Knowl Inf Syst (2015) 43:599–631
DOI 10.1007/s10115-014-0745-z

REGULAR PAPER

Improving contextual advertising matching by using Wikipedia thesaurus knowledge

Guandong Xu · Zongda Wu · Guiling Li · Enhong Chen

Received: 18 February 2013 / Revised: 14 January 2014 / Accepted: 22 March 2014 /
Published online: 4 April 2014

Abstract As a prevalent type of Web advertising, contextual advertising refers to the placement of the most relevant commercial ads within the content of a Web page, to provide a better user experience and as a result increase the user's ad-click rate. However, due to the intrinsic problems of homonymy and polysemy, the low intersection of keywords, and a lack of sufficient semantics, traditional keyword matching techniques are not able to effectively handle contextual matching and retrieve relevant ads for the user, resulting in an unsatisfactory performance in ad selection. In this paper, we introduce a new contextual advertising approach to overcome these problems, which uses Wikipedia thesaurus knowledge to enrich the semantic expression of a target page (or an ad). First, we map each page into a keyword vector, upon which two additional feature vectors, the Wikipedia concept and category vector derived from the Wikipedia thesaurus structure, are then constructed. Second, to determine the relevant ads for a given page, we propose a linear similarity fusion mechanism, which combines the above three feature vectors in a unified manner. Last, we validate our approach using a set of real ads, real pages along with the external Wikipedia thesaurus. The experimental results show that our approach outperforms the conventional contextual advertising matching approaches and can substantially improve the performance of ad selection.

Keywords Wikipedia · Contextual advertising · Similarity measure

G. Xu
Faculty of Engineering and IT, University of Technology, Sydney, Australia

Z. Wu (✉)
Oujiang College, Wenzhou University, Wenzhou, Zhejiang, China
e-mail: zongda1983@163.com

G. Li
School of Computer Science and Technology, China University of Geosciences, Wuhan, China

E. Chen
School of Computer Science, University of Science and Technology of China, Hefei, China

1 Introduction

With the overwhelming prevalence of Internet technologies in our daily life, computational advertising that is a newly emerging interdisciplinary section of textual search, statistical learning, optimization and marketing, has become one of the most important media channels for advertising. PwC[1] predicts that computational advertising will become the second largest advertising medium in America after TV within the next 4 years, and spending in this area will increase from 24.2 billion dollars in 2009 to 34.4 billion dollars in 2014. In computational advertising, the core challenge is to match a given user's information need with a relevant advertisement in a certain context.[2] Based on the various contextual scenarios and targeted contents, thus, computational advertising could be categorized into different advertising channels. For example, it could be divided into graphical ads and textual ads according to the characteristics of the targeted content, where the former delivers the visualized impressions while the latter focuses on textual processing. In contrast, the context of a user searching a query in a search engine leads to a *sponsored search*, whereas that of browsing a Web page is more suitable for *contextual advertising* and *displayed ads*. Since the majority of Web content is in the form of textual information, textual Web advertising has been well studied in this area.

Generally speaking, there are two main types of textual advertising, i.e., sponsored search and contextual advertising [1,3]: (1) sponsored search, which selects ads based on keywords in search queries given by users, is characterized by placing paid textual ads links on the result pages returned by a search engine (e.g., Google) and (2) contextual advertising, which judges the content relevance of ads to the page that the user is browsing, refers to the selection of relevant ads for the targeted page. Sponsored search is mainly restricted to the use of a search engine, while contextual advertising could be used in a broad spectrum of Internet services including generic sites ranging from individual bloggers and small niche communities to large publishers. Now, almost all for-profit non-transactional sites, i.e., the sites that do not sell anything directly, rely heavily on the revenues from contextual advertising. Without contextual advertising, the Web will lose the most of its market value.

In contextual advertising, the most important task is to select the best matched ads to the target page because a number of studies have confirmed that the relevance of ads to the page where the ads are placed has a direct impact on the amount of users' ad-clicks [4,32]. For example, given a page about "travel in China," embedding the page with an ad about "hotel information" would attract more user attention than a randomly chosen ad. Most conventional approaches to contextual advertising deal with ad-relevance based on the bag-of-words model (aka, the keyword matching). However, as pointed out in [1,3,23], keyword matching can lead to the following problems that degrade its effectiveness:

1. Homonymy and polysemy cause the lexicon ambiguity and, as a result, may lead to the selection of irrelevant ads for a Web page. For example, a page about "puma, an American feline resembling a lion" might trigger an ad about "puma sport," a sport brand, obviously, which is semantically irrelevant to the page.
2. Low intersection of keywords between pages and ads, which is caused by the limited length of ad content, makes it difficult to match ads that are semantically relevant to the page concepts, which are represented in different terms (i.e., synonyms). For example, "United States" has a low intersection with "USA" or "Yankee Land," but is obviously quite relevant.

[1] PricewaterhouseCoopers—www.pwc.com.

[2] Introduction to Computational Advertising—www.stanford.edu/class/msande239/.

3. Context mismatch occurs because of a lack of sufficient semantics, even when keyword matching is met [3]. For example, an ad about "travel in China" can be incorrectly selected to the page that is about "earthquake in China," because here, it is only concerned with the common word "China" contained in both, but neglects the difference in the topics.

Traditional keyword matching approaches only take the term-occurrence between pages and ads into consideration and do not consider semantic relevance. That is, using traditional keyword matching alone is difficult to select the most relevant ads for a target page accurately.

Inspired by the research findings in related topics, a feasible solution is to conduct semantic analysis on the page and content to reveal the underlying semantic relevance between them. More theoretically, the above problems are stemmed as the semantic analysis of textual documents [7,8,15], which is an interesting and popular topic in information retrieval and natural language processing. The key idea of semantic analysis in this context is to utilize an external thesaurus or corpus as a knowledge base, upon which explicit semantic analysis will be conducted. So far a number of external knowledge bases have been developed and utilized to enhance the inherent semantic expression for texts, such as WordNet, *Open Directory Project* (ODP), and Wikipedia.

Likewise, this kind of technique was introduced into contextual advertising in [22,23]. In their work, around 1,000 *feature* articles are first selected from Wikipedia on various topics as a reference space upon which all pages and ads are remapped as vectors. Each element in these vectors is determined by the traditional cosine similarity between all the feature articles and each page (and each ad) over the *tf-idf* formula. Then, the relevant ads for a given page are recommended by the similarity calculated using all the feature articles as dimensions in the intermediate reference model. The experimental results have confirmed the effectiveness of this approach to overcome the problems encountered in keyword matching. However, this approach still possesses the following problems, thereby degrading its capability in practical contextual advertising:

1. Limited coverage over semantic concepts. In order to improve the running performance, it only chooses a small part of articles from Wikipedia as reference articles (dimensions). As a result, for many pages that are not properly characterized by the reference articles, the remapped vectors are very sparse (i.e., most elements have almost no weights), consequently leading to the return of many irrelevant ads for the pages.
2. Time-consuming overhead. To overcome the limited coverage over the semantic concepts, an intuitive way is to choose a sufficient number of reference articles from Wikipedia. However, this will result in a seriously decreased performance, since the time spent on full-text matching between all the reference articles and a page (or an ad) is high.

In practical contextual advertising, selection effectiveness and efficiency are two main concerns. On one hand, the selected ads should be as relevant to the targeted page as possible, so as to attract more user interest and increase the user's ad-click rate; but on the other hand, the selection procedure should be completed within a reasonable response time in order to avoid the user becoming impatient and thus increase the user's Web browsing satisfaction. However, as mentioned above, the existing Wikipedia matching approaches do not solve the contradiction between effectiveness and efficiency and thus are not suitable to be applied into practical applications. In this paper, we aim to tackle this problem by proposing an improved contextual matching approach by using a Wikipedia thesaurus ontology.

Wikipedia, as one of the largest human open knowledge repositories in the world, contains broad coverage over many diverse semantic concepts [20] and thus has been widely used to represent semantic attributes in the areas of artificial intelligence and information retrieval [13]. Hence, the basic idea behind this paper is to incorporate this kind of external knowledge

to reinforce the semantic expression of texts along with the traditional keyword matching strategy, so as to improve contextual matching effectiveness. In our approach, we particularly take two aspects of similarity measures between pages and ads into consideration: (1) the keyword-based similarity capturing textual commonness, and (2) the Wikipedia-based similarity measuring the relevance from the semantic perspectives of thesaurus ontology.

Similar to existing techniques, our approach consists of the following several steps. First of all, we choose a sufficient number of articles and categories from Wikipedia, to cover as many semantic concepts as possible. Second, we build up the keyword expression for each targeted page (as well as each ad) and then derive the additional two feature vectors from the Wikipedia thesaurus ontology, i.e., the concept vector and category vector. Last, we propose a linear similarity fusion mechanism, which combines the above three types of similarity measures in a unified manner, to make the top-N ads selection. In particular, the whole process is divided into an offline stage and an online stage. All processing on articles and structures is carried out in the offline stage, while only the processing on the targeted page and ad selection is online. This means that apart from the process of generating feature vector representation for each page, no time-consuming full-text matching between the page and all the Wikipedia reference articles is needed. This strategy will better deal with the trade-off between the effectiveness and efficiency of contextual matching, which highlights the distinctive advantage of our approach.

In order to evaluate the effectiveness of our approach, we have conducted experiments over a dataset containing real ads and pages. The experimental results show that our approach, which combines Wikipedia-based semantic matching with keyword matching, can substantially improve the accuracy of the similarity measure between pages and ads, consequently improving contextual advertising effectiveness. Besides, the experimental results also show that, due to the elimination of time-consuming full-text matching between all the reference articles and pages, our approach also has a good running efficiency.

The rest of this paper is organized as follows. Section 2 provides the background on the contextual advertising platform and Wikipedia link structure. Section 3 presents the problem statement and reviews some existing approaches to contextual advertising, used as a comparison with ours. Section 4 details our methodology of integrating the Wikipedia concept and category information into keyword matching. Section 5 presents the experimental evaluation results. Last, Sect. 6 surveys related work and Sect. 7 concludes our work.

2 Background

2.1 Contextual advertising

The first major contextual advertising platform was provided by Google in 2003. Now, almost all popular search engines such as Baidu, Yahoo! and Microsoft Bing provide similar platforms. A contextual advertising platform generally comprises four parts [1,9], shown as Fig. 1.

1. The **advertiser** provides the supply of ads, which is usually a company that wants to use the platform to promote their products and needs to pay for its ads.
2. The **publisher** is the owner of a Web site on which ads are placed, who typically aims to provide a good user experience and increase the number of ad-clicks, so as to maximize the market revenue.

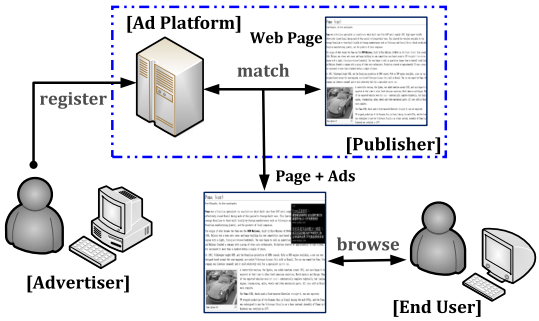


Fig. 1 Key players in contextual advertising platform

3. The **ad platform** is a software system of matching ads to pages, which selects appropriate ads based on content similarity between pages and ads to maximize ad revenue for the publisher.
4. End **users** consist of customer groups who have potential interest in the ads while browsing the content of a Web page, supplied by the publisher.

The most dominant online advertising pricing model for ads is pay-per-click (PPC), where the advertisers pay a certain amount to the publisher and the ad platform for each user's click on the ads. Besides, there are also other types of pricing models for textual ads, including (1) pay-per-impression, where the advertisers pay for the number of ads displayed on a Web page; and (2) pay-per-action, where the payment made by the advertiser is calculated by each sale originating from the ads. Since most existing contextual advertising approaches are mainly based on the PPC model, in this paper, we also focus on this model to address the advertising issue.

A number of user studies have confirmed that the users' ad-click rate can be boosted by increasing the ad's relevance to the targeted page [1,4]. Therefore, we can simply assume that the probability of a user clicking on any ad of a page is determined by the relevance of the ad with respect to the page. Moreover, for simplicity, in this paper, we also ignore the positional effect of ad placement, as used in [1,3,9,17]. Based on the above consideration, we can conclude that under the PPC model, for a given page, selecting more relevant ads to the content of the page is more desirable, because it boosts the revenue received from the advertisers by increasing the probability of a user clicking on the ads.

2.2 Wikipedia thesaurus

Launched in 2001, Wikipedia[3] is a Web-based, free-content, multilingual encyclopedia, which has been written collaboratively by more than 91,000 regular contributors around the world [20,27]. Wikipedia is a very dynamic and rapidly growing external knowledge resource, where articles about newsworthy events are often added within few days of their occurrence [16]. As pointed out in [23], compared to other knowledge repositories, Wikipedia has the following three distinctive advantages, which motivates us to choose Wikipedia in our work: (1) It has very broad knowledge coverage about different concepts, due to the comprehensive contributions by volunteers around the world; (2) Its articles are updated regularly and frequently, and consequently, its knowledge database is always up to date and in step with the times; and (3) It contains a large number of new terms that cannot be found in other linguistic corpora, due to its Web-based open characteristic.

[3] Wikipedia—www.wikipedia.org.

