words. For each query we gather and mix the top 200 suggestions from each of the methods and the labelers are asked to manually determine the relevant ones.

6.2 Evaluation Metrics

The measures to quantify the precision of our system are defined according to the standard measure approaches. That is, precision, recall, F1-measure and NDCG [13]. Precision is the proportion of relevant items among all the items returned by the system. Recall is the proportion of relevant items among all the actual relevant items returned by the system. F1 is the harmonic mean of precision and recall:

$$F1 = \frac{2 \times P \times R}{P + R}$$

NDCG score is calculated for the sorted list of results for each of the phrases as follows:

$$NDCG = N \sum_{j=1}^k \frac{2^{score(j)} - 1}{\log_2(j + 1)}$$

where N is the normalization constant chosen so that a perfect ordering of the results will receive the score of one. $score(j)$ is the gain value associated with the label of the item at the j th position of the ranked list. In the $NDCG$ formula, the sum computes the cumulative information gain to the user from the already inspected documents. $\log_2(j + 1)$ is a discounting function that reduces the document’s gain value as its rank increases.

6.3 Evaluation of Concept Hierarchy

In the first experiment we intend to measure the quality of the concept contents. As shown in Table 1, DF-ICF weight achieves better contents. Labelers report that the web-page-based content consists of several widespread but meaningless phrases. So the major problem in building the content is how to filter these noises. Since DF-ICF weight tends to reduce the weight of common phrases, this might be the reason why it is so important in refining the contents.

Table 1. Evaluation of the concept content (compared to weight = DF) of each method

Feature	Average NDCG	Improvement
DF	0.868	N/A
DF-ICF	0.890	2.2%

In the next part, we pay attention to the accuracy of term-concept matching. We use DF (which means only support is taken into consideration) as the baseline. In the implementation, phrases whose ranks are out of the top 200 relevant phrases of the concept c are eliminated. As shown in Table 2, the baseline obtained the average NDCG score of 0.468. The DFSCF achieved an approximate 1% increase in the average score while DFICF made a significant improvement of 10%.

Table 2. Evaluation of the concept matching (compared to weight = DF) of each method

Feature	Average NDCG	Improvement
DF	0.468	N/A
DF-SCF	0.480	1.2%
DF-LCF	0.568	10%

6.4 Evaluation of Keyword Suggestion

6.4.1 Categorizing

Categorizing process aims at providing representative and distinguishing clusters of the relevant concepts. We conduct an experiment to evaluate the categorizing performance. Some ambiguous terms are used as queries to acquire categorized concept clusters.

Given a query, each method will categorize the relevant concepts into *clusters* and generate a *representative concept* to represent each of the clusters. Labelers will then manually identify the *distinguishing clusters* (i.e. none of these clusters are similar) and the *best representative concept* for each of the clusters. Note that if the representative concept of one cluster is identified by the labeler, the cluster itself is certainly distinguishing. On the contrary, if the cluster is distinguishing, it is still possible that its representative concept is not good enough. The performance of each method is measured using several aspects:

- *Found* number. When given a query, the number of clusters found by each method.
- *Hit* number. The number of *best representative concepts* found by each method. Greater number implies that this method is more accurate.
- *Relevant* number. The number of *distinguishing clusters* found whose representative concept is **not** the *best* (usually a bit more generalized or specialized).
- *Redundant* number. When there is more than one cluster with the similar meaning found, the redundant number will be counted. Greater number implies that this method will produce more useless clusters.
The clusters which are actually irrelevant to the query are also counted as “redundant.”
- *Missing* number. Chances are that a method failed to generate clusters that “hit” or are “relevant” to some of the identified *distinguishing clusters*. The number of failures will be shown in “missing.” Greater number implies that this method is less accurate.

Table 3. Sample of the categorizing evaluation, query = “apple”

Clusters	Representative Concept	Hit by
The computer company	Top/Computers	DF, DFSCF, DF LCF
The Fruit	Top/Business/Food and Related Products/Produce/Fruits/Apples	DF LCF
The wine “Cider”	Top/Recreation/Food/Drink/Cider	DF, DFSCF, DF LCF

Table 3 illustrates the evaluation with query “apple.” Three clusters are provided and DF LCF “hit” all the 3 clusters. Table 4 shows the average performance of each method. DFSCF and DF LCF show better accuracy while they still tend to produce more redundant clusters.