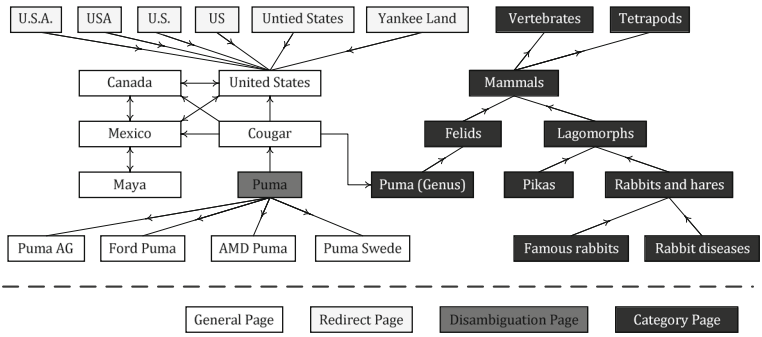


Fig. 2 Example structure from Wikipedia

Each article in Wikipedia describes a single concept, and its title is a succinct, well-formed phrase that resembles terms in a conventional thesaurus. Each article must belong to at least one category. Moreover, there are many hyperlinks between articles, which reflect many semantic relations, such as equivalent relations (synonymy), hierarchical relations (hyponymy) and associative relations. Below, we briefly introduce the linkage structure in Wikipedia (see [20] for more details).

2.2.1 Redirect pages

In Wikipedia, there is only one article for each concept. However, there can be many equivalent titles for a concept due to the existence of synonyms, etc. Wikipedia uses a redirect page, which only contains redirect hyperlinks, to link each equivalent concept to the source article. Redirect hyperlinks can handle capitalization and spelling variations, abbreviations, synonyms, colloquialisms and scientific terms. As shown in Fig. 2, an entry with a considerably higher number of redirect pages is "United States." Its redirect pages correspond to acronyms (U.S., U.S.A., US, USA), misspellings (Untied States) or synonyms (Yankee land).

2.2.2 Disambiguation pages

In Wikipedia, disambiguation pages are created for ambiguous terms, i.e., the terms that denote two or more concepts, e.g., the term "Cell" may refer to many concepts such as the basic life unit, a microprocessor architecture and a scientific journal. Wikipedia provides disambiguation pages that contain various possible meanings, from which users can select articles corresponding to their intended concepts. For example, the disambiguation page for "Puma" lists 22 associated concepts, four of which are given in Fig. 2, from persons (Puma Swede), to vehicles (Ford Puma) and a company (Puma AG).

2.2.3 Article pages

In Wikipedia, each article can link to several entries, thus forming an interconnected network over articles. An editor can insert a hyperlink between a word or phrase and its corresponding Wikipedia entry when editing an article. If we denote each article as a node, and each hyperlink between articles as an edge, pointing from one node to another, then the articles and their hyperlinks form a directed graph (see the left side of Fig. 2).

2.2.4 Category pages

In Wikipedia, each article can belong to more than one category, e.g., the article about the "iPhone" belongs to two categories: "Apple Inc. mobile phones" and "Digital audio players." Moreover, these categories can be further categorized by associating them with one or more parent categories. As shown in Fig. 2, the category about "Mammals" belongs to two parent categories: "Vertebrates" and "Tetrapods." Thus, the category structure does not form a simple tree-structured taxonomy, but a directed acyclic graph (see the right side of Fig. 2), where multiple categorization schemes coexist simultaneously.

All the above linkage and categorization information form a huge thesaurus, in which the semantic relationships are associated and reflected (similar to an ontology graph). In this paper, our major motivation is to utilize this informative and useful graph to improve contextual semantic matching between pages and ads.

3 Problem statement

3.1 Problem definition

Without loss of generality, the task of contextual advertising is defined as the selection of the most relevant ads to a given page. Let $\boldsymbol{p}$ be a targeted page used to match candidate ads. Let $\mathcal{A}$ be the candidate ad database that contains N_a ads, represented by $\mathcal{A} = \{a_j\}_{j=1}^{N_a}$. Let N be the number of expected ads to be embedded into a page, generally, which is given by the publisher. Let $\mathbf{sim}(\boldsymbol{p}, a_j)$ be the similarity metric, which is used to compute the relevance between the page $\boldsymbol{p}$ and the ad a_j. Then, the above expectation about selecting the most relevant N ads for a given page $\boldsymbol{p}$ from the candidate ad database $\mathcal{A}$ can be formulated as follows [where x_j indicates whether the ad a_j is selected ($x_j = 1$) or not ($x_j = 0$)]:

$$\max_{(\boldsymbol{x})} f(\boldsymbol{x}) = \sum_{j=1}^{N_a} x_j \mathbf{sim}(\boldsymbol{p}, a_j) \; \boldsymbol{s.t.} \sum_{j=1}^{N_a} x_j = N, \quad x_j \in \{0, 1\} \tag{1}$$

Based on Eq. (1), we conclude that $\mathbf{sim}(\boldsymbol{p}, a_j)$ is an essential metric, whose accuracy directly determines the accuracy of selected ads to their pages. In other words, the most essential problem in practical contextual advertising is how to accurately and efficiently judge the relevance of an ad to a given Web page. More specifically, a good similarity metric that is able to accurately measure the relevance between a page $\boldsymbol{p}$ and an ad a_j [i.e., $\mathbf{sim}(\boldsymbol{p}, a_j)$] should satisfy the following two requirements: (1) **good accuracy** on relevance judgment between pages and ads, i.e., the more relevant the page $\boldsymbol{p}$ is to the ad a_j, the greater value $\mathbf{sim}(\boldsymbol{p}, a_j)$ should be; and (2) **good efficiency**, i.e., it should be as efficient as possible when computing the value of $\mathbf{sim}(\boldsymbol{p}, a_j)$, to decrease the time spent on contextual advertising.

However, in practical contextual advertising, it is difficult to balance the accuracy and efficiency. In general, different relevance judgments may result in different contextual advertising techniques. In the following subsections, we will briefly introduce the three main approaches of contextual advertising and discuss their advantages and disadvantages.

3.2 Keyword matching

The well-known keyword matching approach estimated ad-relevance based on the co-occurrence of the same keywords between pages and ads. It has been widely applied in

text classification [17,21,25] and has begun to be applied in contextual advertising [22,23]. In general, this approach was implemented by using the *tf-idf* weight [30] together with the cosine metric [11].

Let $\mathcal{P}$ be a set of pages and $\mathcal{K}$ a set of all the keywords contained in $\mathcal{P}$. Let $\boldsymbol{p}$ be a page and k a keyword, i.e., $\boldsymbol{p} \in \mathcal{P}$ and $k \in \mathcal{K}$. We define $\mathbf{tf}(k, \boldsymbol{p}) = (\mathbf{count}(k, \boldsymbol{p}))/(|\mathbf{words}(\boldsymbol{p})|)$, where $\mathbf{count}(k, \boldsymbol{p})$ is the number of occurrences of the keyword k in the page $\boldsymbol{p}$, and $|\mathbf{words}(\boldsymbol{p})|$ is the number of keywords in $\boldsymbol{p}$. Then, the *tf-idf* value of k related to $\boldsymbol{p}$ is calculated as follows:

$$\mathbf{tfidf}(k, \boldsymbol{p}) = \mathbf{tf}(k, \boldsymbol{p}) \log \left(\frac{|\mathcal{P}|}{|\{p : p \in \mathcal{P}, k \in \mathbf{words}(p)\}|} \right) \tag{2}$$

Next, let $\mathbf{words}(\boldsymbol{p})$ be a set of all the keywords contained in the page $\boldsymbol{p}$, and $\mathbf{words}(\boldsymbol{p}) = \left\{ k_1^p, k_2^p, \ldots, k_{N_k^p}^p \right\}$. Based on Eq. (2), a feature vector for the page $\boldsymbol{p}$ (called a **keyword vector**) is constructed, which consists of the *tf-idf* values of all the keywords in $\boldsymbol{p}$:

$$\mathbf{K}(\boldsymbol{p}) = \left(\mathbf{tfidf}\left(k_1^p, \boldsymbol{p}\right), \mathbf{tfidf}\left(k_2^p, \boldsymbol{p}\right), \ldots, \mathbf{tfidf}\left(k_{N_k^p}^p, \boldsymbol{p}\right) \right) \tag{3}$$

Similarly, for a given candidate ad $\boldsymbol{a}$ in the ad database (i.e., $\boldsymbol{a} \in \mathcal{A}$), and $\mathbf{words}(\boldsymbol{a}) = \left\{ k_1^a, k_2^a, \ldots, k_{N_k^a}^a \right\}$, the keyword feature vector for the ad $\boldsymbol{a}$ can be given as follows:

$$\mathbf{K}(\boldsymbol{a}) = \left(\mathbf{tfidf}\left(k_1^a, \boldsymbol{a}\right), \mathbf{tfidf}\left(k_2^a, \boldsymbol{a}\right), \ldots, \mathbf{tfidf}\left(k_{N_k^a}^a, \boldsymbol{a}\right) \right) \tag{4}$$

Last, given the two keyword vectors $\mathbf{K}(\boldsymbol{p})$ and $\mathbf{K}(\boldsymbol{a})$, the textual similarity between $\boldsymbol{p}$ and $\boldsymbol{a}$ is measured by the cosine similarity between vector $\mathbf{K}(\boldsymbol{p})$ and $\mathbf{K}(\boldsymbol{a})$ [11]:

$$\mathbf{sim}^{\mathbf{k}}(\boldsymbol{p}, \boldsymbol{a}) = \frac{\sum_{\forall i \forall j, k_i^p = k_j^a} \mathbf{tfidf}\left(k_i^p, \boldsymbol{p}\right) \mathbf{tfidf}\left(k_j^a, \boldsymbol{a}\right)}{\sqrt{\sum_{j=1}^{N_k^p} \mathbf{tfidf}\left(k_j^p, \boldsymbol{p}\right)^2} \sqrt{\sum_{j=1}^{N_k^a} \mathbf{tfidf}\left(k_j^a, \boldsymbol{a}\right)^2}} \tag{5}$$

The keyword matching approach uses Eq. (5) as the similarity metric to judge the relevance between pages and ads. It can be seen that this approach would have a good running performance, which only needs to do one full-text matching operation to judge the relevance between the page $\boldsymbol{p}$ and the ad a_j, i.e., the time complexity is $\mathbf{O}\left(N_k^p N_k^a\right)$, where N_k^p is the number of keywords in $\boldsymbol{p}$, and N_k^a is the number of keywords in a_j.

However, as pointed out in [15,23], the main drawback of this approach is that it may lead to problems such as homonymy and polysemy, and context mismatch, resulting in mismatching ads to pages. In short, keyword matching may lead to good efficiency but bad accuracy.

3.3 Wikipedia matching

A feasible solution to the problems encountered in keyword matching is to conduct semantic analysis on pages and ads to reveal the underlying semantic relevance between them, typically, such as explicit semantic analysis [7,8]. Later, this approach was introduced into contextual advertising [22,23], and renamed Wikipedia matching.

The basic principle of this approach is to utilize the Wikipedia reference article rather than a keyword as the attribute/dimension in the vector space. More particularly, in this work, a group of feature articles is first selected from Wikipedia to form a reference vector space. Let $\mathcal{W}$ be the set of the selected reference articles (N_w denotes its size), represented by $\mathcal{W} = \{w_j\}_{j=1}^{N_w}$. Next, for the target page $\boldsymbol{p}$, a full-text comparison is made between all

the reference articles and the page by using the keyword similarity metric. The similarity values on various reference articles are then obtained to form a new feature vector (called a **Wikipedia-matching vector**) for the page, which is given as follows:

$$\mathbf{W}(\boldsymbol{p}) = \left(\mathbf{sim}^{\mathbf{k}}(w_1, \boldsymbol{p}), \mathbf{sim}^{\mathbf{k}}(w_2, \boldsymbol{p}), \ldots, \mathbf{sim}^{\mathbf{k}}\left(w_{N_w}, \boldsymbol{p}\right)\right) \tag{6}$$

Similarly, for each candidate ad $\boldsymbol{a}$ from the ad database (i.e., $\boldsymbol{a} \in \mathcal{A}$), a Wikipedia-matching feature vector can also be generated by computing the keyword similarity between all the reference articles and the ad:

$$\mathbf{W}(\boldsymbol{a}) = \left(\mathbf{sim}^{\mathbf{k}}(w_1, \boldsymbol{a}), \mathbf{sim}^{\mathbf{k}}(w_2, \boldsymbol{a}), \ldots, \mathbf{sim}^{\mathbf{k}}\left(w_{N_w}, \boldsymbol{a}\right)\right) \tag{7}$$

Then, the two vectors $\mathbf{W}(\boldsymbol{p})$ and $\mathbf{W}(\boldsymbol{a})$, which represent the remapped vectors of the page and ad on the reference space, are used to measure the similarity between the page $\boldsymbol{p}$ and the ad $\boldsymbol{a}$ over this projected space. In [23], similarity was measured by the Euler distance or dot product between $\mathbf{W}(\boldsymbol{p})$ and $\mathbf{W}(\boldsymbol{a})$. However, for consistency, here, we still use the cosine metric:

$$\mathbf{sim}^{\mathbf{w}}(\boldsymbol{p}, \boldsymbol{a}) = \frac{\sum_{j=1}^{N_w} \mathbf{sim}^{\mathbf{k}}(w_j, \boldsymbol{p}) \mathbf{sim}^{\mathbf{k}}(w_j, \boldsymbol{a})}{\sqrt{\sum_{j=1}^{N_w} \mathbf{sim}^{\mathbf{k}}(w_j, \boldsymbol{p})^2} \sqrt{\sum_{j=1}^{N_w} \mathbf{sim}^{\mathbf{k}}(w_j, \boldsymbol{a})^2}} \tag{8}$$

In Wikipedia matching, the content representation of each text is projected over the semantic reference space, such that the similarity between texts can be measured at the semantic level; thus, the problems encountered in keyword matching can be handled to a certain extent. However, from the above, we can see that for establishing a new vector for a given page, this approach needs to calculate the individual similarity between each reference article and the page, which is time-consuming. In a case where there is a large number of reference articles, such time-consuming computation is almost unacceptable in practical contextual advertising.

To improve the running performance, a straightforward way is to restrict the reference articles to a relatively small size, e.g., [22] only used 1,000 articles. However, as mentioned previously, this obviously leads to another problem, i.e., the limited coverage of semantic concepts, thus decreasing the accuracy of ad selection. Moreover, if we only use a small number of reference articles, the main advantages of Wikipedia, such as broad knowledge coverage about many different concepts, are almost lost completely. In short, Wikipedia matching usually ensure good accuracy by sacrificing high computational cost.

3.4 Selective Wikipedia matching

To solve the contradiction between effectiveness and efficiency, an improved Wikipedia matching approach to contextual advertising called SIWI: Selective Wikipedia Matching was proposed by our team in [31], which deals with reference article selection by the trade-off of concept coverage and computational cost using three selective matching strategies. This approach first chooses a large enough number of reference articles. Next, for each candidate ad, selective matching strategies are used to refine the selection of really relevant reference articles, in turn, generating a new vector expression of the ad over the refined reference article space. Likewise, for a target page, the same procedure is employed to obtain a new vector as well. Finally, based on the remapped vectors, the similarity between the page and the ad is calculated, upon which the top relevant ads to the page are selected.

This approach is built on top of the previously described Wikipedia matching, where the key improvement is to use the matching strategies to balance the semantic concept coverage and efficient matching. More concretely, the matching strategies are initiated by the following two aspects of observations. On one hand, to form a complete and accurate vector expression for a page (or an ad) over the new referential space, the concept coverage should be as broad as possible. However, most of the reference articles are actually not closely relevant to the page at all. Therefore, only the similarity computation needs to be carried out on the dimensions of really relevant articles, rather than the whole article set. On the other hand, the conceptual meaning of each article in Wikipedia describes is governed by the title of the article [14]. Thus, if the title (or a part of the title) of a reference article occurs in a page (or an ad), it is very likely that the article is related to the page; otherwise, it should be ignored.

Based on the above observations, it is believed that the contextual matching efficiency of page and ad can be improved by performing the similarity computation only on the relevant dimensions (i.e., on the relevant articles). Accordingly, the following three selective matching strategies were proposed in [31] to determine the relevant reference articles in Wikipedia matching.

Strategy 1. For a page $\boldsymbol{p}$ (or an ad), a reference article is considered to be **relevant** to the page $\boldsymbol{p}$ only if there is at least one title of the article contained in $\boldsymbol{p}$.

Strategy 2. For a page $\boldsymbol{p}$ (or an ad), a reference article is considered to be **relevant** to the page $\boldsymbol{p}$ only if there is at least one title of the article, satisfying the condition that all the keywords of the title are contained in $\boldsymbol{p}$.

Strategy 3. For a page $\boldsymbol{p}$ (or an ad), a reference article is considered to be **relevant** to the page $\boldsymbol{p}$ only if there is at least one title of the article, satisfying that there is at least one keyword of the title contained in $\boldsymbol{p}$.

After the selective matching process, the similarity calculation is conducted only on the full-text comparison between the relevant reference articles (which are in a small size) and the page, which significantly decreases the computation cost. As the reference articles are selected from the whole Wikipedia database, they will span a more diverse and broader coverage of semantic concepts (i.e., solving the problem of limited semantics coverage), thereby improving the accuracy of ad selection.

Despite the considerable improvement achieved in contextual matching, these approaches still cannot meet the practical requirements of contextual matching, especially in real-time applications. The limitations are mainly due to the involvement of the full-text comparison and the insufficient use of the rich semantic knowledge hidden in the Wikipedia thesaurus, as evidenced by the experimental results given in Sect. 5. Thus, this challenge motivates us to address the contextual matching from the perspective of semantic analysis without incurring in large computational overhead.

4 Methodology

In this section, we present a new contextual advertising approach by combining informative Wikipedia thesaurus knowledge with conventional keyword matching. In this work, we aim to improve the state-of-the-art contextual matching approaches (e.g., those mentioned in Sect. 3), so as to not only achieve a better accuracy of ad selection, but also to have a better running performance. The main framework of our approach is presented in Fig. 3.

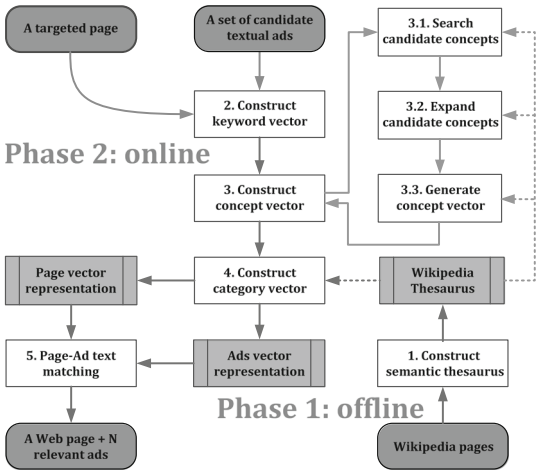


Fig. 3 Wikipedia-based contextual advertising framework

It comprises the following five steps to select relevant ads for a targeted page. In Step 1, we use a set of articles and categories chosen from Wikipedia to construct a semantic thesaurus that is the basis of generating concept vectors and category vectors. In Step 2, we use traditional keyword matching to generate a keyword feature vector for the targeted page. In Step 3, based on the Wikipedia concept information, we generate a concept feature vector for the targeted page by searching candidate concepts (Step 3.1), expanding candidate concepts (Step 3.2) and constructing a concept vector (Step 3.3). In Step 4, based on the concept feature vector and the Wikipedia category information, we generate a category feature vector for the targeted page. Now, the targeted page is represented as the three feature vectors: a keyword vector, a concept vector and a category vector. Likewise, for each candidate ad, we follow the same procedure and generate three similar feature vectors. Once obtaining the feature vectors for the targeted page and all the candidate ads, we calculate the relevance score between each ad and the page via a linear similarity fusion mechanism of the three feature vectors. Last, in Step 5, according to the relevance scores, we rank all the candidate ads in an order where the ads with the top-N highest ranking values are considered to be the top-N most relevant ads for the targeted page.

To improve the real-time response performance, we divide the whole matching process into two phases: offline Phase 1 and online Phase 2, and let most preprocessing work be completed in Phase 1. Phase 1 (i.e., Step 1 and Steps 2 to 4 on ads) includes the following three subtasks: (1) the preprocessing of Wikipedia articles and categories, (2) the preprocessing of all the candidate ads, and (3) establishing the feature vectors for all the ads. In contrast, Phase 2 (i.e., Steps 2 to 5 on pages) is to generate the required feature vectors for the targeted page, and to make the contextual matching between the page and all ad candidates in real time. Since only Phase 2 is completed online while Phase 1 is done offline, the total actual running cost, which is mainly dependent on the time used in Phase 2, would be dramatically decreased to improve the real-time matching performance.

4.1 Construction of Wikipedia thesaurus

The task in this process is to construct an easy-to-use thesaurus based on the link structure of Wikipedia, in which, there are three data structures: the concept graph, the category graph and the index of concepts.

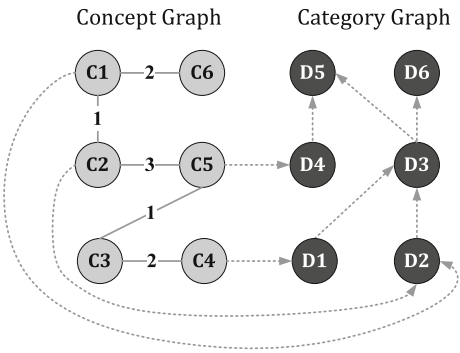


Fig. 4 A concept graph and a category graph

4.1.1 Concept graph

The concept graph is constructed based on the concept articles and the hyperlinks within them in Wikipedia, which is shown on the left side of Fig. 4. In the concept graph, each node represents a semantic concept, and each undirected edge between two concepts indicates that the two concepts are semantically related to each other. It should be noted that only the general pages are viewed as semantic concepts, not including redirect pages, disambiguation pages and category pages. Moreover, to reflect the relatedness between two concepts, a weighted connecting edge is created in the concept graph when two concept pages are interlinked to each other; and the weight is determined by the number of interlinks.

4.1.2 Category graph

The category graph is constructed based on concept articles and category articles, as well as the hyperlinks within categories, or between concepts and categories. It is shown on the right side of Fig. 4. In the category graph, each node represents a semantic concept or a category; each directed edge between two categories represents that one category belongs to the other (i.e., a subcategory of the other); and each directed edge from a concept to a category represents that the concept belongs to the category. As shown in Fig. 4, the category graph is a directed acyclic graph.

4.1.3 Index of concepts

In order to improve the performance of generating a concept vector, we also build an index over concept titles (as shown in Fig. 5). The index is constructed based on the example in Fig. 2. As shown in Fig. 5, the index is a many-to-many table defined from concept titles to semantic concepts, where all the concept titles are sorted in ascending order, with links from a title to all the concepts associated with this title. Given any query term, we can discover all the concept titles associated with this term quickly by using a binary search and further find out all the related semantic concepts.

4.2 Construction of concept vector

In Sect. 3.2, we describe traditional keyword matching, where each textual document is expressed by a keyword vector, and the lexicon closeness between textual documents is

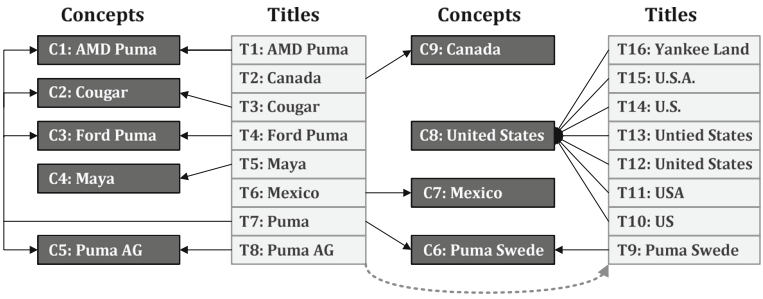


Fig. 5 An index over concept titles

measured by the similarity of the two keyword vectors. The keyword matching approach is undoubtedly a well-known textual processing model in information retrieval and natural language processing.

In our approach, we also use keyword vector representation to measure the textual similarity between pages and ads, by establishing a keyword vector $\mathbf{K}(\boldsymbol{p})$ for each page $\boldsymbol{p}$, and $\mathbf{K}(\boldsymbol{a})$ for each ad $\boldsymbol{a}$. However, as mentioned above, some problems of homonymy and polysemy, context mismatch, etc., may severely impact the accuracy of similarity measured by using keyword matching alone.

In this work, we aim to use the informative Wikipedia thesaurus knowledge along with keyword matching, to judge the overall semantic and lexical similarity between pages and ads and thus enhance the contextual matching of pages and ads. To do this, we introduce two new feature vectors, i.e., concept vector and category vector, and leverage them to measure the semantic relevance between pages and ads.

4.2.1 Search candidate concepts

The task of this process is to discover all the concepts mentioned in a targeted page (or an ad) and count the number of occurrences of all the concepts in the page. To do this, given a page $\boldsymbol{p}$, we first need to find out all the concept titles mentioned in $\boldsymbol{p}$. Such titles are called **candidate titles**, and the concepts named by the titles are called **candidate concepts**. We search candidate titles by using a similar method to that mentioned in [28], which comprises the following three steps: (1) split the context of the page $\boldsymbol{p}$ into a vector of keyword sequences by using punctuation such as semicolons, question and exclamation marks; (2) find out candidate titles in each term sequence via a window filtering condition (see [28] for detail), which can be completed efficiently based on the index over concept titles; and (3) filter the candidate titles to remove the titles subsumed by other candidate titles.

Upon completion of the above process, we obtain (1) a set of candidate titles mentioned in $\boldsymbol{p}$, represented by $\mathbf{titles}(\boldsymbol{p})$ and (2) a set of candidate concepts, represented by $\mathbf{cots}(\boldsymbol{p})$, which is generated by searching the index over the concept titles using each title in $\mathbf{titles}(\boldsymbol{p})$. Besides, we also can obtain the occurrence frequency of a candidate title $\boldsymbol{t}$ ($\boldsymbol{t} \in \mathbf{titles}(\boldsymbol{p})$) in the page $\boldsymbol{p}$, represented by $\mathbf{count}(\boldsymbol{t}, \boldsymbol{p})$.

As mentioned in Sect. 3.4, each article in Wikipedia describes one concept, and the concept is named after the title of the article. Thus, if the title of a concept occurs in a page, it is very likely that the concept is related to the page; otherwise, it should be deemed less related. Thus, the above process of searching candidate titles in a page actually determines a set of concepts semantically associated with the page, i.e., each candidate concept $\boldsymbol{c}$ in $\mathbf{cots}(\boldsymbol{p})$

should be semantically related to the page $\boldsymbol{p}$ to a certain degree, similar to the phenomenon of the relatedness between keywords to their pages. This allows the semantic expression of a page (especially an ad of short length) to be enriched substantially by introducing a concept vector based on the concepts associated with the page. To do this, we need to compute the occurrence frequency of a candidate concept $\boldsymbol{c}$ related to a candidate title $\boldsymbol{t}$ in the page $\boldsymbol{p}$.