Table 4. Average categorizing accuracy of each method.

Method	Found	Hit	Relevant	Redundant	Missing
DF	6.4	1.8	0.7	3.9	0.4
DFSCF	7.8	2.4	0.3	5.1	0.2
DF LCF	7.1	2.4	0.3	4.5	0.2

6.4.2 Direct Suggestion

Direct suggestion means keyword suggestion without categorization. In this test queries that have relatively simple meanings are used. We evaluate the suggestion effect for several approaches. The baseline is a naive approach which ranks the relevance by co-occurrence of the phrases. DF, DFSCF and DF LCF are evaluated here.

Table 5. Average performance of plain suggestion

Method	Precision	Recall	F1-measure
Baseline	0.74	0.38	0.51
DF	0.77	0.55	0.62
DFSCF	0.81	0.52	0.65
DF LCF	0.82	0.55	0.66

Table 5 summarizes the average results for the four methods. DF LCF outperforms the other four methods while the performance of DFSCF methods is also acceptable. The advantage of mutual factor is that the keyword suggestion process can maintain a relatively high performance while it tries to maximize the categorization quality.

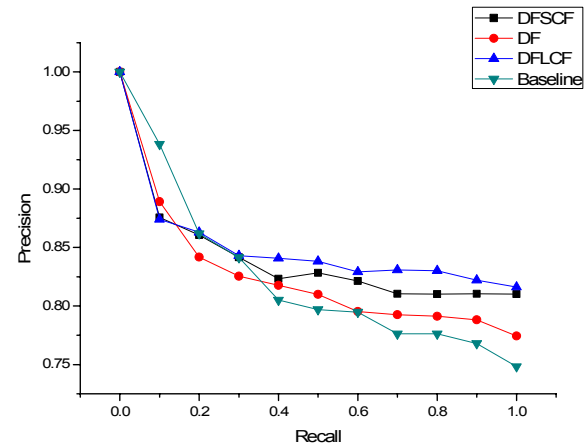


Figure 6. 11-point recall-precision curves

Note that in about 1/3 of the cases, the co-occurrence method is unable to provide enough co-occurring phrases, which is why the baseline method has a relatively low recall value. Figure 6 shows the 11-point recall-precision curves. From the figure we can see that the co-occurrence method could provide highly relevant suggestion at the very beginning. With the growth of the requested number of new phrases, the performance falls quickly.

Table 6. Results for the sample query “matrix”

Categorized		Google	Overture	WordTracker
Top/Arts	Top/Science/Math			
matrix revolutions	matlab	matrix screensaver	the matrix	matrix
matrix reloaded	calculator	matrix reloaded	belief bridging divine matrix miracle space time	the matrix
matrix review	toolbox	matrix revolutions	toyota matrix	matrix reloaded
keanu reeves	mathematica	matrix multiplication	matrix reloaded	toyota matrix
film	functions	matrix inverse	matrix screensaver	matrix soundtrack
neo	software	matrix soundtrack	matrix revolution	matrix revolutions
matrix trilogy	linear algebra	matrix wallpaper	matrix game	matrix theme
movie	scientific calculator	rotation matrix	matrix soundtrack	matrix wallpaper
matrix revolutions review	algebra	matrix code	dot matrix printer	matrix mp3
revolutions	matlab toolbox	matrix revolution	matrix online	matrix screensaver
larry wachowski	biochemistry	matrix inversion	enter the matrix	matrix background
reloaded	graphing	determinant matrix	matrix movie	the matrix reloaded
matrix movies	analysis	trinity matrix	matrix 3d	matrix ping pong
agent smith	calculators	math matrix	enter guide matrix official strategy	matrix code
laurence fishburne	data analysis	symmetric matrix	matrix mris	matrix movie
andy wachowski	computer algebra system	matrix properties	matrix hair product	
morpheus	department	rank matrix	matrix wallpaper	
neo and trinity	linear	algebra matrix	matrix trilogy	
matrix movie	archaeology	matrix product	matrix neo path	
matrix revisited	molecular biology	matrix hacking	matrix shampoo	