For a candidate title $\boldsymbol{t}$, the frequency value of occurrences of any candidate concept $\boldsymbol{c}$ related to $\boldsymbol{t}$ in the page $\boldsymbol{p}$ can be computed as follows: (1) if $\boldsymbol{t}$ is not a title of the candidate concept $\boldsymbol{c}$, then $\mathbf{count}(\boldsymbol{c}, \boldsymbol{t}, \boldsymbol{p}) = 0$; (2) if $\boldsymbol{c}$ is the only candidate concept that uses $\boldsymbol{t}$ as title, then $\mathbf{count}(\boldsymbol{c}, \boldsymbol{t}, \boldsymbol{p}) = \mathbf{count}(\boldsymbol{t}, \boldsymbol{p})$; and (3) if $\boldsymbol{t}$ is an ambiguous title, i.e., there is a group of candidate concepts, represented by $\mathbf{cots}(\boldsymbol{t})$, each of which satisfying that at least one of its titles is identical to $\boldsymbol{t}$ (obviously, $\mathbf{cots}(\boldsymbol{t}) \in \mathbf{cots}(\boldsymbol{p})$), and $\boldsymbol{c} \in \mathbf{cots}(\boldsymbol{t})$, i.e., $\boldsymbol{t}$ is a title of the candidate concept $\boldsymbol{c}$, then

$$\mathbf{count}(\boldsymbol{c}, \boldsymbol{t}, \boldsymbol{p}) = \frac{\mathbf{sim^k}(\boldsymbol{c}, \boldsymbol{p})\, \mathbf{count}(\boldsymbol{t}, \boldsymbol{p})}{\sum_{u \in \mathbf{cots}(\boldsymbol{t})} \mathbf{sim^k}(u, \boldsymbol{p})} \tag{9}$$

where $\mathbf{sim^k}(\boldsymbol{c}, \boldsymbol{p})$ denotes the keyword similarity between the page $\boldsymbol{p}$ and the article corresponding to $\boldsymbol{c}$, and $\mathbf{sim^k}(u, \boldsymbol{p})$ is similar.

Third, as each concept may contain several titles (see Fig. 2), we sum the occurrence frequency value of the concept related to each of its titles in the page, to compute the actual occurrence frequency of the candidate concept in the given page. For any candidate concept $\boldsymbol{c}$ in $\mathbf{conceps}(\boldsymbol{p})$, the occurrence frequency value of $\boldsymbol{c}$ in the targeted page $\boldsymbol{p}$ is computed as follows:

$$\mathbf{count}(\boldsymbol{c}, \boldsymbol{p}) = \sum_{\boldsymbol{t} \in \mathbf{titles}(\boldsymbol{c})} \mathbf{count}(\boldsymbol{c}, \boldsymbol{t}, \boldsymbol{p}) \tag{10}$$

where $\mathbf{titles}(\boldsymbol{c})$ denotes a set of all the titles of the concept $\boldsymbol{c}$, and obviously, $\mathbf{titles}(\boldsymbol{c}) \in \mathbf{titles}(\boldsymbol{p})$.

Algorithm 1 details the process of searching all the candidate concepts in a targeted page and computing the occurrence frequency values of all the candidate concepts in the page. Let $\mathbf{cots}(\boldsymbol{p}) = \left\{c_j^p\right\}_{j=1}^{N_c^p}$, where N_c^p denotes the number of candidate concepts. Now, we obtain a set $F_C(\boldsymbol{p}) = \left\{\mathbf{count}\left(c_j^p, \boldsymbol{p}\right)\right\}_{j=1}^{N_c^p}$, consisting of occurrence frequency values of all the candidate concepts in the page $\boldsymbol{p}$. Likewise, for a candidate ad $\boldsymbol{a}$, we can also obtain $\mathbf{cots}(\boldsymbol{a})$ and $F_C(\boldsymbol{a})$ by using Algorithm 1.

In the above process, synonymy keywords would be mapped into the same concept, thereby solving the context mismatch caused by synonyms. Moreover, for an ambiguous keyword (polysemy) in a page, which may be associated with different semantic concepts, a further full-text comparison [i.e., Eq. (5)] would be made between each of these concepts and the page, to determine the semantic concept distribution represented in the page in terms of various weights. For example, the page about "puma, a feline resembling a lion" would be assigned a higher weight on the concept "cougar" than on other less related concepts titled by "puma"; thus, the ambiguous keyword "puma" in this page is considered to be more related to "cougar." Hence, the problem of homonymy and polysemy can be tackled effectively.

Unlike Wikipedia matching to match all the reference articles for a page, Algorithm 1 only needs to make a full-text comparison on a small part of all reference articles, i.e., on the reference articles whose titles are mentioned in the page and represented as polysemy. As a

Algorithm 1: Searching Candidate Concepts

Input: (1) a vector $\mathbf{words}(\boldsymbol{p})$, of keyword sequences, and (2) an index over concept titles.
Output: a frequency set $F_C(\boldsymbol{p})$, consisting of frequency values of all the candidate concepts in the page $\boldsymbol{p}$.
begin

search each term sequence in $\mathbf{words}(\boldsymbol{p})$ to obtain a set $\mathbf{titles}(\boldsymbol{p}) = \{t_j\}_{j=1}^{N_t^p}$, consisting of candidate titles, and count the number of occurrences of each candidate title in $\mathbf{titles}(\boldsymbol{p})$ to obtain a set $F_t(\boldsymbol{p}) = \{\mathbf{count}(t_j, \boldsymbol{p})\}_{j=1}^{N_t^p}$.

filter candidate titles to remove each title in $\mathbf{titles}(\boldsymbol{p})$ subsumed by another candidate title in $\mathbf{titles}(\boldsymbol{p})$.

$F_C(\boldsymbol{p}) \leftarrow empty$.

for *each candidate title* $\boldsymbol{t} \in \mathbf{titles}(\boldsymbol{p})$ **do**

obtain a set $\mathbf{cots}(\boldsymbol{t})$ of candidate concepts named by the candidate title $\boldsymbol{t}$ based on the index over concept titles.

for *each candidate concept* $\boldsymbol{c} \in \mathbf{cots}(\boldsymbol{t})$ **do**

$$\mathbf{count}(\boldsymbol{c}, \boldsymbol{t}, \boldsymbol{p}) \leftarrow \mathbf{sim^k}(\boldsymbol{c}, \boldsymbol{p}) \cdot \mathbf{count}(\boldsymbol{t}, \boldsymbol{p}) \cdot \left(\sum_{u \in \mathbf{cots}(\boldsymbol{t})} \mathbf{sim^k}(u, \boldsymbol{p}) \right)^{-1}$$, where $\mathbf{sim^k}(\boldsymbol{c}, \boldsymbol{p})$ denotes the keyword similarity between $\boldsymbol{p}$ and $\boldsymbol{c}$; and $\mathbf{count}(\boldsymbol{t}, \boldsymbol{p})$ denotes the frequency value of occurrences of $\boldsymbol{c}$ related to $\boldsymbol{t}$ in $\boldsymbol{p}$.

if *there is* $\mathbf{count}(\boldsymbol{c}, \boldsymbol{p}) \in F_C(\boldsymbol{p})$ *corresponding to the concept* $\boldsymbol{c}$ **then**
$\mathbf{count}(\boldsymbol{c}, \boldsymbol{p}) \leftarrow \mathbf{count}(\boldsymbol{c}, \boldsymbol{p}) + \mathbf{count}(\boldsymbol{c}, \boldsymbol{t}, \boldsymbol{p})$.

else
$\mathbf{count}(\boldsymbol{c}, \boldsymbol{p}) \leftarrow \mathbf{count}(\boldsymbol{c}, \boldsymbol{t}, \boldsymbol{p})$. $F_C(\boldsymbol{p}) \leftarrow F_C(\boldsymbol{p}) \cup \{\mathbf{count}(\boldsymbol{c}, \boldsymbol{p})\}$.

return $F_C(\boldsymbol{p})$.

result, the computational cost of searching candidate concepts is reduced in comparison with Wikipedia matching.

4.2.2 Expand candidate concepts

In general, since there are fewer keywords or phrases contained in an ad than a generic document due to its limited size, the number of candidate concepts extracted from an ad should be smaller, i.e., the frequency set $F_C(\boldsymbol{a})$ for an ad $\boldsymbol{a}$ should be of a smaller size. This will lead to a low intersection of candidate concepts between pages and ads, similar to the problem of the low intersection of keywords encountered in keyword matching, thereby reducing the accuracy of using concept information to measure the similarity between pages and ads. Thus, the process of expanding candidate concepts is employed to expand the concept representation $F_C(\boldsymbol{a})$ of an ad $\boldsymbol{a}$ (or a page) with other semantically related concepts.

As described in [16,27], given that there is a connection between two concepts in Wikipedia, it is likely that the two concepts share common topics. More mathematically, given an ad $\boldsymbol{a}$, and $\boldsymbol{c_1}$ being one of its candidate concepts and $\boldsymbol{c_2}$ is not, i.e., $\boldsymbol{c_1} \in \mathbf{cots}(\boldsymbol{a})$ and $\boldsymbol{c_2} \notin \mathbf{cots}(\boldsymbol{a})$, if there is a connection between $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$, then it is likely that the concept $\boldsymbol{c_2}$ is also semantically related to the ad $\boldsymbol{a}$. Therefore, the task of expanding candidate concepts can be implemented based on the rich connections within the concept graph.

First, we define how to calculate the occurrence frequency value (or called related frequency value) of a non candidate concept in an ad (or a page). Let $\mathbf{cots}(\boldsymbol{c})$ be all the concepts in-linked from a candidate concept $\boldsymbol{c}$ of an ad $\boldsymbol{a}$. Then, the related occurrence frequency of each concept $\boldsymbol{e}$ in $\mathbf{cots}(\boldsymbol{c})$ in ad $\boldsymbol{a}$ is computed as:

$$\mathbf{count}(\boldsymbol{e}, \boldsymbol{a}) = \frac{\mathbf{num^k}(\boldsymbol{e}, \boldsymbol{c})\mathbf{count}(\boldsymbol{c}, \boldsymbol{a})}{\sum_{u \in \mathbf{cots}(\boldsymbol{c})} \mathbf{num^k}(u, \boldsymbol{c})} \tag{11}$$

Algorithm 2: Expanding Candidate Concepts

Input: (1) an initial set $F_C(\boldsymbol{a})$ of frequency values of all the candidate concepts mentioned in the ad $\boldsymbol{a}$; (2) a concept graph; and (3) a threshold value μ_E used to stop breadth-first graph traversal.

Output: an expanded set $F_E(\boldsymbol{a})$ consisting of frequency values of all the candidate concepts and the new introduced concepts semantically related to the candidate concepts.

begin

$Q \leftarrow empty$. // Q is a first-in first-out queue.

for *each candidate concept* $\boldsymbol{c}$ *associated with* $F_C(\boldsymbol{a})$ **do**

$Q \leftarrow Q \cup \{\boldsymbol{c}\}$. // use candidate concepts as start points.

while Q *is not empty* **do**

pop up the tail element $\boldsymbol{c}$ in Q.

based on the concept graph, obtain all the concepts $\mathbf{cots}(\boldsymbol{c})$, each of which is semantically related to the concept $\boldsymbol{c}$.

for *each concept* $\boldsymbol{e} \in \mathbf{cots}(\boldsymbol{c})$ **do**

$$\mathbf{count}(\boldsymbol{e}, \boldsymbol{a}) \leftarrow \left(\mathbf{num^k}(\boldsymbol{e}, \boldsymbol{c}) \cdot \mathbf{count}(\boldsymbol{c}, \boldsymbol{a})\right) \cdot \left(\sum_{u \in \mathbf{cots}(\boldsymbol{c})} \mathbf{num^k}(u, \boldsymbol{c})\right)^{-1}$$, where $\mathbf{num^k}(\boldsymbol{e}, \boldsymbol{c})$ denotes the connection value from $\boldsymbol{e}$ to $\boldsymbol{c}$ in the concept graph; and $\mathbf{count}(\boldsymbol{c}, \boldsymbol{a})$ denotes the number of occurrences of $\boldsymbol{c}$ in the ad $\boldsymbol{a}$.

if $\mathbf{count}(\boldsymbol{e}, \boldsymbol{a}) > \mu_E$ *and* $\boldsymbol{c}$ *not visited* **then**

$F_C(\boldsymbol{a}) \leftarrow F_C(\boldsymbol{a}) \cup \{\mathbf{count}(\boldsymbol{e}, \boldsymbol{a})\}$. // the frequency set is renewed by adding the frequency value of the new expanded concept $\boldsymbol{e}$.

$Q \leftarrow Q \cup \{\boldsymbol{e}\}$.

return $F_C(\boldsymbol{a})$.

where $\mathbf{num^k}(\boldsymbol{e}, \boldsymbol{c})$ denotes the number of hyperlinks between the two concepts $\boldsymbol{e}$ and $\boldsymbol{c}$; and $\mathbf{num^k}(u, \boldsymbol{c})$ has a similar meaning.

Second, we need to expand the initial frequency set $F_C(\boldsymbol{a})$ generated by Algorithm 1 to introduce the related occurrence frequency values of the concepts that are reachable to the candidate concepts in the concept graph. Actually, the expansion of candidate concepts is a breadth-first traversal process over the concept graph, using the candidate concepts as start traversal nodes. Algorithm 2 details the traversal process.

In Algorithm 2, $F_C(\boldsymbol{a})$ may be expanded not only with the concepts semantically related to the candidate concepts (i.e., with the concepts linked by the candidate concepts), but also with the concepts semantically related to the newly introduced concepts (i.e., with the concepts reachable to the candidate concepts). Likewise, for a page $\boldsymbol{p}$, by using Algorithm 2, $F_C(\boldsymbol{p})$ can also be expanded. In Algorithm 2, μ_E is an important parameter used to control the traversal depth over the concept graph. A smaller μ_E value will lead to a better result on expanding candidate concepts, but results in a longer search time, whereas a larger μ_E value will achieve a shorter expansion result but a smaller running cost. Thus, we choose different μ_E values for the expansion on ads and for the expansion on pages:

1. We use a smaller μ_E value ($\mu_E = 0.02$) for the expansion on ads, so as to obtain a good expansion effectiveness, i.e., to obtain more related concepts. This is a due to the limited size of an ad and the off-line generation for the feature vector of an ad, resulting in our prior consideration of the expansion effect.
2. We use a greater μ_E value ($\mu_E = 1$) for the expansion on the given page so as to obtain good running performance. This is due to the online generation of the feature vector of the page, resulting in our prior consideration of efficiency.