6.5 Case Studies

To better understand the effect of the categorized suggestion procedure, we conduct a case study to reveal the performance of the system. “Matrix” is used as the seed keyword here. Our system will find relevant concepts according to the DF-LCF weight. The categorized suggestion will be compared to three existing keyword suggestion tools. Table 6 shows the top 20 results for this sample query.

For the input “matrix,” the DF-LCF factor will generate four categories with two out of them successfully “hitting” the human labeled cluster, “Top/Arts” and “Top/Science/Math”. The other two, “Top/Games/Video_Games” and “Top/ Kids_and_Teens/Entertainment/ Animation/ Anime/ Animatrix_The”, are not displayed because they are considered as redundant results. As we can see, the categorized suggestion can provide new phrases tightly relevant to the distinguishing meaning. A further analysis of the top 50 results shows that categorizing the concepts effectively avoids interference in the suggestion. The suggestion from Google’s Adword is a mixture of both kinds of meanings, resulting in a heavily interfered output. The suggestion from Overture and WordTracker, on the other hand, is about the movie on the whole, resulting in less interference but leaving other important meanings out of sight. Generally speaking, it is not easy, if possible, for traditional suggestion methods to disambiguate the meaning of a keyword. So it is quite natural that they meet obstacles in dealing with this kind of keywords.

Another advantage of our method is that our approach can provide plenty of phrases that do not contain the input keyword like “matrix.” As we have mentioned in section 2, traditional methods cannot provide keywords that do not contain the seed if they make use of query log mining or proximity search. Though some solutions which take synonymy and advertisers’ logs into account have been proposed to address such issue, it is still a challenging task for them to provide a large number of phrases.

7. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel approach for advertising keyword suggestion. We investigated a method to derive a concept hierarchy from a web directory. Three weighting criteria were proposed to tightly associate concepts with phrases. By utilizing the domain knowledge and relationships contained in the concept hierarchy, our approach provides new phrases which are categorized into distinguishing sorts. Experiments show that our system presents a wider and more accurate vision than the traditional techniques. The idea of categorized suggestion is an important contribution focusing on user experience.

In the future we plan to take further advantage of the features of concept hierarchies in order to provide more control for advertisers than the existing methods. Advertisers can easily change the scope of suggestion – either generalize or specialize – along the concept hierarchy, resulting in a new suggestion from a wider or narrower vision.

Application of the concept hierarchy is limited by its lack of flexibility. Concept hierarchy is a relatively static structure so it is not easy for the system to catch up with the changing world. If a new keyword is never seen in the concept content, it cannot be mapped to the concept hierarchy. To solve this problem in the future, a mechanism of automatic content evolution will be proposed to keep the concept hierarchy up-to-date.

8. ACKNOWLEDGMENTS

Thanks to Yiyang Liu and Cody Dunne for proofreading the paper.

9. REFERENCES

- [1] Abhishek, V. Keyword Generation for Search Engine Advertising using Semantic Similarity between Terms. Workshop on Sponsored Search Auctions, WWW (2007).