By using Algorithm 2, the feature representation of an ad would be further enriched with more semantically related concepts, thereby easing the problem of the low intersection of concepts between pages and ads. For example, given a page about "travel," and an ad about "hotel" where there are such concepts as "hotel" and "phone," the ad on the surface is considered not to be related to the page due to the low intersection of the same keywords or concepts between them. However, because there is a connection between the concepts "travel" and "hotel," after expansion by Algorithm 2, the ad would be represented by associations with such concepts as "hotel," "travel," thereby making the page semantically related to the ad to some extent. Moreover, from the above example, we see that the problem of the low intersection of keywords between pages and ads is accordingly solved after the candidate concept expansion.

In Algorithm 2, the time-consuming full-text matching no longer needs to be conducted between pages and reference articles. Moreover, due to the parameter μ_E assigned with a smaller value, the expansion of candidate concepts over a page should be efficient, only needing to visit a very small part of all the reference concept articles. Thus, this process ensures the satisfactory running performance of constructing a concept vector.

4.2.3 Generate concept vector

After the above two steps, we obtain a set of occurrence frequency values of related semantic concepts for a page (or an ad). By combining the concept frequency set with the *tf-idf* weight, we then construct a corresponding concept feature vector.

First, we define the *tf-idf* value of a concept related to a page. Let $\boldsymbol{p}$ be a targeted page and $\boldsymbol{c}$ a concept related to $\boldsymbol{p}$. We define $\mathbf{tf}(\boldsymbol{c}, \boldsymbol{p}) = (\mathbf{count}(\boldsymbol{c}, \boldsymbol{p}))/(|\mathbf{cots}(\boldsymbol{p})|)$, where $\mathbf{count}(\boldsymbol{c}, \boldsymbol{p})$ is the number of occurrences of the concept $\boldsymbol{c}$ in the page $\boldsymbol{p}$, and $|\mathbf{cots}(\boldsymbol{p})|$ is the number of all the concepts related to $\boldsymbol{p}$. Then, the *tf-idf* value of the concept $\boldsymbol{c}$ related to the page $\boldsymbol{p}$ is calculated as:

$$\mathbf{tfidf}(\boldsymbol{c}, \boldsymbol{p}) = \mathbf{tf}(\boldsymbol{c}, \boldsymbol{p}) \log\left(\frac{|\mathcal{P}|}{|\{p : p \in \mathcal{P}, \boldsymbol{c} \in \mathbf{cots}(p)\}|}\right) \tag{12}$$

Next, we construct a concept vector, which consists of the *tf-idf* values of all the concepts related to $\boldsymbol{p}$. We assume that $\mathbf{cots}(\boldsymbol{p}) = \left\{c_j^p\right\}_{j=1}^{N_c^p}$, where N_c^p denotes the number of all the concepts related to $\boldsymbol{p}$, generated by Algorithm 1 and expanded by Algorithm 2. A feature vector for the targeted page $\boldsymbol{p}$ (called a **concept vector**) is constructed based on Eq. (12), represented by $\mathbf{C}(\boldsymbol{p}) = \left\{\mathbf{tfidf}\left(c_j^p, \boldsymbol{p}\right)\right\}_{j=1}^{N_c^p}$. Similarly, for a candidate ad $\boldsymbol{a}$, assuming that $\mathbf{cots}(\boldsymbol{a}) = \left\{c_j^a\right\}_{j=1}^{N_c^a}$, we can also obtain a concept vector for the ad $\boldsymbol{a}$ as follows: $\mathbf{C}(\boldsymbol{a}) = \left\{\mathbf{tfidf}\left(c_j^a, \boldsymbol{a}\right)\right\}_{j=1}^{N_c^a}$.

Based on the new concept vector expressions $\mathbf{C}(\boldsymbol{p})$ and $\mathbf{C}(\boldsymbol{a})$ for the page $\boldsymbol{p}$ and the ad $\boldsymbol{a}$, we can obtain a new similarity over concept space between $\boldsymbol{p}$ and $\boldsymbol{a}$:

$$\mathbf{sim^c}(\boldsymbol{p}, \boldsymbol{a}) = \frac{\sum_{\forall i \forall j, c_i^p = c_j^a} \mathbf{count}\left(c_i^p, \boldsymbol{p}\right) \mathbf{count}\left(c_j^a, \boldsymbol{a}\right)}{\sqrt{\sum_{j=1}^{N_c^p} \mathbf{count}\left(c_j^p, \boldsymbol{p}\right)^2} \sqrt{\sum_{j=1}^{N_c^a} \mathbf{count}\left(c_j^a, \boldsymbol{a}\right)^2}} \tag{13}$$

4.3 Construction of category vector

In Wikipedia, category information provides additional thesaurus knowledge to reflect the semantic relationship between concepts. For a category, if there are some concepts belonging to the category and semantically related to a page, then it is likely that the category is also semantically related to the page. For example, a page about "puma, a feline resembling a lion" is related to the concept "cougar," resulting in its semantic relatedness with a category "mammal" (that the concept belongs to). Thus, the category information can also be used to enrich the semantic representation of pages and ads. In this subsection, by utilizing the concept feature vector and the category graph, which reflects the hierarchical relation between concepts and categories or within categories, we describe how to construct a category vector for a targeted page (or an ad), so as to further enrich the semantic representation.

4.3.1 Search related categories

A category is considered to be semantically related to a page (or an ad), only if there exists at least one concept belonging to the category and related to the page as well. The task of this process of searching related categories is to find out all the categories semantically related to a page (or an ad).

First, we define how to compute the frequency (called related frequency value) of occurrences of a category in a page (or an ad). Let $\mathbf{cots}(\boldsymbol{d})$ be a set of all the concepts that belong to a category $\boldsymbol{d}$, and $\mathbf{cats}(\boldsymbol{d})$ a set of all the immediate subcategories that belong to $\boldsymbol{d}$ (i.e., there is a hyperlink from each category in $\mathbf{cats}(\boldsymbol{d})$ to $\boldsymbol{d}$). Both $\mathbf{cots}(\boldsymbol{d})$ and $\mathbf{cats}(\boldsymbol{d})$ can be determined by the category graph. Then, we define the related frequency value of the category $\boldsymbol{d}$ appearing in the page $\boldsymbol{p}$ as follows:

$$\mathbf{count}(\boldsymbol{d}, \boldsymbol{p}) = \sum_{c \in \mathbf{cots}(\boldsymbol{d})} \frac{\mathbf{count}(c, \boldsymbol{p})}{\alpha_C} + \sum_{d \in \mathbf{cats}(\boldsymbol{d})} \frac{\mathbf{count}(d, \boldsymbol{p})}{\alpha_D} \tag{14}$$

where α_C and α_D are two attenuation coefficients, which are used to balance the importance of frequency values of categories in different depths.

Next, we detail how to generate a set of frequency values of related categories appearing in a page (or an ad), shown in Algorithm 3. From Algorithm 3, we know that for the generation of a category frequency set, we need to conduct a breadth-first traversal over the category graph, using the concepts associated with an input concept frequency set as start nodes. In Algorithm 3, the parameter μ_D is used to filter out the categories with lower frequency values, which, similar to Algorithm 2, is negatively correlated with the size of a category frequency set and positively with the running performance. Thus, we assign a smaller value to μ_D for searching related categories for ads ($\mu_D = 0.05$), and a greater value for pages ($\mu_D = 0.5$).

Using Algorithm 3, the semantic representation for a page (or an ad) would be further enriched with categories, each of which is semantically related to the page to a certain extent. As such, given a page and an ad, which are semantically related to each other but do not share common keywords or concepts, they may be enriched with some common categories, thereby being shifted closer to each other at a higher semantic level (i.e., the categorical hierarchy). For example, let us consider a page about "smart dolphin" and an ad about "monkey play." As no keywords or concepts are shared, the similarity computation based on Eq. (5) or (13) may be deemed to be void. However, according to concept information, both "monkey" and "dolphin" belong to the same category "mammal," so the ad and the page would be considered to be relevant to each other.

Algorithm 3: Searching Related Categories

Input: (1) a set $F_C(\boldsymbol{p})$ of frequency values of occurrences of related concepts in the page $\boldsymbol{p}$; (2) the category graph, which is a directed acyclic graph G_D; and (3) a threshold value μ_D.
Output: a frequency set $F_D(\boldsymbol{p})$, consisting of frequency values of all the categories related to the page $\boldsymbol{p}$.
begin
  $F_D(\boldsymbol{p}) \leftarrow empty$. // initialize the category frequency set.
  for *each category* $\boldsymbol{d}$ *in the category graph* **do**
    $\mathbf{depth}(\boldsymbol{d}) \leftarrow$ the length of the longest path from the category $\boldsymbol{d}$ to its top parent category.
    $\mathbf{count}(\boldsymbol{d}, \boldsymbol{p}) \leftarrow 0$.
  for *each concept* $\boldsymbol{c}$ *associated with* $F_C(\boldsymbol{p})$ **do**
    based on the category graph, obtain the category $\boldsymbol{d}$ that the concept $\boldsymbol{c}$ belongs to.
    $\mathbf{count}(\boldsymbol{d}, \boldsymbol{p}) \leftarrow \mathbf{count}(\boldsymbol{d}, \boldsymbol{p}) + \mathbf{count}(\boldsymbol{c}, \boldsymbol{p}) \cdot \frac{1}{\alpha_C}$.
  for $\boldsymbol{h}$ *from* $\left(\max_{\boldsymbol{d} \in G_D} (\mathbf{depth}(\boldsymbol{d}))\right)$ *to 1* **do**
    based on the category graph, obtain a set $\mathbf{cats}(\boldsymbol{h})$, consisting of all the categories of the depth $\boldsymbol{h}$.
    for *each category* $\boldsymbol{d} \in \mathbf{cats}(\boldsymbol{h})$ **do**
      based on the category graph, obtain a set $\mathbf{cats}(\boldsymbol{d})$, which consists of all the immediate subcategories of the category $\boldsymbol{d}$.
      $\mathbf{count}(\boldsymbol{d}, \boldsymbol{p}) \leftarrow \mathbf{count}(\boldsymbol{d}, \boldsymbol{p}) + \left(\sum_{d \in \mathbf{cats}(\boldsymbol{d})} \mathbf{count}(d, \boldsymbol{p}) \cdot \frac{1}{\alpha_D}\right)$
      if $\mathbf{count}(\boldsymbol{d}, \boldsymbol{p}) > \mu_D$ **then**
        $F_D(\boldsymbol{p}) \leftarrow F_D(\boldsymbol{p}) \cup \{\mathbf{count}(\boldsymbol{d}, \boldsymbol{p})\}$.
  return $F_D(\boldsymbol{p})$. // return the category frequency set.

Algorithm 3 needs to scan all the category nodes in the category graph. However, such a category access operation does not take a long time since the number of categories is relatively small. Thus, Algorithm 3, as a part of feature representation for a page (or an ad), would not significantly impact the matching performance.

4.3.2 Generate category vector

Let $\boldsymbol{p}$ be a targeted page and $\boldsymbol{d}$ a category related to $\boldsymbol{p}$. We define $\mathbf{tf}(\boldsymbol{d}, \boldsymbol{p}) = (\mathbf{count}(\boldsymbol{d}, \boldsymbol{p}))/(|\mathbf{cats}(\boldsymbol{p})|)$, where $\mathbf{count}(\boldsymbol{d}, \boldsymbol{p})$ is the frequency value of occurrences of the category $\boldsymbol{d}$ in $\boldsymbol{p}$, and $|\mathbf{cats}(\boldsymbol{p})|$ is the number of categories related to $\boldsymbol{p}$. Then, the *tf-idf* value of the category $\boldsymbol{d}$ related to the page $\boldsymbol{p}$ is computed as:

$$\mathbf{tfidf}(\boldsymbol{d}, \boldsymbol{p}) = \mathbf{tf}(\boldsymbol{d}, \boldsymbol{p}) \log\left(\frac{|\mathcal{P}|}{|\{p : p \in \mathcal{P}, \boldsymbol{d} \in \mathbf{cats}(p)\}|}\right) \tag{15}$$

Next, we construct a category vector, consisting of *tf-idf* values of all the categories related to $\boldsymbol{p}$. We assume that $\mathbf{cats}(\boldsymbol{p}) = \left\{d_j^p\right\}_{j=1}^{N_d^p}$, where N_d^p denotes the number of all the categories related to $\boldsymbol{p}$, generated by Algorithm 3. Based on Eq. (15), a feature vector for the page $\boldsymbol{p}$ (called a **category vector**) is constructed, represented by $\mathbf{D}(\boldsymbol{p}) = \left\{\mathbf{tfidf}\left(d_j^p, \boldsymbol{p}\right)\right\}_{j=1}^{N_d^p}$. Similarly, for a candidate ad $\boldsymbol{a}$, assuming that $\mathbf{cats}(\boldsymbol{a}) = \left\{d_j^a\right\}_{j=1}^{N_d^a}$, we can also obtain a concept vector for the ad $\boldsymbol{a}$ as follows: $\mathbf{D}(\boldsymbol{a}) = \left\{\mathbf{tfidf}\left(d_j^a, \boldsymbol{a}\right)\right\}_{j=1}^{N_d^a}$. Last, based on the

category vector $\mathbf{D}(\boldsymbol{p})$ and $\mathbf{D}(\boldsymbol{a})$ for the page $\boldsymbol{p}$ and the ad $\boldsymbol{a}$, we can obtain a new similarity over category space:

$$\mathbf{sim^d}(\boldsymbol{p}, \boldsymbol{a}) = \frac{\sum_{\forall i \forall j, d_i^a = d_j^p} \mathbf{count}\left(d_i^a, \boldsymbol{a}\right) \mathbf{count}\left(d_j^p, \boldsymbol{p}\right)}{\sqrt{\sum_{j=1}^{N_d^a} \mathbf{count}\left(d_j^a, \boldsymbol{a}\right)^2} \sqrt{\sum_{j=1}^{N_d^p} \mathbf{count}\left(d_j^p, \boldsymbol{p}\right)^2}} \tag{16}$$

4.4 Similarity fusion

Now, each page (or ad) has been represented as three feature vectors: a keyword vector, a concept vector and a category vector. When measuring the relevance between an ad and a page, we combine the similarity values calculated by using the three vectors. For a page $\boldsymbol{p}$ and an ad $\boldsymbol{a}$, the relevance between them can be computed as:

$$\mathbf{sim}(\boldsymbol{a}, \boldsymbol{p}) = w_k \mathbf{sim^k}(\boldsymbol{a}, \boldsymbol{p}) + w_c \mathbf{sim^c}(\boldsymbol{a}, \boldsymbol{p}) + w_d \mathbf{sim^d}(\boldsymbol{a}, \boldsymbol{p}) \tag{17}$$

where (w_k, w_c, w_d) control the weight of the concept vector and category vector in the semantic matching between page and ad, and the balance between keyword matching and semantic matching, and (w_k, w_c, w_d) satisfy the constraints: $1 \geq (w_k, w_c, w_d) \geq 0$ and $(w_k + w_c + w_d) = 1.0$.

In our approach, before ad selection, the feature vectors for all the candidate ads are established in advance (offline); thus, the computation cost of ad selection for a page mainly depends on the time spent in conducting: (1) feature representation of the page; and (2) similarity computation between the page and each ad in an ad database. Now, we analyze their time complexities, respectively. Sections 4.2 and 4.3 show that searching candidate concepts is the most time-consuming operation in the feature representation of a page. Let N_k^w be the average number of keywords contained in each concept article, N_k^p the number of keywords in the page, N_k^g the number of ambiguous keywords in the page, and N_c^g the average number of concepts associated with each ambiguous keyword in the page. Then, the time complexity of generating feature vectors for a page is equal to $\mathbf{O}(N_k^w N_k^p N_k^g N_c^g)$. Obviously, $0 \leq (N_k^g N_c^g) \ll N_w$, where N_w denotes the number of all the reference articles chosen from Wikipedia. From Eq. (1), we know that the process of selecting relevant ads for a page is implemented by conducting the similarity computation between the feature vectors of the page and the feature vectors of each ad in a candidate ad database. Let N_a be the number of candidate ads, N_k^a the average number of keywords in each ad, and N_k^p the number of keywords in the page. Then, the time complexity of selecting relevant ads for the page is equal to $\mathbf{O}((N_k^p + N_k^a)N_a)$. Thus, in our approach, the computation complexity of ad selection for a page is equal to $\mathbf{O}(N_k^w N_k^p N_k^g N_c^g + (N_k^p + N_k^a)N_a)$.

5 Experiments

In this section, we experimentally evaluate our approach. In the first two subsections, we present data and the evaluation methodology, and candidate strategies used as a comparison with our approach. Then, in the subsequent subsections, we evaluate our approach from the following four aspects: (1) the degree to which expanded concepts and categories are relevant to their pages (relevance ratio evaluation); (2) time spent selecting ads for a page (efficiency evaluation); (3) contextual advertising effectiveness over general pages (effectiveness evaluation); and (4) contextual advertising effectiveness over ambiguous pages (ambiguous evaluation).

Table 1 Dataset characteristic

Item	Number
General pages in dataset	80
Ambiguous pages in dataset	35
Candidate ads in dataset	10,244
Wikipedia reference articles (concepts)	730,500
Wikipedia reference categories	24,000

5.1 Data and evaluation methodology

We conducted experiments to evaluate our approach using a dataset that contains 80 generic pages, 35 ambiguous pages and 10,224 candidate ads. Detailed characteristics of this dataset are shown in Table 1. Below, we briefly introduce how this dataset was prepared (a more detailed description can be seen in [31], a previous paper of our team).

Generic pages were downloaded from the Internet, with care being taken to ensure that there was an even representation of various areas such as business, electronics, entertainment and health. Each page was processed by an HTML extraction tool [6], keeping only the title and the main text content. Next, function words (e.g., articles and prepositions) were removed from each page. Last, each keyword of the page was processed by a stemming algorithm [6] that truncates the suffix of the word and reduces it to a stem.

To obtain candidate ads, we queried the Google search engine with a list of nouns which were selected from the word library of Youdao Dictionary.[4] Next, by processing the returned pages with blocks of sponsored ads supplied by Google Adwords,[5] we collected 10,224 textual ads. Last, these ads were put through the same process as the pages.

From Wikipedia, we selected a set of articles and categories. We first downloaded the compressed XML file[6] and imported it into a MySQL database by using an XML extraction tool,[7] and then selected articles and categories from the database. Next, these articles were put through the process of tokenization, stemming and function words filtering. However, as pointed out in [27], some of the obtained articles and categories (e.g., "2010s") are meaningless (they are only used for management or administration). Thus, we used the rules mentioned in [27] to filter these useless entities and, consequently, obtained 730,500 articles and 24,000 categories.

Since the purpose of contextual advertising is to select the top-N most relevant ads for a page, we evaluated the average precision for the top-1, top-3 and top-5. For each page, we collected human judgment scores that describe the relevance of ads selected by each of the candidate strategies (see Table 2). The human judgment scores for the relevance of embedded ads to a page were determined by using a scoring method similar to that in [14,15,18,19,31] and were completed by at least two assessors on a scale between 0.0 and 1.0. The detailed scoring grade is given as follows:

1. Fully relevant (1.0), if the embedded ad is related to the main subject of a page directly. For example, if the page is about "travel" and the ad is about "travel service," it would be scored 1.0.

[4] Netease Youdao Dictionary—http://dict.youdao.com.

[5] Google Adwords—http://adwords.google.com.

[6] Wikimedia dump—http://download.wikimedia.org/enwiki.

[7] Mwdumper—www.mediawiki.org/wiki/Mwdumper.

2. Somewhat relevant (0.5), if the embedded ad is related to the secondary subject of a page, or related to the topic of a page in a general way. For example, if the page is about "Cambridge University" and the ad is about "hotels near Cambridge University," it would be scored 0.5.
3. Irrelevant (0.0), if the embedded ad has no relevance to a page. For example, if the page is about "Puma, an American feline" and the ad is about "Puma shoes," it would be scored 0.0.

We invited 52 undergraduate students from the Department of Computer Science, all of whom had sufficient Internet browsing experience and judgment ability to conduct the evaluation, to act as assessors to score the embedded ads based on the relevance of each ad to the targeted page. Each embedded ad was first scored by assessors independently, and then, to determine the final relevance score of the ad to its page, we averaged the relevance scores given by two assessors for each ad.

5.2 Candidate strategies

In our experiments, we used the contextual advertising strategy solely based on keyword vector as a baseline (that is, traditional keyword matching). Moreover, other contextual advertising strategies based on different linear combinations of keyword vectors, concept vectors and category vectors are presented in the first five rows in Table 2, where the fifth KCD strategy is recommended in our paper. In the five strategies, (w_k, w_c, w_d) in Eq. (17) were set in the following simple way.

1. For the strategy which combines keyword vectors and concept vectors together (i.e., KC), (w_k, w_c, w_d) were set to $(0.2, 0.8, 0.0)$, $(0.4, 0.6, 0.0)$, $(0.6, 0.4, 0.0)$, and $(0.8, 0.2, 0.0)$, respectively. Then, in the effectiveness evaluation (see Sects. 5.5 and 5.6), we take the average relevance scores of the four runs as the final relevance score of the selected ads to their pages.
2. For the strategy which combines keyword vectors and category vectors together (i.e., KD), (w_k, w_c, w_d) were set to $(0.2, 0.0, 0.8)$, $(0.4, 0.0, 0.6)$, $(0.6, 0.0, 0.4)$, and $(0.8, 0.0, 0.2)$, respectively. Then, we average the relevance scores of the four runs as the final relevance score of the selected ads to their pages.
3. For the strategy which combines concept vectors and category vectors together (i.e., CD), (w_k, w_c, w_d) were set to $(0.0, 0.2, 0.8)$, $(0.0, 0.4, 0.6)$, $(0.0, 0.6, 0.4)$, and $(0.0, 0.8, 0.2)$, respectively. Then, we average the relevance scores of the four runs as the final relevance score of the selected ads to their pages.
4. For the strategy which combines keyword vectors, concept vectors and category vectors together (i.e., KCD), (w_k, w_c, w_d) were set based on the above three strategies, namely they were set to the values that produce good results in the three strategies.

Except for keyword matching (i.e., the K strategy), we used the selective Wikipedia matching approach mentioned in Sect. 3.4 as baselines, including: (1) matching based on Strategy 1; (2) matching based on Strategy 2; and (3) matching based on Strategy 3. These candidate strategies are shown in the last three rows in Table 2. However, we here did not consider Wikipedia matching, because a comparison between Wikipedia matching and selective Wikipedia matching has been detailed in the work [31], which shows that compared to Wikipedia matching, selective Wikipedia matching has almost equal accuracy, but is better in terms of efficiency.

Moreover, we also used the semantic–syntactic matching approach proposed in [3] as a baseline (i.e., the KS strategy in Table 2). The KS approach uses a commercial taxonomy for

Table 2 Candidate contextual advertising strategies based on different combinations of feature vectors

Notation	Explanation
K	Measure similarity between pages and ads based on keyword vectors
KC	Measure similarity between pages and ads based on the combination of keyword vectors and concept vectors
KD	Measure similarity between pages and ads based on the combination of keyword vectors and category vectors
CD	Measure similarity between pages and ads based on the combination of concept vectors and category vectors
KCD	Measure similarity between pages and ads based on the combination of keyword vectors, concept vectors and category vectors
KS	Semantic–syntactic matching
S1	Selective Wikipedia matching over Strategy 1 (see Sect. 3.4)
S2	Selective Wikipedia matching over Strategy 2 (see Sect. 3.4)
S3	Selective Wikipedia matching over Strategy 3 (see Sect. 3.4)

classifying pages and ads. However, the taxonomy is commercially built by Yahoo! Corp., and it is not available publicly. Thus, we used a taxonomy from the Wikipedia category system. In addition, to determine a set of classification categories for a given page (or a candidate ad), the approach needs to conduct full-text matching operations between the meta-document of each category and the page. Thus, for each category in the taxonomy, we used the main article to represent its meta-document.[8]

5.3 Relevance ratio evaluation

As described in Sect. 4, we know that the effectiveness of our approach depends to a large extent on whether the concepts and categories in the semantic feature vectors are really related to their page (or ad). In the first group of experiments, we therefore aimed to evaluate the relevance ratio of concepts and categories introduced by our approach with respect to their pages (or ads).

Let $\boldsymbol{p}$ be a page, and $\mathbf{cots}(\boldsymbol{p})$ a set of related concepts generated by Algorithms 1 and 2. Let $\mathbf{cots}^{\mathbf{true}}(\boldsymbol{p})$ be a subset of $\mathbf{cots}(\boldsymbol{p})$, each of which is considered to be relevant or somewhat relevant to $\boldsymbol{p}$ by assessors. Then, the relevance ratio of concepts in $\mathbf{cots}(\boldsymbol{p})$ with respect to the page $\boldsymbol{p}$ is defined as follows:

$$\mathbf{ratio}\,(\boldsymbol{p})\,[\mathbf{cots}(\boldsymbol{p})] = (|\mathbf{cots}^{\mathbf{true}}(\boldsymbol{p})|)/(|\mathbf{cots}(\boldsymbol{p})|) \tag{18}$$

Similarly, the relevance ratio of categories with respect to their page is also defined.

Table 3 presents the relevance ratio of concepts generated by our approach with respect to their pages, where the third row presents the number of generated concepts, i.e., the size of $\mathbf{cots}(\boldsymbol{p})$. Table 3 shows that, when the parameter μ_E is set with a large enough value ($\mu_E = 10^3$ at the last column), the relevance ratio would generally be of the greatest value (97.5 %), and the number of related concepts would be of the smallest value (94.4). However, with an increase of μ_E value, the number of related concepts would decrease and the relevance ratio would increase. Table 4 presents the relevance ratio of generated concepts with respect to their ads, which overall shows similar results to Table 3.

[8] some categories in Wikipedia without main articles would not be included in the taxonomy, so the number of categories in the taxonomy is about 8,000.

Table 3 Relevance ratio of concepts determined by Algorithms 1 and 2 with respect to their pages

Setting (μ_E)	0.5	1.0	1.5	2.0	10^3
Ratio (%)	84.0	89.7	90.7	90.3	91.5
#Concepts	138.2	116.6	109.4	104.8	94.4

Table 4 Relevance ratio of concepts determined by Algorithms 1 and 2 with respect to their ads

Setting (μ_E)	0.004	0.02	0.1	0.5	10^3
Ratio (%)	40.2	75.6	80.1	87.2	87.1
#Concepts	34.8	11.2	6.5	4.7	3.1

Table 5 Relevance ratio of categories determined by Algorithm 3 with respect to their pages ($\mu_E = 1.0$)

Setting (μ_D)	0.25	0.5	1.0	2.0	4.0
Ratio (%)	79.9	81.4	83.9	88.3	90.2
#Concepts	83.2	72.6	58.2	43.0	27.5

Table 6 Relevance ratio of categories determined by Algorithm 3 with respect to their ads ($\mu_E = 0.02$)

Setting (μ_D)	0.025	0.05	0.1	0.2	0.4
Ratio (%)	70.5	73.2	76.4	79.1	80.3
#Concepts	10.1	9.3	7.8	6.8	5.3

A comparison between Tables 3 and 4 shows that, using Algorithm 1, a generic page can obtain more candidate concepts than an ad (94.4 vs. 3.1). This is caused by the smaller size of an ad, which would not only make the number of candidate concepts contained in an ad smaller than that in a page, but also makes the frequency value of each candidate concept in an ad smaller than that in a page. To better enrich the concept vector representation of an ad, we set a smaller μ_E value ($\mu_E = 0.02$) for expanding candidate concepts of an ad. Moreover, we set a larger μ_E value ($\mu_E = 1.0$) for expanding concepts of a page, which is mainly due to the running performance requirement of online contextual advertising.