- [2] Baeza-Yates, R. A. and Ribeiro-Neto, B. Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., 1999.
- [3] Bartz, K., Murthi, V. and Sebastian, S. Logistic Regression and Collaborative Filtering for Sponsored Search Term Recommendation. Proceedings of the Second Workshop on Sponsored Search Auctions (2006).
- [4] Brown, P. F., Della Pietra, V. J., deSouza, P. V. and Lai, J. C. Class-based ngram models of natural language. Computational Linguistics, 18(1992).
- [5] Buckley, C., Salton, G., Allan, J. and Singhal, A. Automatic Query Expansion Using SMART: TREC 3. Overview of the Third Text REtrieval Conference (TREC-3) (1994).
- [6] Carrasco, J., Fain, D., Lang, K. and Zhukov, L. Clustering of bipartite advertiser-keyword graph. International Conference on Data Mining (2003).
- [7] Cimiano, P., Pivk, A., Schmidt-Thieme, L. and Staab, S. Learning taxonomic relations from heterogeneous sources. Proceedings of the ECAI 2004 Ontology Learning and Population Workshop (2004).
- [8] Doan-Nguyen, H. and Kosseim, L. Using Terminology and a Concept Hierarchy for Restricted-Domain Question-Answering. Research on Computing Science, Special issue on Advances in Natural Language Processing, 18(2006)
- [9] Ganesan, P., Garcia-Molina, H. and Widom, J. Exploiting hierarchical domain structure to compute similarity. ACM Trans. Inf. Syst. 2003.
- [10] Gelbukh, A. F., Sidorov, G. and Guzman-Arenas, A. Document Indexing With a Concept Hierarchy. New Developments in Digital Libraries, Proceedings of the 1st International Workshop on New Developments in Digital Libraries (2001).
- [11] Google AdWords Keyword Tool, <https://adwords.google.com/select/KeywordToolExternal>
- [12] Huang, C.-C., Chuang, S.-L. and Chien, L.-F. Liveclassifier: creating hierarchical text classifiers through web corpora. Proceedings of the 13th international conference on World Wide Web (2004).
- [13] JÄarvelin, K. and KekÄalÄainen, J. IR evaluation methods for retrieving highly relevant documents. Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (2000).
- [14] Jones, R., Rey, B., Madani, O. and Greiner, W. Generating query substitutions. Proceedings of the 15th international conference on World Wide Web (2006).
- [15] Joshi, A. and Motwani, R. Keyword Generation for Search Engine Advertising. Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops (2006).
- [16] Koller, D. and Sahami, M. Hierarchically Classifying Documents Using Very Few Words. Proceedings of the Fourteenth International Conference on Machine Learning (1997).
- [17] Nanas, N., Uren, V. and Roeck, A. D. Building and applying a concept hierarchy representation of a user profile. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (2003).
- [18] Open Directory Project, <http://www.dmoz.org>
- [19] Overture Keyword Selection Tool, <http://inventory.overture.com/d/searchinventory/suggestion/>
- [20] Qiu, Y. and Frei, H.-P. Concept based query expansion. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (1993).
- [21] Ribeiro-Neto, B., Cristo, M., Golgher, P. B. and Moura, E. S. d. Impedance coupling in content-targeted advertising. Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (2005).
- [22] Sanderson, M. and Croft, B. Deriving concept hierarchies from text. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (1999).
- [23] Santamaria, C., Gonzalo, J., Verdejo, F. Automatic Association of Web Directories to Word Senses. Computational Linguistics (2003)
- [24] The Google adwords. Google content-targeted advertising. http://adwords.google.com/select/ct_faq.html, November 2004.
- [25] Wang, BB., McKay, RI, Abbass, HA, Barlow, M., Learning text classifier using the domain concept hierarchy, IEEE International Conference on Communications (2002).
- [26] Wang, W., Meng, W. and Yu, C. Concept Hierarchy Based Text Database Categorization in a Metasearch Engine Environment. Proceedings of the First International Conference on Web Information Systems Engineering (2000).
- [27] Wang, K., Zhou, S. and Liew, S. C. Building Hierarchical Classifiers Using Class Proximity. Proceedings of the 25th International Conference on Very Large Data Bases (1999).
- [28] Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C. and Nevill-Manning, C. G. KEA: practical automatic keyphrase extraction. Proceedings of the fourth ACM conference on Digital libraries (1999).
- [29] WordTracker, <http://www.wordtracker.com/>
- [30] Xu, J. and Croft, W. B. Improving the effectiveness of information retrieval with local context analysis. ACM Press, City, 2000.
- [31] Yih, W.-t., Goodman, J. and Carvalho, V. R. Finding advertising keywords on web pages. Proceedings of the 15th international conference on World Wide Web (2006).