Tables 5 and 6 present the relevance ratio of categories generated by our approach to their pages and to their ads, respectively. From Tables 5 and 6, we can see that the change of categories to the μ_D value is similar with the change of concepts to the μ_E value, i.e., with the increase of μ_D value, the number of related categories would increase and the relevance ratio would decrease. In our implementation, we assign a smaller value to μ_D for searching related categories for ads ($\mu_D = 0.05$), and a greater value for pages ($\mu_D = 0.5$).

From Tables 3 to 6, we conclude that the concepts and categories introduced by our approach not only enrich the feature representation of pages (or ads), but also have good relevance ratios with respect to their pages (or ads), thereby ensuring the effectiveness of our approach.

5.4 Efficiency evaluation

In the second group of experiments, we aimed to evaluate the running performance of our approach. The hardware and software setting of our experiments is shown in Table 7. In our experiments, the work of generating feature vectors for all the ads has been completed in advance. Thus, in each ad selection for a page, we are only concerned about the execution time consumed by: (1) extracting all the keywords from the page; (2) generating the feature vectors for the page; and (3) calculating the similarity between the page and each ad to choose the most relevant ads. Figure 6 presents the experiment results.

Table 7 Hardware and software setting

Item	Explanation
OS	MS Windows XP SP2 Professional
CPU	Intel Core 2 Duo @ 2.93 GHz
Running memory	2.00 GB
Hard disk	500 GB

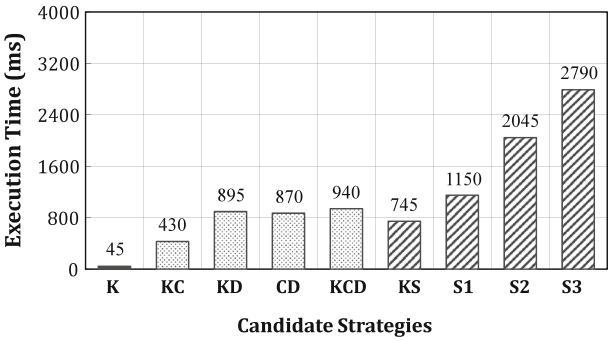


Fig. 6 Execution times for the eight candidate strategies to select relevant ads for a targeted page

As we can see from Fig. 6, the strategies, KC, KD, CD and KCD, which need to generate additional concept vectors or category vectors on the basis of keyword vectors for a page, increase the time spent on ad selection to a certain extent, but such a time increase is not too serious (<1,000 ms). However, the strategies, S1, S2 and S3 (i.e., selective Wikipedia matching), need to spend several seconds (2.0–3.0 s) for each ad selection, obviously, which is worse in terms of efficiency than our recommended KCD strategy. Moreover, for the KS strategy, in order to classify a given page, it needs to conduct full-text matching between each category document and the page, so it is also relatively time-consuming (due to the relatively large number of categories in the taxonomy). Thus, we conclude that our approach obtains a better running performance than existing ones. It should be pointed out that the KC strategy performs better in terms of efficiency than other strategies (i.e., KD, CD and KCD), because this strategy only needs to generate a concept vector for a page, without the need to generate a category vector.

5.5 Effectiveness evaluation

In the third group of experiments, we aimed to evaluate the effectiveness of our approach. In our experiments, we first used the eight candidate strategies, respectively, to select the top-N ads and embed them into each page and then invited the evaluation assessors to score each ad, based on the relevance of the ad to its page. We then averaged the relevance scores given by the assessors. Below, we describe the process of calculating the relevance score for each page.

Let N_u denote the number of assessors. Let $\boldsymbol{p}$ be a targeted page, and N the number of ads expected to be embedded into $\boldsymbol{p}$. Let N_r be the number of times that a candidate strategy C_S (C_S may be K, KC, KD, CD, KS, S1, S2, S3 or KCD) is repeated over the page $\boldsymbol{p}$ (from Sect. 5.2, we know that such candidate strategies such as CD, KC and KD need to run several times under different parameters for each ad selection). Let $\mathbf{score}(u_i, a_j, r_k)$ be the score

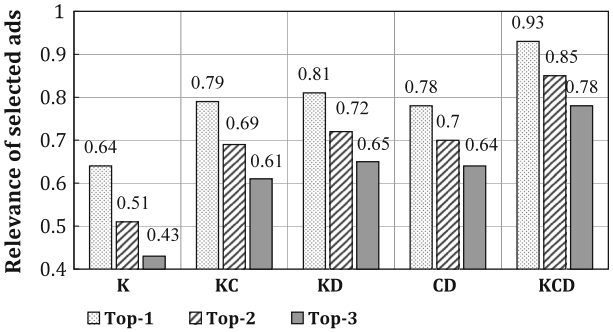


Fig. 7 Average relevance for the candidate ads selected out by the five candidate strategies (i.e., K, KC, KC, CD and KCD) for general pages

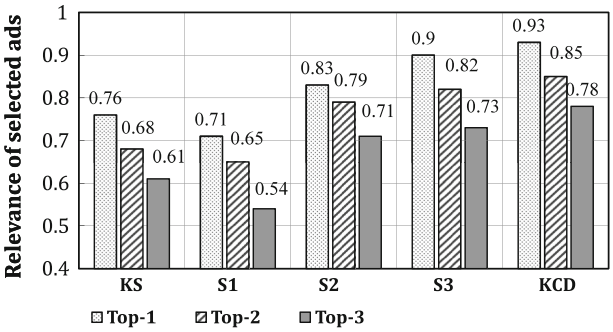


Fig. 8 Average relevance for the candidate ads selected out by the five candidate strategies (i.e., KS, S1, S2, S3 and KCD) for general pages

given by the assessor u_i $(1 \leq i \leq N_u)$ for the ad a_j $(1 \leq j \leq N)$ chosen in the run k $(1 \leq k \leq N_r)$. Then, the relevance score for the page $\boldsymbol{p}$ determined by the strategy C_S is computed as follows:

$$\textbf{relevance}(\boldsymbol{p})[C_S] = \sum_{k=1}^{N_r} \sum_{j=1}^{N} \sum_{i=1}^{N_u} \left(\frac{\textbf{score}(u_i, a_j, r_k)}{N_u \cdot N \cdot N_r} \right) \tag{19}$$

Figures 7 and 8 present the experimental results. Figure 7 shows that keyword matching solely based on a keyword vector performs worse in terms of effectiveness than the strategies that use two or three feature vectors. Figure 7 also shows that the recommended KCD strategy, which combines all the three feature vectors together to measure the similarity between pages and ads, outperforms all the other strategies, especially the benchmark approach (i.e., the keyword matching): compared to keyword matching, the average ad-relevance scores of the KCD strategy dramatically increase by about 90 %, achieving up to about 40 % improvement. Figure 8 shows that compared to selective Wikipedia matching (especially to the S3 strategy), the recommended KCD strategy has approximately equal effectiveness in contextual advertising. Moreover, for the KS strategy, it is sensitive to the classification precision for pages and ads, while the classification itself for short text is a challenging task, which degrades its performance of ad selection to a certain degree. From Figs. 7 and 8, we conclude that our approach, which leverages the Wikipedia concept and category information to enrich

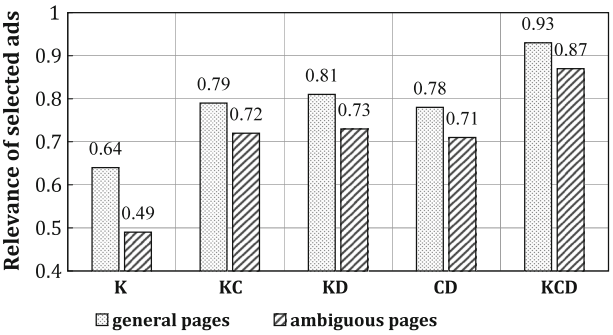


Fig. 9 Average relevance for the candidate ads selected out by the five candidate strategies (i.e., K, KC, KD, CD and KCD) for general pages and ambiguous pages

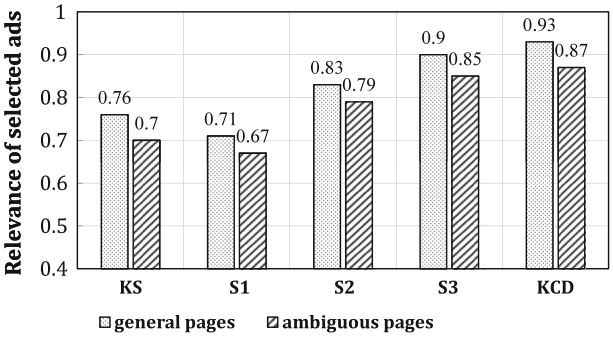


Fig. 10 Average relevance for the candidate ads selected out by the five candidate strategies (i.e., KS, S1, S2, S3 and KCD) for general pages and ambiguous pages

the content representation of pages and ads, can improve the accuracy of selected ads to their pages effectively, consequently increasing the precision of ad selection.

5.6 Ambiguous evaluation

In the fourth group of experiments, to demonstrate that our approach helps to overcome problems such as homonymy and polysemy, the low intersection of keywords and context mismatch, which cannot be solved easily by traditional keyword matching, we have chosen a special dataset that consists of 35 ambiguous pages (which were obtained by using ambiguous keywords to query the search engine). In the pages, there are many ambiguous keywords, such as Puma (company vs. lion), Rock (person vs. music), Driver (software vs. car), Game (software vs. sports) and Window (OS vs. glass). The experimental results are shown in Figs. 9 and 10.

Figure 9 shows that our strategies outweigh the benchmark keyword matching strategy, in terms of the relevance of both the datasets of generic pages and ambiguous pages. Specifically, for ambiguous pages, the relevance score derived from the KCD strategy could reach up to 0.87, resulting in a significant improvement of 106 % over the keyword matching strategy. This reflects that the semantic information contained in the pages and ads has better stability than the surface textual information, i.e., based on the semantic information, the

similarity between pages and ads could be measured more accurately. In addition, Fig. 10 shows that compared to selective Wikipedia matching and semantic–syntactic matching, our recommended strategy still maintains weak superiority. As seen from Figs. 9 and 10, it is concluded that by using our approach, i.e., by integrating the Wikipedia concept and category information into keyword matching to enrich the content representation of pages and ads, we can reduce the negative effect caused by the problem of semantic ambiguity.

6 Related work

6.1 Keyword matching

Traditional keyword matching can be used to estimate ad-relevance by analyzing the co-occurrence of the same keywords or phrases within the ad and within the page. In keyword matching, the cosine metric [26] is often used to calculate the similarity between a page and an ad.

A recent study on applying keyword matching to contextual advertising was described in [25]. In this work, all the ads and pages were represented as vectors in a vector space. To solve the problem of low intersection of keywords, the authors proposed to augment a page with additional keywords from other pages that are similar to that page. Next, the authors explored ten different strategies to select different parts of pages and ads, used as a basis for the vectors of the pages and ads. Last, the authors matched the pages and the ads based on the cosine of the angle between the ad vector and the page vector to select relevant ads for a page.

In [5,21], under the assumption that an ad can be viewed as a noisy translation of a page, the authors selected the ads that provide the best translation for the page. To obtain a relevance score, the authors used the algorithms used in machine translation techniques: NIST and BLEU [33] to determine the quality of machine-translated texts.

In [9,10], an approach to using sentiment detection to improve contextual advertising was presented, which combines contextual advertising matching with sentiment analysis to select ads relevant to the positive (or neutral) aspects of a blog and ranks the ads according to their relevance. In [24], the authors proposed the utilization of lexical graphs created from Web corpora as a means of computing improved content similarity metrics between ads and pages. The results indicated that using lexical graphs can provide evidence of significant improvement in the perceived relevance of the recommended ads. A new architecture (called blog context overlay network) was proposed in [12], to fulfill context matching between blog-based knowledge management systems, and as a conclusion, a measurement for contextual similarity between blogs was also presented.

Recent research in [3] proposed a semantic approach to contextual advertising. To overcome the problems of homonymy and polysemy and context mismatch, the authors proposed to apply automatic classification for pages and ads, so as to help to filter out irrelevant ads and increase the performance of ad selection. The authors also proposed semantic–syntactic matching, which combines the semantic approach with traditional keyword matching. However, it is pointed out in [23] that this approach is sensitive to the classification precision.

As the follow-up work to [3], in [1,2], a technique for ad matching was proposed that is based on the semantic–syntactic matching and the summarization of a page. Using the page summary instead of the whole page lowers the network traffic between a Web page and an ad platform along with decreasing the system load while sacrificing little ad-relevance. However, the problem of classification precision still remains for this approach.

As pointed out in [14,15,23,28], the main drawback of keyword matching is that it may lead to the problems of homonymy and polysemy and context mismatch, resulting in dramatically degrading the relevance of selected ads to their pages.

6.2 Wikipedia matching

To solve the problems encountered in keyword matching, recently, a new technique called Wikipedia matching was proposed, which uses Wikipedia as the reference model to enhance the semantic representation of text documents and thus improve the precision of the similarity measure of text documents. The basic idea of Wikipedia matching is to introduce Wikipedia articles as a semantic reference space, and in turn, the semantics of each text document are reflected and enhanced by projecting the original term space of text documents into this additional reference model.

In [22,23], the authors presented a solution based on Wikipedia matching to contextual advertising. In their work, 1,000 feature articles are first chosen from Wikipedia. Next, for each ad, the feature articles that are related to the ad are selected by using the cosine measure; and for a page, the same procedure is followed and articles which are related to the page are selected. Last, using the feature articles as the reference model on which the ads and the page are re-expressed as vectors, the approach determines the ads that exhibit more relevance to the page, and construct a ranking function to select the most relevant ads for the page.

In [14,15,28], the authors proposed a similar method, aiming at textual document clustering. First, the authors automatically construct a thesaurus of concepts from Wikipedia. Then, they introduce a framework to expand the traditional representation of document terms with semantic relations (i.e., synonymy, hyponymy and associative relations), demonstrating its efficacy in enhancing previous methods for text classification.

In [16], the authors proposed to use Wikipedia to understand a user's query intent, without the need to collect large quantities of examples to train an intent classifier. In this approach, the Wikipedia concepts are used as the intent representation space; therefore, each intent domain is represented as a set of Wikipedia articles and categories; and then, the intent of any input query is identified through mapping the query into the Wikipedia representation space. Compared to previous work, this approach can achieve better coverage to classify queries in an intent domain, although the number of seed intent examples is small.

However, the main drawback of traditional Wikipedia matching is that it may lead to the problems that we mentioned in Sect. 1, thereby dramatically degrading the relevance of selected ads to their pages, and limiting its application in practical contextual advertising.

7 Conclusion and future work

In this paper, we have presented a new contextual advertising approach by incorporating rich Wikipedia knowledge into traditional keyword matching to enrich the content representation of pages and ads. Due to the problems caused by homonymy and polysemy, the low intersection of keywords, etc., traditional keyword matching, which is solely based on the keyword information to measure the text similarity between the pages and the ads, generally has poor accuracy. Although being capable of overcoming the problems encountered in keyword matching, the main drawback of the previously published Wikipedia matching approaches is its limited coverage over semantic concepts and its time-consuming performance, thereby limiting its application in real contextual advertising.

In our proposed approach, we introduce three feature vectors: a keyword vector, a concept vector and a category vector, as the content representation of a page (or an ad). Next, we select relevant ads for a targeted page based on a similarity measure which combines the three feature vectors together, where the keyword vector is used to measure the textual similarity between pages and ads, while the concept vector and the category vector constructed based on Wikipedia knowledge are used to measure the semantic similarity. Last, to evaluate the effectiveness of our approach, we have conducted experiments over a dataset that consists of 10,244 ads, 80 generic pages and 35 ambiguous pages downloaded from the Internet, as well as a set of 730,500 concepts and 24,000 categories chosen from Wikipedia.

From the experimental results, we reached the following two conclusions. First, our approach obtains a satisfactory running performance: the time spent selecting ads for a page is $<$1,000 ms. This is because there is no need to conduct the time-consuming full-text matching operation between pages and many reference articles, which cannot be avoided in the previously published Wikipedia matching approaches to contextual advertising. Second, our approach largely improves the accuracy of selected ads to their pages: the relevance score of embedded ads to their pages is generally $>$0.8. This is due to the fact that we leverage the Wikipedia concept and category information to enrich the semantic representation of pages and ads and then use them to measure the semantic similarity between pages and ads; while compared to the surface textual information contained in pages and ads, the semantic information has better stability.

In summary, the main aim of this paper is to address the framework of our proposal on contextual advertising at semantic conceptual level by leveraging the Wikipedia thesaurus knowledge, and to evaluate the proposed conceptual model in terms of effectiveness, efficiency and the trade-off between them. Apart from the theoretical feasibility and comparably superior performance achieved in the empirical study, there are still a few factors needed to be considered in real application deployment.

The first consideration is the impact of multiple factors on ad selection. Besides main factor of the lexical and syntactic relevance between targeted pages and candidate ads, some cognitive features also have a significant influence on ad selection. For example, aesthetic factor also plays an important role for online advertising. Will the stylistic factors impact users' online ad selection? What is the influential degree of such impact? Which factor is more important? Despite that this study is undoubtedly important in dealing with ad selection, it is out of main interest of this paper. We aim to address this in future.

The second issue is the applicability of our framework in real applications. The running time is an essential key point in real online advertising settings, which should be reasonable and acceptable. In order to speed up ad selection, we divide the whole process into offline and online two parts, where the online part determines the time spent on real online advertising. Thus, how to solve this bottleneck and improve the online processing efficiency is an important task we plan to invest efforts. In addition, our current experimental investigation is carried out on a small-size data set due to the difficulty in collecting such a corpus, so the scalability test is another important task in future works. The obtained experimental results on this collected corpus provide us the feasibility indication in real online application settings.

Another practical concern we have to raise is the dynamic evolution of Wikipedia corpus. The Wikipedia thesaurus knowledge is the base of vector space and similarity calculation. However, Wikipedia changes constantly: new categories are introduced, category and subcategory relationships are changing, and topics can be added or deleted. This will result in changes in the concept and category graph. How to maintain the proposed approach in such case? This is an important topic in Wikipedia related studies. So far, many studies have been conducted to investigate the evolution of Wikipedia knowledge. One simplest approach is

to update the concept and category graph in a regular manner, which will incur in additional computational cost. Leveraging the incremental algorithm in graph structure updating is a practical direction we can explore to reduce this overhead and increase the real-time performance.

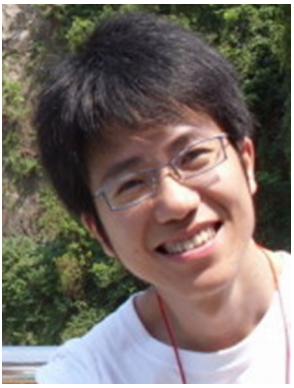
Acknowledgments We thank anonymous reviewers for their very useful comments and suggestions. This work is supported by grants from the National Natural Science Foundation of China (Nos. 61202171 and 61272018), the China Postdoctoral Science Foundation funded projects (Nos. 2013T60623 and 2012M521251), the Zhejiang Provincial Natural Science Foundation of China (Nos. LY12F01016 and LQ13F020009), the Provincial Natural Science Foundation of Hubei (No. 2013CFB415) and the Fundamental Research Funds for the Central Universities (No. CUGL120281).

References

1. Anagnostopoulos A, Broder AZ, Gabrilovich E, Josifovski V, Riedel L (2011) Web page summarization for just-in-time contextual advertising. ACM Trans Intell Syst Tech 3(1):14:1–14:32
2. Anagnostopoulos A, Broder A, Gabrilovich E, Josifovski V, Riedel L (2007) Just-in-time contextual advertising. In: Proceedings of the 16th ACM international conference on information and knowledge management (CIKM'07), ACM, New York, pp 331–340
3. Broder A, Fontoura M, Josifovski V, Riedel L (2007) A semantic approach to contextual advertising. In: Proceedings of the 34th annual ACM SIGIR conference (SIGIR'07), ACM, New York, pp 559–566
4. Chatterjee P, Hoffman DL, Novak TP (2003) Modeling the clickstream: implications for web-based advertising efforts. Mark Sci 22(4):520–541
5. Ciaramita M, Murdock V, Plachouras V (2008) Semantic associations for contextual advertising. J Electron Commer Res 9(1):1–15
6. Comprehensive perl archive network (2007) http://search.cpan.org/jzhang/HTML-ContentExtractor-0.03/lib/HTML/ContentExtractor.pm
7. Evgeniy G, Shaul M (2009) Wikipedia-based semantic interpretation for natural language processing. J Artif Intell Res 34:443–498
8. Evgeniy G, Shaul M (2007) Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: Proceeding of the 20th AAAI international conference on artificial intelligence (AAAI'11), AAAI, San Francisco
9. Fan TK, Chang CH (2010) Sentiment-oriented contextual advertising. Knowl Inf Syst 23(3):321–344
10. Fan TK, Chang CH (2011) Blogger-centric contextual advertising. Expert Syst Appl 38(3):1777–1788
11. Gupta S, Kaiser GE, Grimm P, Chiang MF, Starren J (2005) Automating content extraction of html documents. World Wide Web J 8(2):179–224
12. Gupta S, Kaiser GE, Grimm P, Chiang MF, Starren J (2009) Knowledge distribution via shared context between blog-based knowledge management systems: a case study of collaborative tagging. Expert Syst Appl 36(2):10,627–10,633
13. Hovy E, Navigli R, Ponzetto SP, Collaboratively built semi-structured content and artificial intelligence: the story so far. Artif Intell 194
14. Hu XH, Zhang XD, Lu CM, Park EK, Zhou XH (2009) Exploiting wikipedia as external knowledge for document clustering. In: Proceedings of the 15th ACM SIGKDD conference on knowledge discovery and data mining (SIGKDD'09), ACM, New York, pp 389–396
15. Hu J, Fang LJ, Cao Y, Zeng HJ, Li H, Yang Q, Chen Z (2008) Enhancing text clustering by leveraging wikipedia semantics. In: Proceedings of the 35th annual ACM SIGIR conference (SIGIR'08), ACM, New York, pp 179–186
16. Hu J, Wang G, Lochovsky F, Sun JT, Chen Z (2009) Understanding user's query intent with wikipedia. In: Proceedings of the 18th world wide web conference (WWW'09), ACM, New York, pp 471–480
17. Lacerda A, Cristo M, Andre MG, Fan W, Ziviani N, Ribeiro-Neto B (2006) Learning to advertise. In: Proceedings of the 33th annual ACM SIGIR conference (SIGIR'06), ACM, New York, pp 549–556
18. Mei T, Hua XS, Li SP (2011) Contextual internet multimedia advertising. Proc IEEE 98(8):1416–1433
19. Mei T, Hua XS, Li SP (2008) Contextual in-image advertising. In: Proceeding of the 16th ACM international conference on multimedia (MM'08), ACM, New York, pp 439–448
20. Milne D, Medelyan O, Witten IH (2006) Mining domain-specific thesauri from wikipedia: a case study. In: Proceedings of the 2006 IEEE/WIC/ACM international conference on web intelligence, IEEE, Los Alamitos, pp 442–448

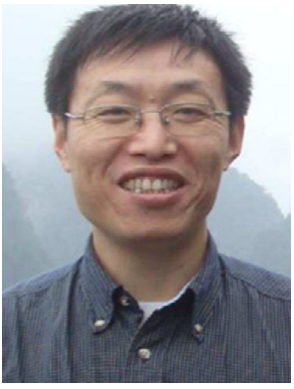
21. Murdock V, Ciaramita M, Plachouras V (2007) A noisy-channel approach to contextual advertising. In: Proceedings of SIGKDD workshops 07, ACM, New York, pp 21–27
22. Pak AN (2011) Using wikipedia to improve prevision of contextual advertising. In: Proceedings of the 4th international conference on human language technology: challenges for computer science and linguistics (LTC'09), Springer, Berlin, pp 533–543
23. Pak AN, Chung CW (2010) A wikipedia matching approach to contextual advertising. World Wide Web J 13(3):251–274
24. Papadopoulos S, Menemenis F, Kompatsiaris Y, Brato B (2009) Lexical graphs for improved contextual ad recommendation. In: Proceedings of the 31st European conference on information retrieval (ECIR'09), pp 216–227
25. Ribeiro-Neto B, Cristo M, Golgher PB, Moura ES (2005) Impedance coupling in content-targeted advertising. In: Proceedings of the 32th annual ACM SIGIR conference (SIGIR'05), ACM, New York, pp 496–503
26. Salton G, Wong A, Yang C (1976) A vector space model for automatic indexing. Commun ACM 18(11):613–620
27. Wang P, Hu J, Zeng HJ, Chen Z (2009) Using wikipedia knowledge to improve text classification. Knowl Inf Syst 19(3):265–281
28. Wang C, Zhang P, Choi R, Eredita M (2002) Understanding consumers attitude toward advertising. In: Proceeding of the 8th Americas conference on information systems (AMCIS'02), pp 1143–1148
29. Wu ZD, Xu GD, Pan R, Zhang YC, Hu ZW, Lu JF (2011) Leveraging wikipedia concept and category information to enhance contextual advertising. In: Proceedings of the 20th ACM international conference on information and knowledge management (CIKM'11), ACM, New York, pp 2105–2108
30. Wu HC, Luk RWP, Wong KF, Kwok KL (2008) Interpreting tf-idf term weights as making relevance decisions. ACM Trans Inf Syst 26(3):13–50
31. Wu ZD, Xu GD, Zhang YC, Dolog P, Lu CL (2012) An improved contextual advertising matching approach based on wikipedia knowledge. Comput J 55(3):277–292
32. Yih W, Goodman J, Carvalho VR (2006) Finding advertising keywords on web pages. In: Proceedings of the 15th world wide web conference (WWW'06), ACM, New York, pp 213–222
33. Zhang Y, Vogel S (2004) Measuring confidence intervals for the machine translation evaluation metrics. In: Proceedings of the 10th international conference on theoretical and methodological issues in machine translation (TMI'04), ACM, New York, pp 4–6

Guandong Xu is a research fellow in Advanced Analytics Institute, Faculty of Engineering and Information Technology, University of Technology, Sydney. He obtained his PhD degree in Computer Science from Victoria University in 2008. After that he worked as a postdoctoral research fellow in Centre for Applied Informatics at Victoria University and then postdoc in Department of Computer Science at Aalborg University, Denmark. He is an Endeavour postdoctoral research fellow in the University of Tokyo in 2008. His research interests include Web information retrieval, Web mining and Web services.



Zongda Wu is an associate professor in Computer Science at Wenzhou University. He received his BSc degree in Computer Science from Wenzhou University in 2005 and PhD degree in Computer Science from Huazhong University of Science and Technology (HUST) in 2009. Now, he is also a postdoctoral research fellow with School of Computer Science and Technology at University of Science and Technology of China (USTC). His research interests are primarily in the area of information retrieval and information security.

Guiling Li is an assistant professor in School of Computer Science at China University of Geosciences (Wuhan). She received her BSc degree in Computer Science from China University of Geosciences (Wuhan) in 2005 and PhD degree in Computer Science from Huazhong University of Science and Technology (HUST) in 2012. Her research interests are primarily in the area of data mining and knowledge discovery, data management for time series data, sensor data and data streams.



Enhong Chen is a professor and a vice dean at School of Computer Science and Technology, University of Science and Technology of China (USTC), IEEE senior member. He received his PhD degree in Computer Science from USTC in 1996. He has been actively involved in the research community by serving as a PC member for more than 30 conferences, such as ICTAI 2006, ICTAI 2007, AIRS 2009, AIRS 2010 and KDD 2010. His research interests include semantic Web, machine learning and data mining, Web information processing and constraint satisfaction problem.